Features

- Selectable resolutions, 8 to 20 bits (device dependent)
- Eleven input ranges for each resolution
- Sample rate 10 sps to 384 ksp
- Operational modes:
  - Single sample
  - Multi-sample
  - Continuous mode
  - Multi-sample (Turbo)
- High input impedance input buffer
  - Selectable input buffer gain (1, 2, 4, 8) or input buffer bypass
- Multiple internal or external reference options
- Automatic power configuration
- Up to four run-time ADC configurations

General Description

The Delta Sigma Analog to Digital Converter (ADC_DelSig) provides a low-power, low-noise front end for precision measurement applications. You can use it in a wide range of applications, depending on resolution, sample rate, and operating mode. It can produce 16-bit audio; high speed and low resolution for communications processing; and high-precision 20-bit low-speed conversions for sensors such as strain gauges, thermocouples, and other high-precision sensors. When processing audio information, the ADC_DelSig is used in a continuous operation mode. When used for scanning multiple sensors, the ADC_DelSig is used in one of the multi-sample modes. When used for single-point high-resolution measurements, the ADC_DelSig is used in single-sample mode.

Delta-sigma converters use oversampling to spread the quantization noise across a wider frequency spectrum. This noise is shaped to move most of it outside the input signal's bandwidth. An internal low-pass filter is used to filter out the noise outside the desired input signal bandwidth. This makes delta-sigma converters good for both high-speed medium-
resolution (8 to 16 bits) applications, and low-speed high-resolution (16 to 20 bits) applications. The sample rate can be adjusted between 10 and 384000 samples per second, depending on mode and resolution. Choices of conversion modes simplify interfacing to single streaming signals such as audio, or multiplexing between multiple signal sources.

The ADC_DelSig is composed of three blocks: an input amplifier, a third-order delta-sigma modulator, and a decimator (see Figure 1). The input amplifier provides a high-impedance input and a user-selectable input gain. The decimator block contains a four-stage CIC decimation filter and a post-processing unit. The CIC filter operates on the data sample directly from the modulator. The post-processing unit optionally performs gain, offset, and simple filter functions on the output of the CIC decimator filter.

**Figure 1. ADC_DelSig Block Diagram**

![ADC_DelSig Block Diagram](image)

**Input/Output Connections**

Input and output connections to the ADC_DelSig component are displayed as pins on the component symbol in the schematic view. An asterisk (*) in the list of I/Os indicates that the I/O may be hidden on the symbol under the conditions listed in the description of that I/O.

**nVref – Input** *

The nVref is an optional pin. It is shown if you have selected the Enable_Vref_Vssa checkbox. This allows you to connect the ADC’s reference Vssa to the analog global (AGL[6]). If the Enable_Vref_Vssa checkbox is not selected, this pin is not shown on the symbol. See the parameter Enable_Vref_Vssa description for more information.

**+Input – Analog**

Positive analog signal input to the ADC_DelSig. The positive input signal is always present in both the single-ended and differential-input modes. The ADC converter output returns a value that represents the difference in voltage between positive input and the negative input signal.
–Input – Analog *

Negative analog signal input to the ADC_DelSig. The negative input pin is only displayed on the component when the ADC Input Mode is set to Differential. When ADC Input Mode is set to Single, the negative input is connected to either Vssa or Vref, depending on the input range selected.

**soc – Input** *

Start of Conversion (soc) starts hardware-triggered ADC conversions when a rising edge is detected. A rising edge on this pin has the same effect as calling the ADC_StartConvert() function. This input is shown when the user selects the Hardware SOC parameter, which enables an external pin to start conversion. If Hardware SOC is not selected, the I/O pin on the component will be hidden. In Single Sample mode, a single conversion is executed and the ADC halts. In Continuous and other modes, ADC conversions continue until either the ADC_StopConvert() or ADC_Stop() functions are executed.

**aclk – Input** *

External clock source. This pin is present if the Clock Source parameter is set to External. If the Clock Source parameter is set to Internal, the clock is configured automatically within the component and the aclk pin is not shown. The aclk input is a clock that is generated outside the component. This clock signal may be derived internal to the chip or from a source external to the PSoC. Set this clock to the value displayed in the Clock Frequency parameter to achieve the selected sample rate. The duty cycle should be 50 percent. This clock determines the conversion rate as a function of conversion method and resolution. A clock that is external to the ADC should always be based on the system clock. If a more accurate and stable clock is required, the system clock should be based on an external (external to the PSoC) clock or oscillator.

**eoc – Output**

A rising edge on the End of Conversion (eoc) signals that a conversion is complete. The pin goes high for one ADC clock period. The eoc is typically connected to an interrupt or DMA request. The DMA request is typically used to transfer the conversion output to system RAM, DFB, or another component. There is already an interrupt, internal to the component, which is connected to this signal.

**Note** The EOC for PSoC 5 is restricted in how it can be connected. The EOC terminal is not routable except to a single DMA component.
Component Parameters

The Delta Sigma ADC is a highly configurable analog to digital converter. Drag an ADC_DelSig component onto your design and double-click it to open the *Configure* dialog.

![Configure Dialog](image)

- **Name**: ADC_DelSig_1
- **Sampling**
  - **Conversion Mode**: 2 - Continuous
  - **Resolution**: 16 bits
  - **Conversion Rate**: 10000 SPS
  - **Clock Frequency**: 640.000 kHz
- **Input Options**
  - **Input Mode**: Differential
  - **Input Range**: +/-1.024V (Input +/- Vref)
  - **Buffer Gain**: 1
  - **Buffer Mode**: Fail to Rail
- **Reference**
  - **Vref**: Internal 1.024 Volts
  - **Volts**: 1.0240
Sampling

Conversion Mode

The ADC_DelSig operates in one of four modes:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - Single Sample</td>
<td>The ADC produces one sample per startup conversion. The interrupt should be enabled for ADC conversion with Single Sample conversion mode when the resolution is above 16 bits. To do so, enable the Global Interrupt (by calling <code>CYGlobalIntEnable</code>) in the application (<code>main.c</code>).</td>
</tr>
<tr>
<td>1 - Multi-Sample</td>
<td>Multi-sample mode captures single samples back to back, resetting itself and the modulator between each sample automatically. This mode is useful when the input is switched between multiple signals. The filters are flushed between each sample so previous samples do not affect the current conversion. <strong>Note</strong> Take care when switching signals between ADC conversions. Either switch the input quickly between conversions with hardware control or stop the ADC conversion (<code>ADC_StopConvert()</code>) while switching the input. Then restart the ADC conversion (<code>ADC_StartConvert()</code>) after the new signal has been connected to the ADC. Failure to do this may result in contamination between signals in the ADC results.</td>
</tr>
<tr>
<td>2 - Continuous</td>
<td>Continuous sample mode operates as a normal delta-sigma converter. Use this mode when measuring a single input signal. There is a latency of three conversion times before the first conversion result is available. This is the time required to prime the internal filter. After the first result, a conversion will be available at the selected sample rate. Do not use this mode when multiple signals are multiplexed and measured with a single ADC.</td>
</tr>
<tr>
<td>3 - Multi-Sample (Turbo)</td>
<td>The Multi-Sample (Turbo) mode operates identically to the Multi-Sample mode for resolutions of 8 to 16 bits. For resolutions of 17 to 20 bits, the performance of this mode is about four times faster than the Multi-Sample mode. <strong>Note</strong> Take care when switching signals between ADC conversions. Either switch the input quickly between conversions with hardware control or stop the ADC conversion (<code>ADC_StopConvert()</code>) while switching the input. Then restart the ADC conversion (<code>ADC_StartConvert()</code>) after the new signal has been connected to the ADC. Failure to do this may result in contamination between signals in the ADC results.</td>
</tr>
</tbody>
</table>

All four ADC modes fully flush the decimator when the ADC initially starts conversions. This ensures that the first reading from the ADC is valid as long as the input voltage is stable before starting conversions with either the `ADC_StartConvert()` API or when triggered by the “soc” input. Although all modes reset the decimator when starting the ADC, only the continuous mode does not reset the decimator between readings. Because of this, the first reading in continuous mode takes four times longer than the subsequent readings. When using an analog mux to scan between multiple inputs, make sure that the ADC is not running while the input switches are changing. To switch input between samples when using modes other than continuous, use the analog hardware mux.

When changing the **Conversion Mode** parameter, the clock frequency changes to maintain the selected sample rate. If the ADC clock frequency exceeds the minimum or maximum, an error message is displayed.
# Configs

You may define up to four different configurations using the # Configs parameter. For example, the system may require switching between continuous mode 16 bits, 48 ksp for audio; single sample mode 20 bits, 60 sps for low level analog sensors; and multi-sample mode for 12-bit general-purpose multi-channel data logging. All configurations must use the same input mode, all single-ended or all differential.

By default, the ADC is set to the first configuration (Config1) unless the ADC_SelectConfiguration() function sets the configuration to a different value. When selecting between two and four configurations, additional tabs appear in the Configure dialog. These multiple configurations allow you to change modes during run time. Each configuration is contained in its own tab.

There are some considerations when using multiple ADC configurations:

- All configurations must use the same input mode, all single-ended or all differential.
- The Common tab contains the Hardware SOC, Clock Source, Enable Charge Pump Clock, and Enable_Vref_Vssa parameters, which are common to all modes. These parameters are described under Common Settings.
- The Vref parameter also has some restrictions. If the options on Config 1 set an external reference or bypass mode, the other configurations may select the same mode or use the internal reference.
- Each configuration has a separate Interrupt Service Routine function. When the ADC_SelectConfiguration() function is called, the interrupt vector changes to the corresponding interrupt vector routine.
- When using a clock external to the ADC (either external to the chip or supplied from a user-selected internal clock), the required clock rate is shown in the Clock Frequency field. It is your responsibility to provide the appropriate clock for each configuration.

Resolution

The resolution of the ADC_DelSig is entered as an integer value, limited to 8 to 20 bits. Higher resolution results in lower sample rate. Default resolution is 16 bits. When you change the Resolution parameter, the clock frequency changes to maintain the selected sample rate. If the ADC clock frequency exceeds the minimum or maximum an error message will be displayed.

Delta-sigma ADCs have inherent instability, resulting in nonlinearity at the positive and negative limits of the operating range. To correct for this, the input has been attenuated by 10 percent at the front end of the modulator. The post processor then compensates for this attenuation with a gain of about 1.11. The end result expands the input range by 10 percent. For example, if you select the input range ±1.024 V, the actual range of the ADC is approximately ±1.126 V. The usable input range remains ±1.024 V, but the ADC will not saturate until the input exceeds ±1.126 V.
The digital output from the ADC also over-ranges by 10 percent. If the ADC is configured for 10-bit operation, normally a 10-bit differential ADC’s output will range from –512 to 511, for an input of –1.024 to +1.022, respectively. Because of this additional 10 percent of range, the digital output will not saturate until about ±563 counts, instead of –512 to 511.

This is not normally a concern unless a resolution of 8 or 16 bits is selected. When the resolution is set to either 8 or 16 bits, make sure that the numerical value does not wrap around from its most positive or negative value to a negative or positive value, respectively. To make sure that this does not occur, it is good practice to use the API function that returns a word larger than the set resolution. For example, if the resolution is set to 16 bits and there is a possibility that the most positive value may be larger than 32767 or less than –32768, use the ADC_GetResult32() function instead of ADC_GetResult16(). The proper 16-bit value is returned without over-ranging. When the resolution is set to 8 bits and the ADC output values may be less than –128 or greater than 127, use the ADC_GetResult16() function. The proper 8-bit value is returned without over-ranging.

Figure 2. Sample Rate Limits for ADC_DelSig

Conversion Rate

ADC conversion rate is entered as an integer decimal value in samples per second (SPS). The maximum sample rate is a function of resolution, sample mode, and maximum operating clock frequency; the higher the resolution, the lower the sample rate. The minimum clock for all resolutions is 128 kHz. The maximum clock for resolutions between 8 and 15 bits is 6.144 MHz. The maximum clock for resolutions between 16 and 20 bits is 3.027 MHz.

See Figure 2 for valid conversion rates for each resolution and conversion mode combination; the same information is presented in tabular form in Table 1.
The following data applies to ADC range = ±1.024 V with Buffer Gain = 1.0.

