Features and Overview

- 11-bit resolution
- Data format available in 2’s complement
- Sample rate to 7.8 ksps
- 256X over sampling with sinc² filter reduces antialias requirements
- Input range defined by internal and external reference options
- Internal or external clock

Note   If this user module is used with the CY8C29xxx family, it consumes an extra 6 mA. As an alternative, use the Delsig User Module instead.

The DELSIG11 User Module provides an 11-bit output. It is based on a 2.6V full scale input range centered around a user selected AGND, when the reference selection in the global parameter window is set to +/- Bandgap. The DELSIG11 supports sample rates from 125 sps to 7.8 ksps, and provides a 2’s complement output. The sample rate is determined by the data clock input and is selectable by the user. Data generated by the DELSIG11 is available in the interrupt routine where the data is collected or through polling functions furnished by the DELSIG11 API.

The DELSIG11 is a pipeline integrating converter, requiring 511 integration cycles to generate a single output sample. If this converter is to have a multiplexed input, two samples must pass before the third and following samples are valid. Note that you need to review the Parameters section before module placement.
**Functional Description**

The DELSIG11 provides a first order modulator formed from a single analog switched capacitor PSoC block, one digital PSoC block, and the decimator, as shown in the following figure.

Figure 2. Schematic of the DELSIG11 with First Order Modulator
A second order modulator can be constructed in the CY8C29/27/24/22xxx, CY8C23x33, CY8CLEDO8/16, CY8C28x45family of PSoC devices using a second switched capacitor PSoC block. This improves performance by shifting more of the quantization noise out of band, for improved SNR. The second order circuit shown in the following schematic diagram uses an analog column comparator bus to modulate the reference selection.

Figure 3. Schematic of the DELSIG11 with Second Order Modulator

The range of the DELSIG11 is set at ±V_{Ref}, where V_{Ref} is set by the user in the Global Resources window of PSoC Designer. For fixed scale, V_{Ref} is set to ±V_{Bandgap} or, for the CY8C29/27/24/22xxx, CY8C23x33, CY8CLEDO8/16, CY8C28x45family of PSoC Devices, ±1.6 V_{Bandgap}. For adjustable scale, V_{Ref} is set to ±Port 2[6]. For supply ratio metric scale, V_{Ref} is set to ±V_{DD}/2.

The analog block is configured as an integrator. The reference control is configured so that the reference voltage is either added or subtracted, depending on the output polarity, from the input and placed in the integrator. This reference control attempts to pull the integrator output back towards zero. The single-bit comparator output is fed to the circuitry used to implement a decimator sinc^2 filter. The response of this filter is given by the following z-domain relation.

\[
H(z) = \left[ \frac{1 - z^{-n}}{1 - z^{-1}} \right]^2, \text{ where } n \text{ is the decimation level.}
\]
The frequency domain magnitude plot below normalizes the frequency so the 11-bit sample rate, $F_{\text{nom}} = 1.0$. The -3 dB point occurs just above $0.318 \times F_{\text{nom}}$ and zeros of the function occur at each integer multiple of $F_{\text{nom}}$. Since the DELSIG11 actually samples 256 times faster than the nominal output rate, the Nyquist limit is 128 times higher, 7 octaves above $F_{\text{nom}}$, which significantly reduces the requirements for an anti-alias filter.

Figure 4. Sinc² Decimation Filter Magnitude Response, with -3 dB point and Nyquist Frequency

This filter is implemented with a combination of hardware and software and is used for both the first and second order modulator topologies. The denominator of the filter is a double integrator that is implemented in hardware. It operates at the data rate. The numerator is a double differentiator. It is calculated at the decimation rate and is implemented in software. To make the integrator function as a delta-sigma ADC, the following digital resources are used:

- A timer to allow the proper number of integration cycles.
- A decimator to process the single-bit output stream from the analog block.

Normally, building an 11-bit delta sigma ADC converter requires an 8-bit timer. It would have to be clocked at one-fourth the frequency used to clock the integrator. For this implementation, the timer is clocked with the same clock used for the column clock, where the integrator block is placed. This causes the timer to be incremented by four for each integration cycle, thus requiring a 10-bit timer.

**Note** When placing this module, it is imperative that it is configured with the same source clock for both the analog and digital blocks. Failure to do so causes it to operate incorrectly.