### Table 1. Sample Rate Limits for ADC_DelSig (Buffer Gain = 1)

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Single Sample</th>
<th>Multi-Sample</th>
<th>Continuous</th>
<th>Multi-Sample Turbo</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min</td>
<td>Max&lt;sup&gt;1&lt;/sup&gt;</td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>8</td>
<td>1911</td>
<td>91701</td>
<td>1911</td>
<td>91701</td>
</tr>
<tr>
<td>9</td>
<td>1543</td>
<td>74024</td>
<td>1543</td>
<td>74042</td>
</tr>
<tr>
<td>10</td>
<td>1348</td>
<td>64673</td>
<td>1348</td>
<td>64673</td>
</tr>
<tr>
<td>11</td>
<td>1154</td>
<td>55351</td>
<td>1154</td>
<td>55351</td>
</tr>
<tr>
<td>12</td>
<td>978</td>
<td>46900</td>
<td>978</td>
<td>46900</td>
</tr>
<tr>
<td>13</td>
<td>806</td>
<td>38641</td>
<td>806</td>
<td>38641</td>
</tr>
<tr>
<td>14</td>
<td>685</td>
<td>32855</td>
<td>685</td>
<td>32855</td>
</tr>
<tr>
<td>15</td>
<td>585</td>
<td>28054</td>
<td>585</td>
<td>28054</td>
</tr>
<tr>
<td>16</td>
<td>495</td>
<td>11861</td>
<td>495</td>
<td>11861</td>
</tr>
<tr>
<td>17</td>
<td>124</td>
<td>2965</td>
<td>124</td>
<td>2965</td>
</tr>
<tr>
<td>18</td>
<td>31</td>
<td>741</td>
<td>31</td>
<td>741</td>
</tr>
<tr>
<td>19</td>
<td>4</td>
<td>93</td>
<td>4</td>
<td>93</td>
</tr>
<tr>
<td>20</td>
<td>2</td>
<td>46</td>
<td>2</td>
<td>46</td>
</tr>
</tbody>
</table>

The ADC buffer has a finite gain bandwidth which affects settling time. Increasing the buffer gain reduces the available maximum sample rate. The maximum sample rate is the sample rate in Table 1 divided by the buffer gain. Other ranges and buffer gains also affect the maximum sample rate.

When changing the Conversion Rate parameter, the clock frequency changes to maintain the selected sample rate. If the ADC clock frequency exceeds the minimum or maximum, an error indication displays next to the parameter. See Invalid Settings.

### Range [ __ SPS ]

This field is a read-only (always unavailable) area that displays the minimum and maximum available conversion rate for the current settings.

---

<sup>1</sup> The effective maximum sample rate for Single Sample mode will be lower than shown. There are postprocessing steps that are overlapped with the conversion in Multi-Sample mode that can’t be overlapped for Single Sample mode. This overhead is not included in the calculation shown. If sampling is not started using the hardware start of conversion method, the time to start the sample with software must also be included in the maximum sample rate calculation.
Clock Frequency

This text box is a read-only (always unavailable) area that displays the required clock rate for the selected operating conditions: conversion mode, resolution, conversion rate, input range, and buffer gain. It is updated when any of these conditions change. The clock frequency is displayed with a resolution of 1 Hz. If the required clock frequency for the selected operating conditions is outside of the minimum and maximum limits, an error indication displays next to the parameter. See Invalid Settings.

The clock frequency is calculated based on the resolution, conversion mode, and conversion rate. The rate is displayed in the Design-Wide Resources Clock Editor, which always shows the clock frequency for Config 1. The ADC API sets the current clock frequency based on the configuration selected during run time when the Clock Source parameter is set to Internal.

When you use a clock external to the ADC (either external to the chip or supplied from a user-selected internal clock), the required clock rate is displayed in this field. It is your responsibility to provide the appropriate clock for each configuration.

Invalid Settings

The parameters Conversion Mode, Resolution, and Conversion Rate all affect the ADC clock frequency. Changing any of these parameters may cause the ADC clock frequency to exceed the maximum or minimum rate. The maximum ADC frequency is a function of Resolution, Buffer Gain, and Input Range. If an invalid setting for these parameters occurs, a red circle with an exclamation point appears, as shown in Figure 3.

Figure 3. Invalid Settings Notification

If you hover the cursor over one of the error symbols, an error message appears. Change the parameters as needed to comply with the ADC specifications.
**Input Options**

**Input Mode**

The ADC is inherently differential; however, you may use this parameter to simplify single-ended use.

This parameter configures the ADC for a differential or single-ended input. The default selection is **Differential**. In this mode, both negative and positive inputs are shown on the symbol. When single-ended mode is selected, the negative input to the ADC is connected to Vssa.

![Differential and Single Ended Modes](image)

Note that the Delta-Sigma ADC in PSoC 3 and PSoC 5 devices inherently has differential input. Single-ended mode is implemented by internally connecting the negative input to ground and shifting the sign bit out of the 2’s complement number.

Figure 4 shows the input-output characteristics of the delta-sigma ADC in VSSA-to-VDDA input range. When the ADC input (Vp-Vn) varies from –VDDA to +VDDA, the ADC counts linearly increase from minimum count (0x8000) to maximum count (0x7FFF). The single-ended mode is implemented by internally connecting the negative input (Vn) to ground. When this is done, the positive input (Vp) can go from 0 to VDDA and the negative input is fixed at 0 V, restricting the range of ADC input (Vp-Vn) to ‘0 to VDDA’. So, only the upper half of the X-axis is usable. But, when the device has a negative offset and the positive input (Vp) is 0 V, the ADC output wraps around. For example, if the negative offset is –1 counts, the ADC output for 0 V will be 0xFFFF, which corresponds to the most positive input in single-ended mode. To prevent this from happening, you must perform offset correction.
Figure 4. Delta-Sigma ADC I/O Characteristics

Note that when you configure the ADC for 16-bit single-ended mode, the output doesn’t stop at 0x7FFF, but goes all the way up to 0xFFFF. As an example, a 16-bit ADC is implemented by considering a 17-bit differential ADC and ignoring half of the range (although the internal implementation is slightly different from that).

The **Input Mode** can only be set in Config 1. If you use more than one configuration, all configurations must use the Input Mode set in Config 1. If your system requires both single-ended and differential modes, you must select differential mode and you can use an analog mux to connect Vssa to the negative input of the ADC to use as a single-ended ADC.

This parameter controls the options available in the **Input Range** parameter.

**Input Range**

This parameter configures the ADC for a given input range. This configures the input to the ADC and is independent of the **Buffer Gain** setting. The analog signals connected to the IC must be between Vssa and Vdda no matter what input range settings are used.

The absolute maximum of the ADC input range is always dictated by the absolute maximum and minimum of the **Buffer Mode**.

The options available for this parameter vary depending on the **Input Mode** selection; Table 2 and Table 3 describe the options.
The following options are available when Input Mode is set to Differential. For systems where both single-ended and differential signals are scanned, connect the negative input to Vssa when scanning a single-ended input. Depending on the application, you can select Rail to Rail, Level Shift, or Bypass Buffer using the Buffer Mode parameter. See the Buffer Mode parameter description for more details.

You can use an external reference to provide a different operating range. The usable input range can be calculated with the applicable equation.

Table 2. Differential Input Range Options

<table>
<thead>
<tr>
<th>Input Range Internal Ref (External Ref)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>±1.024 V (–Input ± Vref)</td>
<td>When using the internal reference (1.024 V), the input range is –Input ± 1.024 V. If the negative input is connected to 2.048 V the usable input range is 2.048 ± 1.024 V or 1.024 to 3.072 V.</td>
</tr>
<tr>
<td>±2.048 V (–Input ± 2*Vref)</td>
<td>When using the internal reference (1.024 V), the input range is –Input ± 2.048 V. If the negative input is connected to 2.028 V the usable input range is 2.048 ± 2.048 V or 0.0 to 4.096 V.</td>
</tr>
<tr>
<td>±6.144 V (–Input ± 6*Vref)</td>
<td>When using the internal reference (1.024 V), the input range is –Input ± 6.144 V, but not exceeding maximum electrical input range. You can use this mode to measure the supply voltages when connecting the negative input to Vssa. If you intend to measure the supply you must bypass the buffer.</td>
</tr>
<tr>
<td>±0.512 V (–Input ± Vref/2)</td>
<td>When using the internal reference (1.024 V), the input range is –Input ± 0.512 V. If –Input is connected to 1.0 V the usable input range is 1.0 ± 0.512 V or 0.488 to 1.512 V.</td>
</tr>
<tr>
<td>±0.256 V (–Input ± Vref/4)</td>
<td>When using the internal reference (1.024 V), the input range is –Input ± 0.256 V. If –Input is connected to 1.0 V the usable input range is 1.0 ± 0.256 V or 0.744 to 1.256 V.</td>
</tr>
<tr>
<td>±0.128 V (–Input ± Vref/8)</td>
<td>When using the internal reference (1.024 V), the input range is –Input ± 0.128 V. If –Input is connected to 1.0 V the usable input range is 1.0 ± 0.128 V or 0.872 to 1.128 V.</td>
</tr>
<tr>
<td>±0.064 V (–Input ± Vref/16)</td>
<td>When using the internal reference (1.024 V), the input range is –Input ± 0.064 V. If –Input is connected to 1.0 V the usable input range is 1.0 ± 0.064 V or 0.936 to 1.064 V.</td>
</tr>
</tbody>
</table>

The options listed in the previous table depend on the Vref setting. The following table shows examples with the Vref equal to 1.024 V and 1.200 V. The leading numeric value depends on the Vref setting in the Reference section.

<table>
<thead>
<tr>
<th>With Vref = 1.024 V</th>
<th>With Vref = 1.2 V</th>
</tr>
</thead>
<tbody>
<tr>
<td>±1.024 V (–Input ± Vref)</td>
<td>±1.200 V (–Input ±Vref)</td>
</tr>
<tr>
<td>±2.048 V (–Input ± Vref*2)</td>
<td>± 2.400 V (–Input ± Vref*2)</td>
</tr>
<tr>
<td>±6.144 V (–Input ± 6*Vref)</td>
<td>±7.200 V (–Input ± 6*Vref)</td>
</tr>
<tr>
<td>±0.512 V (–Input ± Vref/2)</td>
<td>±0.600 V (–Input ± Vref/2)</td>
</tr>
<tr>
<td>±0.256 V (–Input ± Vref/4)</td>
<td>±0.300 V (–Input ± Vref/4)</td>
</tr>
</tbody>
</table>
The following options are available when **Input Mode** is set to **Single**. To simulate single-ended operation, connect the negative input to an internal reference value (Vssa or Vref).

Depending on the application, you can select **Rail to Rail**, **Level Shift**, or **Bypass Buffer** using the **Buffer Mode** parameter. See the **Buffer Mode** parameter description for more details.

You can use an external reference to provide a different operating range. The usable input range can be calculated with the applicable equation.

### Table 3. Single-Ended Input Range Options

<table>
<thead>
<tr>
<th>Input Range Internal Ref (External Ref)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vssa to 1.024 V (0 to Vref)</td>
<td>When using the internal reference (1.024 V), the usable input voltage to the ADC is 0.0 to 1.024 V.</td>
</tr>
<tr>
<td>Vssa to 2.048 V (0.0 to 2*Vref)</td>
<td>When using the internal reference (1.024 V), the usable input voltage to the ADC is 0.0 to 2.048 V. This range requires that the input buffer gain be equal to 1. If a gain other than 1 is selected, the ADC will not operate properly. This is because the reference voltage in this case is connected to the negative input directly. The gain of the input buffer will only amplify the positive input and not the negative, causing an unbalanced amplification.</td>
</tr>
<tr>
<td>Vssa to Vdda</td>
<td>This mode is ratiometric of the supply voltage. The input range is Vssa to Vdda. Do not use an external reference for this setting. This range requires that the input buffer gain be equal to 1. If a gain other than 1 is selected, the ADC will not operate properly.</td>
</tr>
<tr>
<td>Vssa to 6.144 V (Vssa to 6*Vref)</td>
<td>When using the internal reference (1.024 V), the input range is 0.0 to 6.144 V, but not exceeding maximum electrical input range. You can use this mode to measure the supply voltage. If you intend to measure the supply you must bypass the buffer.</td>
</tr>
</tbody>
</table>

The following table shows the example with Vref = 1.024 V and 1.2 V. The leading numeric value depends on the **Vref** setting in the **Reference** section.

<table>
<thead>
<tr>
<th>Vssa to 1.024 V (0.0 to Vref)</th>
<th>Vssa to 1.200 V (0.0 to Vref)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vssa to 2.048 V (0.0 to Vref*2)</td>
<td>Vssa to 2.400 V (0.0 to Vref*2)</td>
</tr>
<tr>
<td>Vssa to Vdda</td>
<td>Vssa to Vdda</td>
</tr>
<tr>
<td>Vssa to 6.144 V (0.0 to 6*Vref)</td>
<td>Vssa to 7.200 V (0.0 to 6*Vref)</td>
</tr>
</tbody>
</table>

**Note** When selecting the Vss to Vdd option, the customizer automatically selects either the Vdda/4 or Vdda/3 reference. This selection is based on the value of Vdda entered in the design-wide resources.
Buffer Gain
This parameter selects the ADC input buffer gain. The ADC buffer has a finite gain bandwidth which affects settling time. Increasing the buffer gain reduces the available maximum sample rate. The maximum sample rate is the sample rate in Table 1 divided by the buffer gain.

To achieve the highest signal-to-noise ratio, it is important to use the full range of the ADC. The input buffer can be used to amplify the input signal to make use of the full range of the ADC. Make sure the Buffer_Gain and ADC_Input_Range settings are compatible.

<table>
<thead>
<tr>
<th>Buffer Gain</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sets input buffer gain at 1.</td>
</tr>
<tr>
<td>2</td>
<td>Sets input buffer gain at 2.</td>
</tr>
<tr>
<td>4</td>
<td>Sets input buffer gain at 4.</td>
</tr>
<tr>
<td>8</td>
<td>Sets input buffer gain at 8.</td>
</tr>
</tbody>
</table>

Buffer Mode
This parameter selects the ADC input buffer mode. The ADC has maximum sample rate when the buffer is used. The unbuffered modes have slightly reduced bandwidth.

<table>
<thead>
<tr>
<th>Buffer Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bypass Buffer</td>
<td>Disables the input buffer gain. If selected, the buffer is disabled to reduce the overall power consumption. The Buffer Gain parameter does not have any effect if this mode is selected. If this mode is selected then input impedance is reduced to less than 500 kΩ. See Figure 5 for specifics on range.</td>
</tr>
<tr>
<td>Rail to Rail</td>
<td>Sets the input buffer mode to rail to rail. See Figure 5 for specifics on range.</td>
</tr>
<tr>
<td>Level Shift</td>
<td>Sets the input buffer mode to Level Shift. Both positive and negative input buffers are used. Level Shift mode allows you to go below the V_{SSA} but not all the way to V_{DDA}. See Figure 5 for specifics on range.</td>
</tr>
</tbody>
</table>

Figure 5 shows the ADC range for all Buffer Modes.
Figure 5. ADC Range

Reference

Vref

This parameter selects the ADC_DelSig reference voltage and configuration. The reference voltage sets the range of the ADC.

<table>
<thead>
<tr>
<th>ADC_Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Vref 1.024 Volts</td>
<td>Use the internal 1.024-V reference (default)</td>
</tr>
<tr>
<td>Internal Vref, Bypassed on P0[3]</td>
<td>Use the internal 1.024-V reference and allow an external bypass capacitor to be connected on pin P0[3].</td>
</tr>
<tr>
<td>Internal Vref, Bypassed on P3[2]</td>
<td>Use the internal 1.024-V reference and allow an external bypass capacitor to be connected on pin P3[2].</td>
</tr>
<tr>
<td>External Vref on P0[3]</td>
<td>Use an external reference on pin P0[3]. See Delta-Sigma ADC DC Specifications for the allowable range.</td>
</tr>
<tr>
<td>Internal Vdda/4</td>
<td>Use the internal Vdda/4 reference.</td>
</tr>
<tr>
<td>Internal Vdda/4, Bypassed on P0[3]</td>
<td>Use the internal Vdda/4 reference and allow an external bypass capacitor to be connected on pin P0[3].</td>
</tr>
<tr>
<td>Internal Vdda/4, Bypassed on P3[2]</td>
<td>Use the internal Vdda/4 reference and allow an external bypass capacitor to be connected on pin P3[2].</td>
</tr>
</tbody>
</table>

2 The accuracy and signal-to-noise ratio are highly dependent on the quality of the reference. The reference supplied to the ADC can be bypassed on either port P0[3] or port P3[2]. The use of an external bypass capacitor is recommended. The reference noise is frequency dependent, as shown in the measurements of Figure 6.
### ADC Reference

<table>
<thead>
<tr>
<th>ADC_Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Vdda/3</td>
<td>Use the internal Vdda/3 reference. This option is valid only for PSoC 3 Production and PSoC 5 ES2 silicon or later. Selecting this option with PSoC 3 ES2 and PSoC 5 silicon causes compilation error.</td>
</tr>
<tr>
<td>Internal Vdda/3, Bypassed on P0[3]</td>
<td>Use the internal Vdda/3 reference and allow an external bypass capacitor to be connected on pin P0[3]. This option is valid only for PSoC 3 Production silicon. Selecting this option with PSoC 3 ES2 and PSoC 5 silicon causes compilation error.</td>
</tr>
<tr>
<td>Internal Vdda/3, Bypassed on P3[2]</td>
<td>Use the internal Vdda/3 reference and allow an external bypass capacitor to be connected on pin P3[2]. This option is valid only for PSoC 3 Production silicon. Selecting this option with PSoC 3 ES2 and PSoC 5 silicon causes compilation error.</td>
</tr>
</tbody>
</table>

### Figure 6. Reference Noise

When the integrated reference noise is less than the integrated quantization noise over the band up to $f_{SAMPLE}/2$, the reference does not limit noise performance. You can select a suitable larger capacitor value according to resolution using the following table.

<table>
<thead>
<tr>
<th>Resolution (bits)</th>
<th>Bypass Capacitor Value (µF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.01</td>
</tr>
<tr>
<td>12</td>
<td>0.01</td>
</tr>
<tr>
<td>14</td>
<td>0.1 to 1.0</td>
</tr>
<tr>
<td>16</td>
<td>0.1 to 1.0</td>
</tr>
<tr>
<td>18</td>
<td>1.0 to 10.0</td>
</tr>
</tbody>
</table>

### Vref value

This parameter sets or displays the reference voltage used by the ADC. If the **Internal 1.024 Volts** reference is used, the value 1.0240 is displayed. If the **Internal Vdda/3** or **Internal Vdda/4** reference option is selected, the value is derived from the Vdda setting in the design-wide resource page. If an external reference is selected, you can enter the reference used to ensure
the counts-to-volts API works properly. The minimum and maximum values that can be entered are 0.9 to 1.3V, respectively. The default is 1.024V. When the selected reference is outside the range of 0.9 to 1.3V, an error symbol at the end of the Vref value numerical drop down menu alerts you that an error condition has occurred. Also, compilation fails if the Vref value exceeds the expected range of 0.9 to 1.3V.

The table below shows the Vref value displayed when different reference options are selected.

<table>
<thead>
<tr>
<th>Reference Selected</th>
<th>Vref Value Displayed</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal 1.024 Volts</td>
<td>1.0240</td>
<td>Value not editable.</td>
</tr>
<tr>
<td>Internal Bypassed on P0.3</td>
<td>1.0240</td>
<td>Value not editable.</td>
</tr>
<tr>
<td>Internal Bypassed on P3.2</td>
<td>1.0240</td>
<td>Value not editable.</td>
</tr>
<tr>
<td>External Vref on P0.3</td>
<td>1.0240</td>
<td>Value is editable with a default value of 1.0240.</td>
</tr>
<tr>
<td>External Vref on P3.2</td>
<td>1.0240</td>
<td>Value is editable with a default value of 1.0240.</td>
</tr>
<tr>
<td>Internal Vdda/4</td>
<td>1.2500*</td>
<td>This value is derived from the Vdda setting in the design-wide resource. Value is not editable. *The example shown is for a Vdda of 5 V.</td>
</tr>
<tr>
<td>Internal Vdda/4 Bypassed on P0.3</td>
<td>1.2500*</td>
<td>This value is derived from the Vdda setting in the design-wide resource. Value is not editable. *The example shown is for a Vdda of 5 V.</td>
</tr>
<tr>
<td>Internal Vdda/4 Bypassed on P3.2</td>
<td>1.2500*</td>
<td>This value is derived from the Vdda setting in the design-wide resource. Value is not editable. *The example shown is for a Vdda of 5 V.</td>
</tr>
<tr>
<td>Internal Vdda/3</td>
<td>1.100**</td>
<td>This value is derived from the Vdda setting in the design-wide resource. Value is not editable. **The example shown is for a Vdda of 3.3 V</td>
</tr>
<tr>
<td>Internal Vdda/3 Bypassed on P0.3</td>
<td>1.100**</td>
<td>This value is derived from the Vdda setting in the design-wide resource. Value is not editable. **The example shown is for a Vdda of 3.3 V</td>
</tr>
<tr>
<td>Internal Vdda/3 Bypassed on P3.2</td>
<td>1.100**</td>
<td>This value is derived from the Vdda setting in the design-wide resource. Value is not editable. **The example shown is for a Vdda of 3.3 V</td>
</tr>
</tbody>
</table>
Common Settings

Hardware SOC (Start of Conversion)

The ADC may be started by firmware with the ADC_StartConvert() function or by triggering with a hardware signal. Checking the Hardware SOC parameter enables an external pin to start conversion. When Hardware SOC is enabled, the pin is displayed on the component; otherwise, no pin is displayed. The conversion starts on the rising edge of the signal on the pin. Conversions continue until ADC_StopConvert() is called. By default, Hardware SOC is disabled. If a conversion is already in process, a Hardware SOC trigger is ignored.

Clock Source

The ADC can be clocked by a source internal to the ADC component, a source external to the component but internal to the chip using a standard clock component or UDB, or by a source external to the chip. The internal or external clock selection is made using a radio button. When external clock is enabled, a clock input pin is displayed on the ADC schematic symbol. External clocks must have 50-percent duty cycle. The internal clock is guaranteed by design to have the correct duty cycle.

Clock stability is important for achieving low noise operation. One of the effects of jitter is substantial spreading of the signal. These are clearly shown in the following FFTs. The signal-to-noise ratio (SNR) of the ADC can be significantly improved with the use of an external clock.
One measure of ADC performance is the distortion caused by INL, DNL, and timing errors. This distortion is measured by driving the ADC with a “perfect” sine wave then evaluating the output using an FFT (fast Fourier transform). Excess INL results in harmonic distortion with “lines” at multiples of the input sine wave frequency. Excess DNL results in an overall increase in the noise floor. Timing errors (frequency jitter) in the clock to the ADC result in spreading of the indicated fundamental from a nominal single line in the FFT. The PSoC internal clock has more frequency jitter than an external crystal oscillator. The upper plot shows the noise and spread in the FFT at the fundamental from the internal clock. The lower plot shows substantial reduction in the spreading of the fundamental as a result of the reduced frequency jitter from the external crystal oscillator.

**Figure 7. Noise versus Clock**

Single-Frequency FFT for Internal Oscillator

![Single-Frequency FFT for Internal Oscillator](image)

Single-Frequency FFT using External Crystal Oscillator

![Single-Frequency FFT using External Crystal Oscillator](image)
**External Charge Pump Clock**

The External Charge Pump Clock option is always enabled to guarantee the optimal charge pump clock frequency. This maximizes ADC performance. Charge pump clock frequency is automatically adjusted for optimal performance, but with the lowest power. This clock is adjusted to be 2 to 4 times that of the ADC clock frequency. This option is not available for user selection and is disabled.

**Enable_Vref_Vssa**

This parameter allows you to connect the negative input of the ADC's reference Vssa to the analog global AGL[6]. For high-precision systems, the Vref_Vssa can be connected to the external Vssa to eliminate any small difference between the on-chip Vssa and the off-chip Vssa. This small difference could cause a gain error in the ADC.

This option is valid only for PSoC 3 Production and PSoC 5 ES2 silicon or later. Selecting this option with PSoC 3 ES2 and PSoC 5 silicon causes a compilation error.

Vref_Vssa is an advanced feature that is useful when using an external reference supplied to the ADC. The Vref_Vssa connection can be routed through the analog routing fabric and brought out to a pin. This enables you to connect to an external reference and eliminate any offset in the reference supplied to the ADC due to I*R drops in the Vssa pin and bonding wire.

The Vref_Vssa is directly connected to Analog Global Left 6 (AGL[6]). (See the analog routing diagram in the device datasheet for more information.) AGL[6] makes direct connections to pins P4[6], P4[2], P0[6] and P0[2]. For the best performance, make sure that Vref_Vssa is connected to one of these pins. Placing Vref_Vssa on another pin causes extra routing resources to be consumed and extra resistance to be added in series with the connection.