The timer is set up to generate an interrupt every 256 counts. At the same time the interrupt is generated, the data in the decimator is latched to output registers. Four of these cycles (1024 counts) are used to calculate the A/D value in the interrupt routine.
The conversion time for the DELSIG11 can be as little as 128 $\mu$s, so fast retrieval of the data is important. Retrieving the data is accomplished by inserting the user’s data handler code into the interrupt routine DELSIG11_ADCconversion_ISR, located in the assembly file DELSIG11INT.asm. The point to insert code is clearly marked. Two different techniques of how the data may be processed are shown in the Sample Firmware Source Code section of this user module.

The conversion equation governing the 11-bit Delta Sigma ADC is shown below. The equation shows that the input range is limited to $V_{\text{ref}}$. The use of this equations is illustrated in this example:

$$V_{\text{in}} = \frac{n - 1024}{1024} V_{\text{ref}}$$  \hspace{1cm} \text{Equation 2}

**Example 1**

For a $V_{\text{ref}}$ of 1.3V, the input voltage can be easily calculated based on the value read from the incremental ADC at the time the data is ready. The equation which can be used is:

$$V_{\text{in}} = \frac{n - 1024}{1024} \cdot 1.3$$  \hspace{1cm} \text{Equation 3}

The result of the calculation is referenced to AGND. For a ADC data value of 1500 the Voltage measured can be calculated to be 0.60V:

$$V_{\text{in}} = \frac{1500 - 1024}{1024} \cdot 1.3 = 0.60V$$  \hspace{1cm} \text{Equation 4}

The value calculated is an ideal value and may differ based on system noise and chips offsets.

To determine the code to be expected given a specific input voltage the equation can be rearranged to give:

$$n = \frac{1024 \cdot V_{\text{in}}}{V_{\text{ref}}} + 1024$$  \hspace{1cm} \text{Equation 5}

**Example 2**

For a $V_{\text{ref}}$ of 1.3V we can easily calculate the expected ADC code based on the input Voltage. The equation which can be used is:

$$n = \frac{1024 \cdot V_{\text{in}}}{1.3} + 1024$$  \hspace{1cm} \text{Equation 6}
For an input voltage of -1V below AGND the code from the ADC can be expected to be 236.3 based on this calculation:

\[ n = \frac{1024 \cdot (-1)}{1.3} + 1024 = 236.3 \]

Equation 7

The opamp in the delsig modulator slews and settles to a new value on each clock pulse. The slew rate required is determined by signal level, reference level and clock rate. The slew rate available is determined by power level. Characterization is done with column clock = 2.0 MHz and yields optimum operation at Power = High and Opamp Bias = Low. Each power step, up (or down), results in an approximate power increase (or decrease) by a factor of two and a speed increase (or decrease) also by a factor of two. Lower power levels require lower sample rates and column clocks to maintain linearity and accuracy.

**DC and AC Electrical Characteristics**

The following values are indicative of expected performance and based on initial characterization data. Unless otherwise specified below TA = 25°C, V_{dd} = 5.0V, Power HIGH, OpAmp bias LOW, output referenced to 2.5V external Analog Ground on P2[4] with 1.25 external Vref on P2[6].

Table 1. 5.0V 2nd Order Modulator DC and AC Electrical Characteristics, CY8C29/27/24/22xxx, CY8C23x33, CY8CLED08/16, CY8C28x45Family of PSoC Devices