The manual analog routing system (MARS) components allow you to and a rule check that ensures that only the specified pins can be used. By placing an Analog Resource Constraint on the Vref_Vssa net, only resources that make direct connections to that net can be used. If you place a pin on that net that is not directly connected to AGL[6], the tool generates an error during the build process. The error indicates that you have connected a resource to that net that has no direct connection to the pin and therefore cannot route. The following graphic is an example:
Resources

The ADC_DelSig uses a decimator, delta-sigma modulator, and a clock source. If an external reference or reference bypass is selected, P0[3] or P3[2] can be used for the external reference or bypass capacitor.

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Datapaths</th>
<th>Macro cells</th>
<th>Status Registers</th>
<th>Control Registers</th>
<th>Counter7</th>
<th>Flash</th>
<th>RAM</th>
<th>Pins (per External I/O)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 to 20 Bits</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4400</td>
<td>14</td>
<td>–</td>
</tr>
</tbody>
</table>

Application Programming Interface

Application Programming Interface (API) routines allow you to configure the component using software. The following table lists and describes the interface to each function. The subsequent sections cover each function in more detail.

By default, PSoC Creator assigns the instance name “ADC_DelSig_1” to the first instance of a component in a given design. You can rename the instance to any unique value that follows the syntactic rules for identifiers. The instance name becomes the prefix of every global function name, variable, and constant symbol. For readability, the instance name used in the following table is “ADC.”

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC_Start()</td>
<td>Sets the initVar variable, calls the ADC_Init() function, and then calls the ADC_Enable() function.</td>
</tr>
<tr>
<td>ADC_Stop()</td>
<td>Stops ADC conversions and powers down.</td>
</tr>
<tr>
<td>ADC_SetBufferGain()</td>
<td>Selects input buffer gain (1,2,4,8)</td>
</tr>
<tr>
<td>ADC_StartConvert()</td>
<td>Starts conversion.</td>
</tr>
<tr>
<td>ADC_StopConvert()</td>
<td>Stops conversions</td>
</tr>
<tr>
<td>ADC_IRQ_Enable()</td>
<td>Enables interrupts at end of conversion.</td>
</tr>
<tr>
<td>ADC_IRQ_Disable()</td>
<td>Disables interrupts.</td>
</tr>
<tr>
<td>ADC_IsEndConversion()</td>
<td>Returns a nonzero value if conversion is complete.</td>
</tr>
<tr>
<td>ADC_GetResult8()</td>
<td>Returns an 8-bit conversion result, right justified.</td>
</tr>
<tr>
<td>ADC_GetResult16()</td>
<td>Returns a 16-bit conversion result, right justified.</td>
</tr>
<tr>
<td>ADC_GetResult32()</td>
<td>Returns a 32-bit conversion result, right justified.</td>
</tr>
<tr>
<td>ADC_SetOffset()</td>
<td>Sets the offset used by the ADC_CountsTo_mVolts(), ADC_CountsTo_uVolts(), and ADC_CountsTo_Volts() functions.</td>
</tr>
<tr>
<td>ADC_SelectConfiguration()</td>
<td>Sets one of up to four ADC configurations.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ADC_SetGain()</td>
<td>Sets the gain used by the ADC_CountsTo_mVolts(), ADC_CountsTo_uVolts(), and ADC_CountsTo_Volts() functions.</td>
</tr>
<tr>
<td>ADC_CountsTo_mVolts()</td>
<td>Converts ADC counts to mV.</td>
</tr>
<tr>
<td>ADC_CountsTo_uVolts()</td>
<td>Converts ADC counts to µV.</td>
</tr>
<tr>
<td>ADC_CountsTo_Volts()</td>
<td>Converts ADC counts to floating point volts.</td>
</tr>
<tr>
<td>ADC_Sleep()</td>
<td>Stops ADC operation and saves the user configuration.</td>
</tr>
<tr>
<td>ADC_Wakeup()</td>
<td>Restores and enables the user configuration.</td>
</tr>
<tr>
<td>ADC_Init()</td>
<td>Initializes or restores the ADC using the Configure dialog settings.</td>
</tr>
<tr>
<td>ADC_Enable()</td>
<td>Enables the ADC.</td>
</tr>
<tr>
<td>ADC_SaveConfig()</td>
<td>Saves the current configuration.</td>
</tr>
<tr>
<td>ADC_RestoreConfig()</td>
<td>Restores the configuration.</td>
</tr>
<tr>
<td>ADC_SetCoherency()</td>
<td>Sets the coherency register.</td>
</tr>
<tr>
<td>ADC_SetGCOR()</td>
<td>Calculates a new GCOR value and sets the GCOR registers with this new value.</td>
</tr>
<tr>
<td>ADC_ReadGCOR()</td>
<td>Returns the normalized GCOR register values.</td>
</tr>
</tbody>
</table>

**Global Variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC_initVar</td>
<td>The ADC_initVar variable indicates whether the ADC has been initialized. The variable is initialized to 0 and set to 1 the first time ADC_Start() is called. This allows the component to restart without reinitialization after the first call to the ADC_Start() routine. If reinitialization of the component is required, then the ADC_Init() function can be called before the ADC_Start() or ADC_Enable() functions.</td>
</tr>
<tr>
<td>ADC_offset</td>
<td>The ADC_offset variable is used to calibrate offset. Initially, this variable is set to zero. Applications can modify it using the ADC_SetOffset() function. It affects only the ADC_CountsTo_Volts(), ADC_CountsTo_mVolts(), and ADC_CountsTo_uVolts() functions by subtracting the given offset.</td>
</tr>
<tr>
<td>ADC_CountsPerVolt</td>
<td>The ADC_countsPerVolt variable is used to calibrate gain. Initially, this variable is calculated for the default ADC configuration. The calculated value depends on resolution, input range, and voltage reference. Applications can modify it using the ADC_SetGain() function. It affects only the ADC_CountsTo_Volts(), ADC_CountsTo_mVolts(), and ADC_CountsTo_uVolts() functions by supplying the correct conversion between ADC counts and the applied input voltage.</td>
</tr>
<tr>
<td>ADC_convDone</td>
<td>The ADC_convDone variable is used as the software flag for checking the ADC conversion in single sample conversion mode for resolutions above 16 bits.</td>
</tr>
</tbody>
</table>
void ADC_Start(void)

Description: Sets the initVar variable, calls the ADC_Init() function, and then calls the ADC_Enable() function.

This function configures and powers up the ADC, but does not start conversions. By default, the ADC is configured for Config1 unless the ADC_SelectConfiguration() function selects an alternate configuration.

Parameters: None

Return Value: None

Side Effects: None

void ADC_Stop(void)

Description: Disables and powers down the ADC.

Note This API is not recommended for use on PSoC 3 ES2 and PSoC 5 silicon. These devices have a defect that causes connections to several analog resources to be unreliable when not powered. The unreliability manifests itself in silent failures (for example, unpredictably bad results from analog components) when the component using that resource stops. When using this silicon, all analog components in a design should be powered up (by calling their respective _Start() APIs, for instance ADC_Start()) at all times. Do not call the ADC_Stop() APIs.

Parameters: None

Return Value: None

Side Effects: None

void ADC_SetBufferGain(uint8 gain)

Description: Sets the input buffer gain.

Parameters: uint8 gain: Input gain setting. The following table shows valid gain constants.

<table>
<thead>
<tr>
<th>Gain Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC_BUF_GAIN_1X</td>
<td>Set input buffer gain to 1</td>
</tr>
<tr>
<td>ADC_BUF_GAIN_2X</td>
<td>Set input buffer gain to 2</td>
</tr>
<tr>
<td>ADC_BUF_GAIN_4X</td>
<td>Set input buffer gain to 4</td>
</tr>
<tr>
<td>ADC_BUF_GAIN_8X</td>
<td>Set input buffer gain to 8</td>
</tr>
</tbody>
</table>

Return Value: None

Side Effects: None
void ADC_StartConvert(void)

Description: Forces the ADC to initiate a conversion. If in Single Sample mode, one conversion is performed then the ADC halts. If in one of the other three conversion modes, the ADC runs continuously.

If the ADC_StartConvert() function is called while the conversion is in progress, the next conversion start is queued and a new conversion will start after finishing the current conversion. If you want to start a new conversion without waiting for the current conversion to finish, then stop the current conversion by calling ADC_StopConvert(). After stopping the conversion, restart the conversion by calling ADC_StartConvert().

Parameters: None
Return Value: None
Side Effects: None

void ADC_StopConvert(void)

Description: Forces the ADC to stop all conversions. If the ADC is in the middle of a conversion, the ADC will be reset and not provide a result for that partial conversion.

Parameters: None
Return Value: None
Side Effects: None

void ADC_IRQ_Enable(void)

Description: Enables interrupts at the end of a conversion. Global interrupts must also be enabled for the ADC interrupts to occur. To enable global interrupts, use the enable global interrupt macro “CYGlobalIntEnable;” in main.c, before interrupts occur.

Parameters: None
Return Value: None
Side Effects: Enables interrupts. Reading the result clears the interrupt.

void ADC_IRQ_Disable(void)

Description: Disables interrupts at the end of a conversion.

Parameters: None
Return Value: None
Side Effects: None
uint8 ADC_IsEndConversion(uint8 retMode)

**Description:** Checks for ADC end of conversion. This function provides the programmer with two options. In one mode this function immediately returns with the conversion status. In the other mode, the function does not return (blocking) until the conversion has completed.

**Parameters:**
- uint8 retMode: Check conversion return mode. See the following table for options.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC_RETURN_STATUS</td>
<td>Immediately returns conversion result status.</td>
</tr>
<tr>
<td>ADC_WAIT_FOR_RESULT</td>
<td>Does not return until ADC conversion is complete.</td>
</tr>
</tbody>
</table>

**Return Value:**
- uint8: If a nonzero value is returned, the last conversion has completed. If the returned value is zero, the ADC is still calculating the last result.

**Side Effects:** None

int8 ADC_GetResult8(void)

**Description:** Returns the result of an 8-bit conversion. If the resolution is set greater than 8 bits, the LSB of the result is returned. When the ADC is configured for 8-bit single-ended mode, use the ADC_GetResult16() function instead. This function returns only signed 8-bit values. The maximum positive signed 8-bit value is 127, but in single-ended 8-bit mode, the maximum positive value is 255.

**Parameters:** None

**Return Value:** int8: The LSB of the last ADC conversion.

**Side Effects:** None

int16 ADC_GetResult16(void)

**Description:** Returns a 16-bit result for a conversion that has a resolution of 8 to 16 bits. If the resolution is set greater than 16 bits, it will return the 16 least significant bits of the result. When the ADC is configured for 16-bit single-ended mode, use the ADC_GetResult32() function instead. This function returns only signed 16-bit results, which allows a maximum positive value of 32767, not 65535.

**Parameters:** None

**Return Value:** int16: The 16-bit result of the last ADC conversion.

**Side Effects:** None
int32 ADC_GetResult32(void)

Description: Returns a 32-bit result for a conversion that has a resolution of 8 to 20 bits.
Parameters: None
Return Value: int32: Result of the last ADC conversion.
Side Effects: None

void ADC_SetOffset(int32 offset)

Description: Sets the ADC offset which is used by the functions ADC_CountsTo_uVolts(), ADC_CountsTo_mVolts(), and ADC_CountsTo_Volts() to subtract the offset from the given reading before calculating the voltage conversion.
Parameters: int32 offset: This value is a measured value when the inputs are shorted or connected to the same input voltage.
Return Value: None.
Side Effects: Affects the ADC_CountsTo_uVolts(), ADC_CountsTo_mVolts(), and ADC_CountsTo_Volts() functions by subtracting the given offset.

void ADC_SetGain(int32 adcGain)

Description: Sets the ADC gain in counts per volt for the voltage conversion functions below. This value is set by default by the reference and input range settings. It should only be used to further calibrate the ADC with a known input or if an external reference is used.
Parameters: int32 adcGain: ADC gain in counts per volt.
Return Value: None.
Side Effects: Affects only the ADC_CountsTo_uVolts(), ADC_CountsTo_mVolts(), and ADC_CountsTo_Volts() functions by supplying the correct conversion between ADC counts and voltage.
void ADC_SelectConfiguration(uint8 config uint8 restart)

Description: Sets one of up to four ADC configurations. Before setting the new configuration, stop and power down the ADC. After setting the new configuration, the ADC can be powered and conversion can be restarted depending up on the value of second parameter restart. If the value of this parameter is 1, then ADC will be restarted. If this value is zero, then you must call ADC_Start() and ADC_StartConvert() to restart the conversion.