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Typical</th>
<th>Limit</th>
<th>Units</th>
<th>Conditions and Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input Voltage Range</td>
<td>---</td>
<td>Vss to Vdd</td>
<td>V</td>
<td>Ref Mux = Vdd/2 ± Vdd/2</td>
</tr>
<tr>
<td>Input Capacitance1</td>
<td>3</td>
<td>---</td>
<td>pF</td>
<td></td>
</tr>
<tr>
<td>Input Impedance</td>
<td>1/(C*clk)</td>
<td></td>
<td>Ω</td>
<td></td>
</tr>
<tr>
<td>Resolution</td>
<td>---</td>
<td>11</td>
<td>Bits</td>
<td></td>
</tr>
<tr>
<td>Sample Rate</td>
<td>---</td>
<td>125 to 7,800</td>
<td>sps</td>
<td></td>
</tr>
<tr>
<td>SNR</td>
<td>66</td>
<td>---</td>
<td>dB</td>
<td></td>
</tr>
<tr>
<td>DC Accuracy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DNL</td>
<td>0.25</td>
<td>---</td>
<td>LSB</td>
<td>Column Clock 2 MHz</td>
</tr>
<tr>
<td>INL</td>
<td>0.5</td>
<td>---</td>
<td>LSB</td>
<td></td>
</tr>
<tr>
<td>Offset Error</td>
<td>5</td>
<td>---</td>
<td>mV</td>
<td></td>
</tr>
<tr>
<td>Gain Error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Including Reference Gain Error</td>
<td>3.0</td>
<td>--</td>
<td>% FSR</td>
<td></td>
</tr>
<tr>
<td>Excluding Reference Gain Error2</td>
<td>0.1</td>
<td>--</td>
<td>% FSR</td>
<td></td>
</tr>
<tr>
<td>Operating Current</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 2. 5.0V 1st Order Modulator DC and AC Electrical Characteristics, CY8C29/27/24/22xxx, CY8C23x33, CY8CLED08/16, CY8C28x45Family of PSoC Devices

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Typical</th>
<th>Limit</th>
<th>Units</th>
<th>Conditions and Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low Power</td>
<td>180</td>
<td>---</td>
<td>uA</td>
<td></td>
</tr>
<tr>
<td>Med Power</td>
<td>840</td>
<td>---</td>
<td>uA</td>
<td></td>
</tr>
<tr>
<td>High Power</td>
<td>3450</td>
<td>---</td>
<td>uA</td>
<td></td>
</tr>
<tr>
<td>Data Clock</td>
<td>---</td>
<td>0.032 to 8.0</td>
<td>MHz</td>
<td>Input to digital blocks and analog column clock</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Typical</th>
<th>Limit</th>
<th>Units</th>
<th>Conditions and Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Voltage Range</td>
<td>---</td>
<td>Vss to Vdd</td>
<td>V</td>
<td>Ref Mux = Vdd/2 ± Vdd/2</td>
</tr>
<tr>
<td>Input Capacitance(^1)</td>
<td>3</td>
<td>---</td>
<td>pF</td>
<td></td>
</tr>
<tr>
<td>Input Impedance</td>
<td>1/(C*clk)</td>
<td>---</td>
<td>Ω</td>
<td></td>
</tr>
<tr>
<td>Resolution</td>
<td>---</td>
<td>11</td>
<td>Bits</td>
<td></td>
</tr>
<tr>
<td>Sample Rate</td>
<td>---</td>
<td>125 to 7,800</td>
<td>sps</td>
<td></td>
</tr>
<tr>
<td>SNR</td>
<td>65</td>
<td>---</td>
<td>dB</td>
<td></td>
</tr>
<tr>
<td>DC Accuracy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DNL</td>
<td>1.4</td>
<td>---</td>
<td>LSB</td>
<td>Column Clock 2 MHz</td>
</tr>
<tr>
<td>INL</td>
<td>1.2</td>
<td>---</td>
<td>LSB</td>
<td></td>
</tr>
<tr>
<td>Offset Error</td>
<td>5</td>
<td>---</td>
<td>mV</td>
<td></td>
</tr>
<tr>
<td>Gain Error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Including Reference Gain Error</td>
<td>2.0</td>
<td>--</td>
<td>% FSR</td>
<td></td>
</tr>
<tr>
<td>Excluding Reference Gain Error(^2)</td>
<td>0.1</td>
<td>--</td>
<td>% FSR</td>
<td></td>
</tr>
<tr>
<td>Operating Current</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low Power</td>
<td>50</td>
<td>---</td>
<td>uA</td>
<td></td>
</tr>
<tr>
<td>Med Power</td>
<td>500</td>
<td>---</td>
<td>uA</td>
<td></td>
</tr>
<tr>
<td>High Power</td>
<td>1900</td>
<td>---</td>
<td>uA</td>
<td></td>
</tr>
<tr>
<td>Data Clock</td>
<td>---</td>
<td>0.032 to 8.0</td>
<td>MHz</td>
<td>Input to digital blocks and analog column clock</td>
</tr>
</tbody>
</table>
The following values are indicative of expected performance and based on initial characterization data. Unless otherwise specified below, $T_A = 25^\circ C$, $V_{dd} = 3.3V$, Power HIGH, OpAmp bias LOW, output referenced to 1.64V external Analog Ground on P2[4] with 1.25 external Vref on P2[6].