Parameters:  
uint8 config: Configuration option between 1 and 4.  
uint8 restart: Restart option. 1 means start the ADC and restart the conversion. 0 means do not start the ADC and conversion.

Return Value: None.

Side Effects: None

int32 ADC_CountsTo_mVolts(int32 adcCounts)

Description: Converts the ADC output to mV as a 32-bit integer. For example, if the ADC measured 0.534 V, the return value would be 534 mV.

Parameters: int32 adcCounts: Result from the ADC conversion.

Return Value: int32: Result in mV.

Side Effects: None

int32 ADC_CountsTo_uVolts(int32 adcCounts)

Description: Converts the ADC output to µV as a 32-bit integer. For example, if the ADC measured –0.02345 V, the return value would be –23450 µV.

Parameters: int32 adcCounts: Result from the ADC conversion.

Return Value: int32: Result in µV.

Side Effects: None

float ADC_CountsTo_Volts(int32 adcCounts)

Description: Converts the ADC output to volts as a floating point number. For example, if the ADC measures a voltage of 1.2345 V, the returned result would be +1.2345 V.

Parameters: int32 adcCounts: Result from the ADC conversion.

Return Value: Float: Result of the last ADC conversion.

Side Effects: None
void ADC_Sleep(void)
Description: The ADC_Sleep() function checks to see if the component is enabled and saves that state. Then it calls the ADC_Stop() function and calls ADC_SaveConfig() to save the user configuration.
Call the ADC_Sleep() function before calling the CyPmSleep() or the CyPmHibernate() function. Refer to the PSoC Creator System Reference Guide for more information about power management functions.
Parameters: None
Return Value: None
Side Effects: Note If you put the ADC hardware to sleep using the CyPmSleep() function, then after coming out of sleep, ADC_Start() and ADC_StartConvert() must be executed to restart conversions.

void ADC_Wakeup(void)
Description: The ADC_Wakeup() function calls the ADC_RestoreConfig() function to restore the user configuration. If the component was enabled before the ADC_Sleep() function was called, the ADC_Wakeup() function will re-enable the component.
Parameters: None
Return Value: None
Side Effects: Calling the ADC_Wakeup() function without first calling the ADC_Sleep() or ADC_SaveConfig() function may produce unexpected behavior.

void ADC_Init(void)
Description: Initializes or restores the component parameters per the Configure dialog settings. You are not required to call this function if ADC_Start() is called.
Parameters: None
Return Value: None
Side Effects: All registers will be reset to their initial values. This reinitializes the component.

void ADC_Enable(void)
Description: Enables the clock and power for ADC.
Parameters: None
Return Value: None
Side Effects: None
void ADC_SaveConfig(void)

Description: This function saves the component configuration. This will save nonretention registers. This function will also save the current component parameter values, as defined in the Configure dialog or as modified by appropriate APIs. This function is called by the ADC_Sleep() function.

Parameters: None

Return Value: None

Side Effects: None.

void ADC_RestoreConfig(void)

Description: This function restores the component configuration. This will restore nonretention registers. This function will also restore the component parameter values to what they were prior to calling the ADC_Sleep() function.

Parameters: None

Return Value: None

Side Effects: Calling this function without first calling the ADC_Sleep() or ADC_SaveConfig() function may produce unexpected behavior.

void ADC_SetCoherency(uint8 coherency)

Description: This function allows you to change which of the ADC’s three-word results will trigger a coherency unlock. The ADC’s result will not be updated until the set byte is read by either the ADC or DMA. By default, the LSB is the coherency byte. If DMA or if a custom API is written where the LSB is not the last byte read, this use this API to set the last byte of the ADC result that is read. If a multi-byte read is performed either by DMA or the ARM processor, the coherency can be set to any byte in the last word read.

Parameters: 

uint8 coherency: Coherency settings. The following table shows the valid coherency values.

<table>
<thead>
<tr>
<th>Coherency Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC_COHER_LOW</td>
<td>The LSB should be the last byte read.</td>
</tr>
<tr>
<td>ADC_COHER_MID</td>
<td>The middle byte should be the last byte read.</td>
</tr>
<tr>
<td>ADC_COHER_HIGH</td>
<td>The MSB should be the last byte read.</td>
</tr>
</tbody>
</table>

Return Value: None

Side Effects: If the coherency is changed, for any reason, it should be changed back to the LSB when the provided ADC_GetResult API is used.
uint8 ADC_SetGCOR(float gainAdjust)

Description: This function calculates a new GCOR (ADC Gain) value and writes it into the GCOR registers. The GCOR value is a 16-bit value that represents a gain of 0 to 2. The ADC result is multiplied by this value before it is placed in the ADC output registers. When executing the function, the old GCOR value is multiplied by gainAdjust input and reloaded into the GCOR register. The GCOR value is normalized based on the GVAL register.

Parameters: float gainAdjust: Valid range for this value is 0.000 to 1.999.

Return Value: uint8: A nonzero value will be returned, if the correction value is outside the GCOR value range of 0.00 to 1.9999.

Side Effects: Actual GCOR value depends on the GVAL register. The GVAL register is set to a value that is equal to number of valid bits in the GCOR register minus one. If GVAL is 15 (0x0F), all 16 bits of the GCOR registers are valid. If GVAL is 11 (0x0B), only 12 bits are valid. The least four bits are lost when the GCOR value is shifted four places to the right.

uint16 ADC_ReadGCOR(void)

Description: This function returns the current GCOR register value, normalized based on the GVAL setting. For example, if the GCOR value is 0x0812 and the GVAL register is set to 11 (0x0B), the returned value is shifted by four bits to the left. (Actual GCOR value = 0x0812, returned value = 0x8120)

Parameters: None

Return Value: uint16: Normalized GCOR value

Side Effects: The GCOR value depends on the GVAL register. If GVAL is set to less than 15 (0x0F), the GCOR value is shifted right by 15 minus GVAL.
DMA Information

The DMA component can be used to transfer converted results from the ADC_DelSig register to RAM or another component, such as the Digital Filter Block (DFB). The DMA data request signal (DRQ) should be connected to the EOC pin from the ADC. You can use the DMA Wizard to configure DMA operation as follows:

<table>
<thead>
<tr>
<th>Name of DMA Source/Destination in DMA Wizard</th>
<th>Direction</th>
<th>DMA Req Signal</th>
<th>DMA Req Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC_DelSig_DEC_SAMP_PTR</td>
<td>source</td>
<td>EOC</td>
<td>Edge</td>
<td>Receives 1-byte conversion result for input analog value that has a resolution of 8 bits.</td>
</tr>
<tr>
<td>ADC_DelSig_DEC_SAMPM_PTR</td>
<td>source</td>
<td>EOC</td>
<td>Edge</td>
<td>Receives 2-byte conversion result for input analog value that has a resolution of 9 to 16 bits.</td>
</tr>
<tr>
<td>ADC_DelSig_DEC_SAMPH_PTR</td>
<td>source</td>
<td>EOC</td>
<td>Edge</td>
<td>Receives 3-byte conversion result for input analog value that has a resolution of 17 to 20 bits.</td>
</tr>
</tbody>
</table>

Sample Firmware Source Code

PSOC Creator provides many example projects that include schematics and example code in the Find Example Project dialog. For component-specific examples, open the dialog from the Component Catalog or an instance of the component in a schematic. For general examples, open the dialog from the Start Page or File menu. As needed, use the Filter Options in the dialog to narrow the list of projects available to select.

Refer to the “Find Example Project” topic in the PSoC Creator Help for more information.

Functional Description

The delta-sigma channel contains the following blocks:

- A high-input-impedance, front-end buffer (with programmable gain) that can be bypassed (and powered down) when not needed.
- A fully differential programmable third-order switched capacitor modulator.
- A downstream digital-filtering consisting of: a fourth-order Cascaded Integrator-Comb (CIC) filter (also called the decimator) and the postprocessing engine (cicdec4_pproc), which optionally performs gain, offset, and simple FIR filtering functions on the data as it leaves the CIC filter.
- ANAIF - Analog interface logic block consists of the register control for the input buffer and the modulator.
Without an input buffer, a switched-capacitor input stage consumes current to charge the capacitor during each cycle. In that case, the equivalent input resistance is of the order of $1/(fs \times C)$, or $1/((3 \text{ MHz}) \times (5 \text{ pF})) = 66 \text{ k}\Omega$. Many sensor applications require a much higher impedance to achieve an accurate reading. Therefore, an input buffer is made a part of the delta sigma channel.

The input buffer must also handle signals closer to ground in some applications and must work closer to the supply rail in others. Input buffer architecture contains two single-ended buffers, which create a differential channel. You can select either buffer for the channel. When the channel operates in a single-ended mode, one of the inputs is connected to ground rail, and the corresponding buffer is bypassed. You can power down the buffers individually. There are two main modes of operation for the buffer:

- **Level-Shifted Mode**: Buffer output can be level shifted up from the input when the input is close to 0-V input common mode voltage range.
- **Rail to Rail Mode**: This mode is used when input is rail to rail.

**Figure 8. ADC Buffer Structure**

The switched capacitor implementation is shown in Figure 9. A dynamic element matching (DEM) block shapes the errors caused by mismatch in the switched-capacitor DAC feedback of the modulator, when operating under 9-level quantization. When the buffer and modulator are correctly configured for a given application, the variable-level programmable quantizer (level 2, 3, or 9) in the modulator produces “the-quantized” bit stream. This quantized bit stream is 8 bits wide and in thermometric format. The conversion of thermometer quantizer code into 2’s complement format, for use in the decimator (Sinc4 and Sinc1), is performed in the ANAIF block.
Figure 9. Switched Capacitor Delta Sigma Modulator Structure

Registers

Sample Registers
The ADC results can be between 8 and 24 bits of resolution. The output is divided into three 8-bit registers. The CPU or DMA can access these registers to read the ADC result.

ADC_DEC_SAMP (ADC Output Data Sample Low Register)

<table>
<thead>
<tr>
<th>Bits</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>Data[7:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ADC_DEC_SAMPM (ADC Output Data Sample Middle Register)

<table>
<thead>
<tr>
<th>Bits</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>Data[15:8]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ADC_DEC_SAMPH (ADC Output Data Sample High Register)

<table>
<thead>
<tr>
<th>Bits</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>Data[23:16]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DC and AC Electrical Characteristics for PSoC 3

The following values indicate expected performance and are based on initial characterization data. Unless otherwise specified, operating conditions are:

- Operation in continuous sample mode
- \( f_{clk} = 3.072 \text{ MHz} \) for resolution = 16 to 20 bits; \( f_{clk} = 6.144 \text{ MHz} \) for resolution = 8 to 15 bits
- Reference = 1.024-V internal reference bypassed on P3[2] or P0[3]
- Unless otherwise specified, all charts and graphs show typical value

Delta-Sigma ADC DC Specifications for PSoC 3

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Conditions</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution</td>
<td>Number of channels, single ended</td>
<td></td>
<td>8</td>
<td>–</td>
<td>20</td>
<td>bits</td>
</tr>
<tr>
<td>Number of channels, differential</td>
<td>Number of channels, differential</td>
<td>Differential pair is formed using a pair of GPIOs.</td>
<td>–</td>
<td>–</td>
<td>No. of GPIO/2</td>
<td>–</td>
</tr>
<tr>
<td>Monotonic</td>
<td>Monotonic</td>
<td>Yes</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Ge</td>
<td>Gain error (^3)</td>
<td>Buffered, buffer gain = 1, input range = ±1.024 V, 16-bit mode</td>
<td>–</td>
<td>–</td>
<td>±0.2</td>
<td>%</td>
</tr>
<tr>
<td>Gd</td>
<td>Gain drift</td>
<td>Buffered, buffer gain = 1, input range = ±1.024 V, 16-bit mode</td>
<td>–</td>
<td>–</td>
<td>50</td>
<td>ppm/°C</td>
</tr>
<tr>
<td>( V_{OS} )</td>
<td>Input offset voltage (^4)</td>
<td>Buffered</td>
<td>–</td>
<td>–</td>
<td>±0.1</td>
<td>mV</td>
</tr>
<tr>
<td>TCV( V_{OS} )</td>
<td>ADC TC input offset voltage</td>
<td>Temperature coefficient, input offset voltage</td>
<td>–</td>
<td>–</td>
<td>55</td>
<td>µV/°C</td>
</tr>
<tr>
<td>Input voltage range, single ended</td>
<td>Input voltage range, single ended</td>
<td></td>
<td>( V_{SSA} ) –</td>
<td>( V_{DDA} )</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

\(^{3}\) Total gain error is sum of ADC error and buffer error.