Table 3. 3.3V 2nd Order Modulator DC and AC Electrical Characteristics, CY8C29/27/24/22xxx, CY8C23x33, CY8CLED08/16, CY8C28x45Family of PSoC Devices

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Typical</th>
<th>Limit</th>
<th>Units</th>
<th>Conditions and Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Voltage Range</td>
<td>---</td>
<td>Vss to Vdd</td>
<td>V</td>
<td>Ref Mux = Vdd/2 ± Vdd/2</td>
</tr>
<tr>
<td>Input Capacitance(^1)</td>
<td>3</td>
<td>---</td>
<td>pF</td>
<td></td>
</tr>
<tr>
<td>Input Impedance</td>
<td>$1/(C*\text{clk})$</td>
<td>---</td>
<td>$\Omega$</td>
<td></td>
</tr>
<tr>
<td>Resolution</td>
<td>---</td>
<td>11</td>
<td>Bits</td>
<td></td>
</tr>
<tr>
<td>Sample Rate</td>
<td>---</td>
<td>125 to 7,800</td>
<td>sps</td>
<td></td>
</tr>
<tr>
<td>SNR</td>
<td>66</td>
<td>---</td>
<td>dB</td>
<td></td>
</tr>
<tr>
<td>DC Accuracy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DNL</td>
<td>0.25</td>
<td>---</td>
<td>LSB</td>
<td>Column Clock 2 MHz</td>
</tr>
<tr>
<td>INL</td>
<td>0.5</td>
<td>---</td>
<td>LSB</td>
<td></td>
</tr>
<tr>
<td>Offset Error</td>
<td>5</td>
<td>---</td>
<td>mV</td>
<td></td>
</tr>
<tr>
<td>Gain Error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Including Reference Gain Error</td>
<td>3.0</td>
<td>--</td>
<td>% FSR</td>
<td></td>
</tr>
<tr>
<td>Excluding Reference Gain Error(^2)</td>
<td>0.3</td>
<td>--</td>
<td>% FSR</td>
<td></td>
</tr>
<tr>
<td>Operating Current</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low Power</td>
<td>130</td>
<td>---</td>
<td>uA</td>
<td></td>
</tr>
<tr>
<td>Med Power</td>
<td>840</td>
<td>---</td>
<td>uA</td>
<td></td>
</tr>
<tr>
<td>High Power</td>
<td>3370</td>
<td>---</td>
<td>uA</td>
<td></td>
</tr>
<tr>
<td>Data Clock</td>
<td>---</td>
<td>0.032 to 8.0</td>
<td>MHz</td>
<td>Input to digital blocks and analog column clock</td>
</tr>
</tbody>
</table>
Table 4. 3.3V 1st Order Modulator DC and AC Electrical Characteristics, CY8C29/27/24/22xxx, CY8C23x33, CY8CLED08/16, CY8C28x45Family of PSoC Devices

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Typical</th>
<th>Limit</th>
<th>Units</th>
<th>Conditions and Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input Voltage Range</td>
<td></td>
<td>Vss to Vdd</td>
<td>V</td>
<td>Ref Mux = Vdd/2 ± Vdd/2</td>
</tr>
<tr>
<td>Input Capacitance¹</td>
<td>3</td>
<td>---</td>
<td>pF</td>
<td></td>
</tr>
<tr>
<td>Input Impedance</td>
<td>1/(C*clk)</td>
<td>---</td>
<td>Ω</td>
<td></td>
</tr>
<tr>
<td>Resolution</td>
<td>---</td>
<td>11</td>
<td>Bits</td>
<td></td>
</tr>
<tr>
<td>Sample Rate</td>
<td>---</td>
<td>125 to 7,800</td>
<td>sps</td>
<td></td>
</tr>
<tr>
<td>SNR</td>
<td>65</td>
<td>---</td>
<td>dB</td>
<td></td>
</tr>
<tr>
<td>DC Accuracy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DNL</td>
<td>1.4</td>
<td>---</td>
<td>LSB</td>
<td>Column Clock 2 MHz</td>
</tr>
<tr>
<td>INL</td>
<td>1.1</td>
<td>---</td>
<td>LSB</td>
<td></td>
</tr>
<tr>
<td>Offset Error</td>
<td>5</td>
<td>---</td>
<td>mV</td>
<td></td>
</tr>
<tr>
<td>Gain Error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Including Reference Gain Error</td>
<td>2.0</td>
<td>--</td>
<td>% FSR</td>
<td></td>
</tr>
<tr>
<td>Excluding Reference Gain Error²</td>
<td>0.3</td>
<td>--</td>
<td>% FSR</td>
<td></td>
</tr>
<tr>
<td>Operating Current</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low Power</td>
<td>50</td>
<td>---</td>
<td>uA</td>
<td></td>
</tr>
<tr>
<td>Med Power</td>
<td>500</td>
<td>---</td>
<td>uA</td>
<td></td>
</tr>
<tr>
<td>High Power</td>
<td>1900</td>
<td>---</td>
<td>uA</td>
<td></td>
</tr>
<tr>
<td>Data Clock</td>
<td></td>
<td>0.032 to 8.0</td>
<td>MHz</td>
<td>Input to digital blocks and analog column clock</td>
</tr>
</tbody>
</table>