\(^{4}\) Total offset voltage error is sum of buffer \( V_{OS} \) and ADC \( V_{OS} \).

\(^{5}\) \( V_{SSA} \) to \( 6 \times V_{BG} \) range is used for direct measurement of \( V_{DDA} \) power supply. Actual scale is limited to \( V_{DDA} \).
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Conditions</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input voltage range, differential</td>
<td>Unbuffered</td>
<td>( V_{SSA} ) – ( V_{DDA} )</td>
<td>V</td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Input voltage range, differential</td>
<td>Buffered</td>
<td>( V_{SSA} ) – ( V_{DDA} - 1 )</td>
<td>V</td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>PSRRb</td>
<td>Power supply rejection ratio, buffered</td>
<td>Buffer gain = 1, 16-bit, Range = ( \pm 1.024 ) V</td>
<td>90</td>
<td>–</td>
<td>–</td>
<td>dB</td>
</tr>
<tr>
<td>CMRRb</td>
<td>Common mode rejection ratio, buffered</td>
<td>Buffer gain = 1, 16-bit, Range = ( \pm 1.024 ) V</td>
<td>85</td>
<td>–</td>
<td>–</td>
<td>dB</td>
</tr>
<tr>
<td>INL20</td>
<td>Integral nonlinearity</td>
<td>Range = ( \pm 1.024 ) V, unbuffered</td>
<td>–</td>
<td>–</td>
<td>( \pm 32 )</td>
<td>LSB</td>
</tr>
<tr>
<td>DNL20</td>
<td>Differential nonlinearity</td>
<td>Range = ( \pm 1.024 ) V, unbuffered</td>
<td>–</td>
<td>–</td>
<td>( \pm 1 )</td>
<td>LSB</td>
</tr>
<tr>
<td>INL16</td>
<td>Integral nonlinearity</td>
<td>Range = ( \pm 1.024 ) V, unbuffered</td>
<td>–</td>
<td>–</td>
<td>( \pm 2 )</td>
<td>LSB</td>
</tr>
<tr>
<td>DNL16</td>
<td>Differential nonlinearity</td>
<td>Range = ( \pm 1.024 ) V, unbuffered</td>
<td>–</td>
<td>–</td>
<td>( \pm 1 )</td>
<td>LSB</td>
</tr>
<tr>
<td>INL12</td>
<td>Integral nonlinearity</td>
<td>Range = ( \pm 1.024 ) V, unbuffered</td>
<td>–</td>
<td>–</td>
<td>( \pm 1 )</td>
<td>LSB</td>
</tr>
<tr>
<td>DNL12</td>
<td>Differential nonlinearity</td>
<td>Range = ( \pm 1.024 ) V, unbuffered</td>
<td>–</td>
<td>–</td>
<td>( \pm 1 )</td>
<td>LSB</td>
</tr>
<tr>
<td>INL8</td>
<td>Integral nonlinearity</td>
<td>Range = ( \pm 1.024 ) V, unbuffered</td>
<td>–</td>
<td>–</td>
<td>( \pm 1 )</td>
<td>LSB</td>
</tr>
<tr>
<td>DNL8</td>
<td>Differential nonlinearity</td>
<td>Range = ( \pm 1.024 ) V, unbuffered</td>
<td>–</td>
<td>–</td>
<td>( \pm 1 )</td>
<td>LSB</td>
</tr>
<tr>
<td>Rin_Buff</td>
<td>ADC input resistance</td>
<td>Input buffer used</td>
<td>10</td>
<td>–</td>
<td>–</td>
<td>MΩ</td>
</tr>
<tr>
<td>Rin_ADC16</td>
<td>ADC input resistance</td>
<td>Input buffer bypassed, 16-bit, Range = ( \pm 1.024 ) V</td>
<td>–</td>
<td>74</td>
<td>–</td>
<td>kΩ</td>
</tr>
<tr>
<td>Rin_ADC12</td>
<td>ADC input resistance</td>
<td>Input buffer bypassed, 12 bit, Range = ( \pm 1.024 ) V</td>
<td>–</td>
<td>148</td>
<td>–</td>
<td>kΩ</td>
</tr>
<tr>
<td>Cin_G1</td>
<td>ADC input capacitance</td>
<td>Gain = 1</td>
<td>–</td>
<td>5</td>
<td>–</td>
<td>pF</td>
</tr>
<tr>
<td>( V_{EXTREF} )</td>
<td>ADC external reference input voltage</td>
<td>Pins P0[3], P3[2]</td>
<td>0.9</td>
<td>–</td>
<td>1.3</td>
<td>V</td>
</tr>
</tbody>
</table>

**Current Consumption**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Conditions</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I_{DD,20} )</td>
<td>Current consumption, 20-bit</td>
<td>187 sps, unbuffered</td>
<td>–</td>
<td>–</td>
<td>1.25</td>
<td>mA</td>
</tr>
<tr>
<td>( I_{DD,16} )</td>
<td>Current consumption, 16-bit</td>
<td>48 ksp, unbuffered</td>
<td>–</td>
<td>–</td>
<td>1.2</td>
<td>mA</td>
</tr>
<tr>
<td>( I_{DD,12} )</td>
<td>Current consumption, 12-bit</td>
<td>192 ksp, unbuffered</td>
<td>–</td>
<td>–</td>
<td>1.4</td>
<td>mA</td>
</tr>
<tr>
<td>( I_{BUFF} )</td>
<td>Buffer current consumption</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>2.5</td>
<td>mA</td>
</tr>
</tbody>
</table>

---

6 Based on device characterization (not production tested).

7 Using switched capacitors at the ADC input creates an effective input resistance. Holding the gain and number of bits constant, the resistance is proportional to the inverse of the clock frequency. This value is calculated, not measured. For more information see the Technical Reference Manual.
## Delta-Sigma ADC AC Specifications for PSoC 3

| Parameter                          | Description                                           | Conditions                                      | Min  | Typ  | Max  | Units |
|------------------------------------|-------------------------------------------------------|                                                |      |      |      |       |
| Startup time                       |                                                       | Buffer gain = 1, 16 bit, Range = ±1.024 V       | –    | –    | 4    | Samples |
| THD                                | Total harmonic distortion 8                           |                                                 | –    | –    | 0.0032 | %     |

### 20-Bit Resolution Mode

| SR20                               | Sample rate 8                                        | Range = ±1.024 V, unbuffered                    | 7.8  | –    | 187  | sps   |
| BW20                               | Input bandwidth at max sample rate 8                 | Range = ±1.024 V, unbuffered                    | –    | 40   | –    | Hz    |

### 16-Bit Resolution Mode

| SR16                               | Sample rate 8                                        | Range = ±1.024 V, unbuffered                    | 2    | –    | 48   | kspss |
| BW16                               | Input bandwidth at max sample rate 8                 | Range = ±1.024 V, unbuffered                    | –    | 11   | –    | kHz   |
| SINAD16int                          | Signal-to-noise ratio, 16-bit, internal reference 8  | Range = ±1.024 V, unbuffered                    | 81   | –    | –    | dB    |
| SINAD16ext                          | Signal-to-noise ratio, 16-bit, external reference 8  | Range = ±1.024 V, unbuffered                    | 84   | –    | –    | dB    |

### 12-Bit Resolution Mode

| SR12                               | Sample rate, continuous, high power 8                | Range = ±1.024 V, unbuffered                    | 4    | –    | 192  | kspss |
| BW12                               | Input bandwidth at max sample rate 8                 | Range = ±1.024 V, unbuffered                    | –    | 44   | –    | kHz   |
| SINAD12int                          | Signal-to-noise ratio, 12-bit, internal reference 8  | Range = ±1.024 V, unbuffered                    | 66   | –    | –    | dB    |

### 8-Bit Resolution Mode

| SR8                                | Sample rate, continuous, high power 8                | Range = ±1.024 V, unbuffered                    | 8    | –    | 384  | kspss |
| BW8                                | Input bandwidth at max sample rate 8                 | Range = ±1.024 V, unbuffered                    | –    | 88   | –    | kHz   |
| SINAD8int                           | Signal-to-noise ratio, 8-bit, internal reference 8   | Range = ±1.024 V, unbuffered                    | 43   | –    | –    | dB    |

---

* Based on device characterization (not production tested).
### Delta-Sigma ADC Sample Rates, Range = ±1.024 V

<table>
<thead>
<tr>
<th>Resolution, Bits</th>
<th>Continuous</th>
<th>Multi-Sample</th>
<th>Multi-Sample Turbo</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min</td>
<td>Max</td>
<td>Min</td>
</tr>
<tr>
<td>8</td>
<td>8000</td>
<td>384000</td>
<td>1911</td>
</tr>
<tr>
<td>9</td>
<td>6400</td>
<td>307200</td>
<td>1543</td>
</tr>
<tr>
<td>10</td>
<td>5566</td>
<td>267130</td>
<td>1348</td>
</tr>
<tr>
<td>11</td>
<td>4741</td>
<td>227555</td>
<td>1154</td>
</tr>
<tr>
<td>12</td>
<td>4000</td>
<td>192000</td>
<td>978</td>
</tr>
<tr>
<td>13</td>
<td>3283</td>
<td>157538</td>
<td>806</td>
</tr>
<tr>
<td>14</td>
<td>2783</td>
<td>133565</td>
<td>685</td>
</tr>
<tr>
<td>15</td>
<td>2371</td>
<td>113777</td>
<td>585</td>
</tr>
<tr>
<td>16</td>
<td>2000</td>
<td>48000</td>
<td>495</td>
</tr>
<tr>
<td>17</td>
<td>500</td>
<td>12000</td>
<td>124</td>
</tr>
<tr>
<td>18</td>
<td>125</td>
<td>3000</td>
<td>31</td>
</tr>
<tr>
<td>19</td>
<td>16</td>
<td>375</td>
<td>4</td>
</tr>
<tr>
<td>20</td>
<td>8</td>
<td>187.5</td>
<td>2</td>
</tr>
</tbody>
</table>

### Figures

**Delta-Sigma ADC I<sub>DD</sub> versus sps, Unbuffered**

**Delta-Sigma ADC I<sub>DD</sub> versus sps, Buffered**
Delta-Sigma ADC Noise Histogram, 1000 Samples, 20-Bit, 187 sps, Ext Ref, $V_{IN} = V_{REF}/2$, Range = ±1.024 V

![Graph of Delta-Sigma ADC Noise Histogram](image1)

Delta-Sigma ADC Noise Histogram, 1000 samples, 16-bit, 48 ksps, Ext Ref, $V_{IN} = V_{REF}/2$, Range = ±1.024 V

![Graph of Delta-Sigma ADC Noise Histogram](image2)

Delta-Sigma ADC Noise Histogram, 1000 samples, 16-bit, 48 ksps, Int Ref, $V_{IN} = V_{REF}/2$, Range = ±1.024 V

![Graph of Delta-Sigma ADC Noise Histogram](image3)

Delta-Sigma ADC INL versus Resolution at Max Sample Rate

![Graph of Delta-Sigma ADC INL versus Resolution](image4)
### Delta-Sigma ADC RMS Noise versus Input Range and Sample Rate, 20-bit, External Reference, Single-Ended

<table>
<thead>
<tr>
<th>RMS Noise, Counts</th>
<th>Input Voltage Range</th>
<th>Sample rate, sps</th>
<th>0 to V_{REF}</th>
<th>0 to V_{REF} × 2</th>
<th>V_{SSA} to V_{DDA}</th>
<th>0 to V_{REF} × 6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>1.28</td>
<td>1.24</td>
<td>6.02</td>
<td>0.97</td>
</tr>
<tr>
<td></td>
<td></td>
<td>23</td>
<td>1.33</td>
<td>1.28</td>
<td>6.09</td>
<td>0.98</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45</td>
<td>1.77</td>
<td>1.26</td>
<td>6.28</td>
<td>0.96</td>
</tr>
<tr>
<td></td>
<td></td>
<td>90</td>
<td>1.65</td>
<td>0.91</td>
<td>6.84</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td></td>
<td>187</td>
<td>1.87</td>
<td>1.06</td>
<td>7.97</td>
<td>1.01</td>
</tr>
</tbody>
</table>

### Delta-Sigma ADC RMS Noise versus Input Range and Sample Rate, 20-bit, External Reference, Differential

<table>
<thead>
<tr>
<th>RMS Noise, Counts</th>
<th>Input Voltage Range</th>
<th>Sample rate, sps</th>
<th>±V_{REF}</th>
<th>±V_{REF}/2</th>
<th>±V_{REF}/4</th>
<th>±V_{REF}/8</th>
<th>±V_{REF}/16</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>0.70</td>
<td>0.84</td>
<td>1.02</td>
<td>1.40</td>
<td>2.65</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11.3</td>
<td>0.69</td>
<td>0.86</td>
<td>0.96</td>
<td>1.40</td>
<td>2.69</td>
</tr>
<tr>
<td></td>
<td></td>
<td>22.5</td>
<td>0.73</td>
<td>0.82</td>
<td>1.25</td>
<td>1.77</td>
<td>2.67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45</td>
<td>0.76</td>
<td>0.94</td>
<td>1.02</td>
<td>1.76</td>
<td>2.75</td>
</tr>
<tr>
<td></td>
<td></td>
<td>61</td>
<td>0.75</td>
<td>1.01</td>
<td>1.13</td>
<td>1.65</td>
<td>2.98</td>
</tr>
<tr>
<td></td>
<td></td>
<td>170</td>
<td>0.75</td>
<td>0.98</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>187</td>
<td>0.73</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Invalid Operating Region*
Delta Sigma Analog to Digital Converter (ADC_DelSig)  
PSOC® Creator™ Component Datasheet

Delta-Sigma ADC INL versus Output Code, 16-bit, 48 ksps, 25 °C V_{DDA} = 3.3 V

Delta-Sigma ADC DNL versus Output Code, 16-bit, 48 ksps, 25 °C V_{DDA} = 3.3 V

Input Capacitance Values (in pF) Used by Each of the Input Ranges for Resolutions 8 to 20

<table>
<thead>
<tr>
<th>Input Range</th>
<th>Input Capacitance in pF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Resolution 8 to 15</td>
</tr>
<tr>
<td>V_{SS} to V_{REF} (Single)</td>
<td>0.896</td>
</tr>
<tr>
<td>V_{SS} to V_{REF}*2 (Single)</td>
<td>0.896</td>
</tr>
<tr>
<td>V_{SS} to V_{DD} (V_{DD}/3 Single)</td>
<td>1.088</td>
</tr>
<tr>
<td>V_{SS} to V_{DD} (V_{DD}/4 Single)</td>
<td>1.088</td>
</tr>
<tr>
<td>V_{SS} to V_{REF}*6 (Single)</td>
<td>0.992</td>
</tr>
<tr>
<td>± V_{REF} (Diff)</td>
<td>0.896</td>
</tr>
<tr>
<td>± V_{REF}*2 (Diff)</td>
<td>1.088</td>
</tr>
<tr>
<td>± V_{REF}*6 (Diff)</td>
<td>0.992</td>
</tr>
<tr>
<td>± V_{REF}/2 (Diff)</td>
<td>1.600</td>
</tr>
<tr>
<td>± V_{REF}/4 (Diff)</td>
<td>2.800</td>
</tr>
<tr>
<td>± V_{REF}/8 (Diff)</td>
<td>3.488</td>
</tr>
<tr>
<td>± V_{REF}/16 (Diff)</td>
<td>5.696</td>
</tr>
</tbody>
</table>
DC and AC Electrical Characteristics for PSoC 5

The following values indicate expected performance and are based on initial characterization data. Unless otherwise specified, operating conditions are:

- Operation in continuous sample mode
- \( f_{clk} = 3.072 \text{ MHz for resolution} = 16 \text{ to } 20 \text{ bits}; f_{clk} = 6.144 \text{ MHz for resolution} = 8 \text{ to } 15 \text{ bits} \)
- Reference = 1.024-V internal reference bypassed on P3[2] or P0[3]
- Unless otherwise specified, all charts and graphs show typical value

Delta-Sigma ADC DC Specifications for PSoC 5

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Conditions</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution</td>
<td></td>
<td>8</td>
<td>–</td>
<td>20</td>
<td></td>
<td>bits</td>
</tr>
<tr>
<td>Number of channels, single-ended</td>
<td></td>
<td>( - )</td>
<td>( - )</td>
<td>( - )</td>
<td>( \text{No. of GPIOs} )</td>
<td></td>
</tr>
<tr>
<td>Number of channels, differential</td>
<td></td>
<td>Differential pair is formed using a pair of GPIOs</td>
<td>( - )</td>
<td>( - )</td>
<td>( \text{No. of GPIOs/2} )</td>
<td></td>
</tr>
<tr>
<td>Monotonic</td>
<td></td>
<td>Yes</td>
<td>( - )</td>
<td>( - )</td>
<td>( - )</td>
<td></td>
</tr>
<tr>
<td>( G_e )</td>
<td>Gain error</td>
<td>Level shift buffered, buffer gain = 1, differential range ±1.024 V, 25 °C, 16-bit</td>
<td>(-0.6)</td>
<td>( - )</td>
<td>( 0.6 )</td>
<td>%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unbuffered, differential range ±1.024 V, 25 °C, 16-bit</td>
<td>(-0.3)</td>
<td>( - )</td>
<td>( 0.3 )</td>
<td>%</td>
</tr>
<tr>
<td>( G_d )</td>
<td>Gain drift</td>
<td>Level shift buffered, buffer gain = 1, range = ±1.024 V, 16-bit</td>
<td>( - )</td>
<td>( - )</td>
<td>( 60 )</td>
<td>ppm/°C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unbuffered, range = ±1.024 V, 16-bit</td>
<td>( - )</td>
<td>( - )</td>
<td>( 70 )</td>
<td>ppm/°C</td>
</tr>
<tr>
<td>( V_{OS} )</td>
<td>Input offset voltage</td>
<td>Level shift buffered, 16-bit mode, ( V_{DDA} = 2.7 \text{ V, 25 °C} )</td>
<td>(-650)</td>
<td>( - )</td>
<td>( 650 )</td>
<td>( \mu \text{V} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unbuffered, 16-bit mode, ( V_{DDA} = 2.7 \text{ V, 25 °C} )</td>
<td>(-200)</td>
<td>( - )</td>
<td>( 200 )</td>
<td>( \mu \text{V} )</td>
</tr>
<tr>
<td>( TCV_{OS} )</td>
<td>Temperature coefficient, input offset voltage</td>
<td>Level shift buffered, buffer gain = 1, 16-bit, range = ±1.024 V, ( V_{DDA} &lt; 3.6 \text{ V} )</td>
<td>( - )</td>
<td>( 10 )</td>
<td>( - )</td>
<td>( \mu \text{V/°C} )</td>
</tr>
</tbody>
</table>
### Delta Sigma Analog to Digital Converter (ADC_DelSig)

#### PSoC® Creator™ Component Datasheet

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Conditions</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Unbuffered, 16-bit, range = ±1.024 V</td>
<td>–</td>
<td>–</td>
<td>0.45</td>
<td>µV/°C</td>
</tr>
<tr>
<td>Input voltage range</td>
<td>Single-ended, unbuffered</td>
<td>$V_{SSA}$</td>
<td>–</td>
<td>$V_{DDA}$</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Differential, unbuffered</td>
<td>$V_{SSA}$</td>
<td>–</td>
<td>$V_{DDA}$</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Differential, level shift buffered</td>
<td>$V_{SSA}$</td>
<td>–</td>
<td>$V_{DDA} – 1$</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>PSRRb</td>
<td>Power supply rejection ratio - 16-bit</td>
<td>Level shift buffered, buffer gain = 1, range = ±1.024 V, $V_{DDA} &lt; 3.6$ V</td>
<td>–</td>
<td>80</td>
<td>–</td>
<td>dB</td>
</tr>
<tr>
<td></td>
<td>Unbuffered, range = ±1.024 V</td>
<td>90</td>
<td>–</td>
<td>–</td>
<td>dB</td>
<td></td>
</tr>
<tr>
<td>CMRRb</td>
<td>Common mode rejection ratio, 16-bit</td>
<td>Level shift buffered, buffer gain = 1, Range = ±1.024 V, $V_{DDA} &lt; 3.6$ V</td>
<td>–</td>
<td>85</td>
<td>–</td>
<td>dB</td>
</tr>
<tr>
<td></td>
<td>Unbuffered, range = ±1.024 V</td>
<td>85</td>
<td>–</td>
<td>–</td>
<td>dB</td>
<td></td>
</tr>
<tr>
<td>INL20</td>
<td>INL for 20-bit</td>
<td>Differential range ±1.024 V</td>
<td>–</td>
<td>–</td>
<td>32</td>
<td>LSB</td>
</tr>
<tr>
<td>DNL20</td>
<td>DNL for 20-bit</td>
<td>Differential range ±1.024 V</td>
<td>–</td>
<td>–</td>
<td>1</td>
<td>LSB</td>
</tr>
<tr>
<td>INL16</td>
<td>Integral nonlinearity, 16-bit</td>
<td>Differential range ±1.024 V</td>
<td>–</td>
<td>–</td>
<td>2</td>
<td>LSB</td>
</tr>
<tr>
<td>DNL16</td>
<td>Differential nonlinearity 16-bit</td>
<td>Differential range ±1.024 V</td>
<td>–</td>
<td>–</td>
<td>1</td>
<td>LSB</td>
</tr>
<tr>
<td>INL12</td>
<td>INL for 12-bit</td>
<td>Differential range ±1.024 V</td>
<td>–</td>
<td>–</td>
<td>1</td>
<td>LSB</td>
</tr>
<tr>
<td>DNL12</td>
<td>DNL for 12-bit</td>
<td>Differential range ±1.024 V</td>
<td>–</td>
<td>–</td>
<td>1</td>
<td>LSB</td>
</tr>
<tr>
<td>INL8</td>
<td>INL for 8-bit</td>
<td>Differential range ±1.024 V</td>
<td>–</td>
<td>–</td>
<td>1</td>
<td>LSB</td>
</tr>
<tr>
<td>DNL8</td>
<td>DNL for 8-bit</td>
<td>Differential range ±1.024 V</td>
<td>–</td>
<td>–</td>
<td>1</td>
<td>LSB</td>
</tr>
<tr>
<td>Rin_Buff</td>
<td>ADC input resistance</td>
<td>Input buffer used</td>
<td>10</td>
<td>–</td>
<td>–</td>
<td>MΩ</td>
</tr>
<tr>
<td>Rin_ADC_16</td>
<td>ADC input resistance</td>
<td>Input buffer bypassed, 16-bit, range = ±1.024 V</td>
<td>–</td>
<td>74</td>
<td>–</td>
<td>kΩ</td>
</tr>
<tr>
<td>Rin_ADC_12</td>
<td>ADC input resistance</td>
<td>Input buffer bypassed, 12-bit, range = ±1.024 V</td>
<td>–</td>
<td>148</td>
<td>–</td>
<td>kΩ</td>
</tr>
</tbody>
</table>

9 Based on device characterization (not production tested).
10 Using switched capacitors at the ADC input creates an effective input resistance. Holding the gain and number of bits constant, the resistance is proportional to the inverse of the clock frequency. This value is calculated, not measured. For more information, see the Technical Reference Manual.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Conditions</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>V(_{\text{EXTREF}})</td>
<td>ADC external reference input voltage, see also internal reference in Voltage Reference Specifications</td>
<td>Pins P0[3], P3[2]</td>
<td>0.9</td>
<td>–</td>
<td>1.3</td>
<td>V</td>
</tr>
</tbody>
</table>

### Current consumption

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Conditions</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDD(_{20})</td>
<td>Current consumption, 20-bit, 187 sps(^9)</td>
<td>Unbuffered</td>
<td>–</td>
<td>–</td>
<td>4</td>
<td>mA</td>
</tr>
<tr>
<td>IDD(_{16})</td>
<td>Current consumption, 16-bit, 48 ksp s(^9)</td>
<td>Unbuffered</td>
<td>–</td>
<td>–</td>
<td>4</td>
<td>mA</td>
</tr>
<tr>
<td>IDD(_{12})</td>
<td>Current consumption, 12-bit, 192 ksp s(^9)</td>
<td>Unbuffered</td>
<td>–</td>
<td>–</td>
<td>4.5</td>
<td>mA</td>
</tr>
<tr>
<td>IDD(_{8})</td>
<td>Current consumption, 8-bit, 384 ksp s(^9)</td>
<td>Unbuffered</td>
<td>–</td>
<td>–</td>
<td>4.5</td>
<td>mA</td>
</tr>
<tr>
<td>IBUFF</td>
<td>Current consumption, 16-bit, 48 ksp s(^9)</td>
<td>Buffer alone</td>
<td>–</td>
<td>–</td>
<td>3.5</td>
<td>mA</td>
</tr>
</tbody>
</table>

### Delta-Sigma ADC AC Specifications for PSoC 5

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Conditions</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Startup time</td>
<td>–</td>
<td>Unbuffered, range ±1.024 V</td>
<td>–</td>
<td>–</td>
<td>4</td>
<td>Samples</td>
</tr>
<tr>
<td>THD</td>
<td>Total harmonic distortion, 16-bit(^{11})</td>
<td>–</td>
<td>–</td>
<td>0.006</td>
<td>%</td>
<td></td>
</tr>
</tbody>
</table>

#### 20-Bit Resolution Mode

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Conditions</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR20</td>
<td>Sample Rate, 20-bit(^{11})</td>
<td>Differential range ±1.024 V</td>
<td>7.8</td>
<td>–</td>
<td>187</td>
<td>sps</td>
</tr>
<tr>
<td>BW20</td>
<td>Bandwidth, 20-bit(^{11})</td>
<td>Differential range ±1.024 V</td>
<td>–</td>
<td>40</td>
<td>–</td>
<td>Hz</td>
</tr>
</tbody>
</table>

#### 16-Bit Resolution Mode

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Conditions</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR16</td>
<td>Sample Rate, 16-bit(^{11})</td>
<td>Differential range ±1.024 V</td>
<td>2</td>
<td>–</td>
<td>48</td>
<td>ksps</td>
</tr>
<tr>
<td>BW16</td>
<td>Bandwidth, 16-bit(^{11})</td>
<td>Differential range ±1.024 V</td>
<td>–</td>
<td>11</td>
<td>–</td>
<td>kHz</td>
</tr>
<tr>
<td>SINAD16int</td>
<td>Signal to noise + distortion - 16-bit, internal reference(^{11})</td>
<td>Unbuffered, range ±1.024 V</td>
<td>81</td>
<td>–</td>
<td>–</td>
<td>dB</td>
</tr>
</tbody>
</table>

\(^{11}\) Based on device characterization (not production tested).
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Conditions</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>SINAD_16ext</td>
<td>Signal to noise + distortion, 16-bit, external reference</td>
<td>Unbuffered, range = ±1.024 V</td>
<td>84</td>
<td>–</td>
<td>–</td>
<td>dB</td>
</tr>
</tbody>
</table>

### 12-Bit Resolution Mode

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Conditions</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR12</td>
<td>Sample Rate, 12-bit</td>
<td>Differential range ±1.024 V</td>
<td>4</td>
<td>–</td>
<td>192</td>
<td>kbps</td>
</tr>
<tr>
<td>BW12</td>
<td>Bandwidth, 12-bit</td>
<td>Differential range ±1.024 V</td>
<td>–</td>
<td>44</td>
<td>–</td>
<td>kHz</td>
</tr>
<tr>
<td>SINAD12int</td>
<td>Signal to noise + distortion, 12-bit, internal reference</td>
<td>Unbuffered, range = ±1.024 V</td>
<td>66</td>
<td>–</td>
<td>–</td>
<td>dB</td>
</tr>
</tbody>
</table>

### 8-Bit Resolution Mode

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Conditions</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR8</td>
<td>Sample Rate, 8-bit</td>
<td>Differential range ±1.024 V</td>
<td>8</td>
<td>–</td>
<td>384</td>
<td>kbps</td>
</tr>
<tr>
<td>BW8</td>
<td>Bandwidth, 8-bit</td>
<td>Differential range ±1.024 V</td>
<td>–</td>
<td>88</td>
<td>–</td>
<td>kHz</td>
</tr>
<tr>
<td>SINAD8int</td>
<td>Signal to noise + distortion, 8-bit, internal reference</td>
<td>Unbuffered, range = ±1.024 V</td>
<td>43</td>
<td>–</td>
<td>–</td>
<td>dB</td>
</tr>
</tbody>
</table>

### Voltage Reference Specifications

Also see the ADC external reference specifications under $V_{\text{EXTREF}}$ in Delta-Sigma ADC DC Specifications for PSoC 5.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Conditions</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{\text{REF}}$</td>
<td>Precision reference voltage</td>
<td>Initial trimming</td>
<td>1.017 (−0.7%)</td>
<td>1.024</td>
<td>1.033 (+0.9%)</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td>Temperature drift</td>
<td></td>
<td>–</td>
<td>–</td>
<td>57</td>
<td>ppm/°C</td>
</tr>
<tr>
<td></td>
<td>Long term drift</td>
<td></td>
<td>–</td>
<td>100</td>
<td>–</td>
<td>ppm/Khr</td>
</tr>
<tr>
<td></td>
<td>Thermal cycling drift (stability)</td>
<td></td>
<td>–</td>
<td>100</td>
<td>–</td>
<td>ppm</td>
</tr>
</tbody>
</table>

---

12 Based on device characterization (not production tested).
Component Changes

This section lists the major changes in the component from the previous version.

<table>
<thead>
<tr>
<th>Version</th>
<th>Description of Changes</th>
<th>Reason for Changes / Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.20.a</td>
<td>Datasheet corrections</td>
<td></td>
</tr>
<tr>
<td>2.20</td>
<td>Three new APIs ADC_SetCoherency(), ADC_SetGCOR() and ADC_ReadGCOR() are implemented</td>
<td>ADC_SetCoherency() for setting the coherency register. Helps to easily change the byte coherency. ADC_SetGCOR() for setting the Gain correction register. ADC_ReadGCOR() for reading the normalized GCOR register value.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added four new Bypass reference options.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added comment and config name parameters</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The config parameter helps to uniquely name the ADC configurations. The comment parameter describes configuration use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ADC_Stop() API changes for PSoC 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Change required to prevent the component from impacting unrelated analog signals when stopped, when used with PSoC 5.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ADC power setting changes based on the ADC clock frequency</td>
</tr>
<tr>
<td></td>
<td></td>
<td>To optimize the power use by ADC component.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fixed an issue with ADC_SelectConfiguration() API.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Calling ADC_SelectConfiguration() API without calling ADC_Start() now works as expected.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added PSoC 5 characterization data to datasheet</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated placeholder graphs in datasheet with real data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>External Charge pump clock is the default option and cannot be user selectable.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ADC now always uses an external charge pump clock instead of an internal clock. This is to guarantee the optimal charge pump clock frequency to maximize ADC performance.</td>
</tr>
<tr>
<td>2.10</td>
<td>Two differential input range ±0.128 V (–Input ± Vref/8) and ±0.064 V (–Input ± Vref/16)</td>
<td>These input ranges have an error in their name.</td>
</tr>
<tr>
<td></td>
<td>names are changed.</td>
<td><strong>Note:</strong> Updating an existing design that uses an older version than 2.0 with the version 2.10 of the ADC_DelSig, may result in a parameter evaluation failed error message. To fix the problem, open the Configure dialog, and then toggle the selection for Input mode. Click <strong>OK</strong> to close the dialog and then build the project.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added two new reference options: Internal V_{DDA}/4 and Internal V_{DDA}/3.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The reference input for the PSoc 3 ADC has a narrow range, 0.9 V to 1.3 V. This may cause the V_{SS} to V_{DDA} ratiometric range to operate incorrectly when V_{DDA} is less than 3.6 V. This new V_{DDA}/3 reference corrects this problem.</td>
</tr>
<tr>
<td>Version</td>
<td>Description of Changes</td>
<td>Reason for Changes / Impact</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Changing the reference value affects the input range list.</td>
<td>Input range list updates are based on the reference voltage selected.</td>
</tr>
<tr>
<td></td>
<td>Error provider is added to Vref Value parameter.</td>
<td>To warn the user if the Vref value exceeds the expected range 0.9V to 1.3V.</td>
</tr>
<tr>
<td></td>
<td>ADC clock frequency is now dependent on input range selection.</td>
<td>ADC clock frequency depends on the selected input range. ADC maximum clock frequency differs based on the input range. If ADC clock frequency exceeds the maximum clock frequency for the selected input range, then an error symbol is set on the clock frequency text box.</td>
</tr>
<tr>
<td></td>
<td>Fixed an issue with offset calculation.</td>
<td>0 to 2*Vref single ended mode for resolutions 9, 10, 11, 13, 14, and 15 now works as expected.</td>
</tr>
<tr>
<td></td>
<td>Fixed an issue with DMA wizard tool generated code.</td>
<td>DMA wizard tool uses correct ADC output register when resolution is above 8 bits.</td>
</tr>
<tr>
<td></td>
<td>Added characterization data to datasheet</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Minor datasheet edits and updates</td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td>Changed the Input Range parameter: added four new input ranges:</td>
<td>When updating an existing project with the 2.0 version of the ADC_DelSig, this change may result in a parameter evaluation failed error message. To fix the problem, open the Configure dialog, and then toggle the selection for Input mode. Click OK to close the dialog and then build the project.</td>
</tr>
<tr>
<td></td>
<td>Vssa to 6*Vref</td>
<td></td>
</tr>
<tr>
<td></td>
<td>±6.144 V (–Input ± 6*Vref)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>±0.125 V (–Input ± Vref/8)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>±0.0625 V (–Input ± Vref/16)</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> This change will likely break existing designs.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Updated to support PSoC 3 ES2 or later and PSoC 5 or later.</td>
<td>This version supports PSoC 3 ES2 or later and PSoC 5 or later. Older versions of the component will display an error message when used with newer versions of the silicon.</td>
</tr>
<tr>
<td></td>
<td>Added Sleep/Wakeup and Init/Enable APIs.</td>
<td>To support low power modes, as well as to provide common interfaces to separate control of initialization and enabling of most components.</td>
</tr>
<tr>
<td></td>
<td>Added new parameters: Input Mode, and Buffer Mode.</td>
<td>These new parameters were not present in the older versions of the ADC_DelSig component. When updating to version 2.0 of the component, the new parameters are given default values. The default value for the Input Mode parameter is “Differential.” The Input Mode parameter drives the selection of the Input Range. Therefore, an updated component previously configured with a “single-ended” Input Range will get a default value of “Differential.”</td>
</tr>
<tr>
<td></td>
<td>Removed the Power parameter and added new Conversion Mode “Single Sample.”</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Added DMA capabilities file to the component.</td>
<td>This file allows the ADC_DelSig to be supported by the DMA Wizard tool in PSoC Creator.</td>
</tr>
<tr>
<td></td>
<td>Added Keil function reentrancy support to the APIs.</td>
<td>Adds the capability for customers to specify individual generated functions as reentrant.</td>
</tr>
<tr>
<td>Version</td>
<td>Description of Changes</td>
<td>Reason for Changes / Impact</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>----------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Edited the Configure Dialog.</td>
<td>Made Voltage Reference parameter editable.</td>
<td>Added different configurations to support changing the configuration during run time.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dialog allows you to modify the voltage values when V_SSA to V-DDA input range is selected.</td>
</tr>
<tr>
<td>Trim values are incorporated into the ADC implementation for the selected input ranges.</td>
<td>Trim values are used to adjust the Decimator gain to improve the performance of ADC.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added different configurations to support changing the configuration during run time.</td>
</tr>
<tr>
<td>Added constants to the header file for easier use.</td>
<td></td>
<td>The ADC component now has constants such as reference used, gain set, mode used, sample rate used, and so on, so that you can use them in your applications.</td>
</tr>
<tr>
<td>New optional connection has been added to the ADC DelSig component</td>
<td></td>
<td>This can be used to connect the -ve input of modulator to AGL6.</td>
</tr>
<tr>
<td>Charge pump power setting has been enabled depending on the clock frequency.</td>
<td></td>
<td>There was a problem with 8-bit ADC range. The problem was caused by not setting charge pump power setting bits with respect to ADC clock in the DSM_CR16 register. ADC code was modified to set these bits depending on the ADC clock frequency.</td>
</tr>
<tr>
<td>Removed the SetPower API.</td>
<td></td>
<td>The SetPower API was a nonfunctioning API. It was removed intentionally because it did not offer any value. If you had this function in your code, you need to remove it.</td>
</tr>
</tbody>
</table>