Electrical Characteristics Notes

1. Includes I/O pin.
2. Reference Gain Error measured by comparing the external reference to $V_{\text{RefHigh}}$ and $V_{\text{RefLow}}$ routed through the test mux and back out to a pin.
Unless otherwise specified, all limits guaranteed for TA = -40°C to, +85°C, Vdd = 5.0V ±10%, Power HIGH, OpAmp bias LOW, output referenced to 2.5V external Analog Ground on P2[4] with 1.25 external Vref on P2[6].

Table 5. 2.7V 2nd Order Modulator DELSIG11 DC and AC Electrical Characterisitics, CY8C24x23 Family of PSoC Devices

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Typical</th>
<th>Limit</th>
<th>Units</th>
<th>Conditions and Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input Voltage Range</td>
<td>---</td>
<td>Vss to Vdd</td>
<td>V</td>
<td>Ref Mux = Vdd/2 ± Vdd/2</td>
</tr>
<tr>
<td>Input Capacitance</td>
<td>3</td>
<td>---</td>
<td>pF</td>
<td></td>
</tr>
<tr>
<td>Input Impedance</td>
<td>1/(C*fclk)</td>
<td>---</td>
<td>Ω</td>
<td></td>
</tr>
<tr>
<td>Resolution</td>
<td>---</td>
<td>11</td>
<td>Bits</td>
<td></td>
</tr>
<tr>
<td>Sample Rate</td>
<td>---</td>
<td>125 to 7,800</td>
<td>sps</td>
<td></td>
</tr>
<tr>
<td>SNR</td>
<td>65</td>
<td>---</td>
<td>dB</td>
<td></td>
</tr>
<tr>
<td>DC Accuracy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DNL</td>
<td>2.3</td>
<td>---</td>
<td>LSB</td>
<td></td>
</tr>
<tr>
<td>INL</td>
<td>32</td>
<td>---</td>
<td>LSB</td>
<td></td>
</tr>
<tr>
<td>Offset Error</td>
<td>6.6</td>
<td>---</td>
<td>mV</td>
<td></td>
</tr>
<tr>
<td>Gain Error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Including Reference Gain Error</td>
<td>0.6</td>
<td>--</td>
<td>% FSR</td>
<td></td>
</tr>
<tr>
<td>Operating Current</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low Power</td>
<td>160</td>
<td>---</td>
<td>µA</td>
<td></td>
</tr>
<tr>
<td>Med Power</td>
<td>440</td>
<td>---</td>
<td>µA</td>
<td></td>
</tr>
<tr>
<td>High Power</td>
<td>1851</td>
<td>---</td>
<td>µA</td>
<td></td>
</tr>
<tr>
<td>Data Clock</td>
<td>---</td>
<td>0.032 to 8.0</td>
<td>MHz</td>
<td></td>
</tr>
</tbody>
</table>
Table 6. 2.7V 1st Order Modulator DELSIG11 DC and AC Electrical Characteristics, CY8C24x23 Family of PSoC Devices

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Typical</th>
<th>Limit</th>
<th>Units</th>
<th>Conditions and Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input Voltage Range</td>
<td>---</td>
<td>Vss to Vdd</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>Input Capacitance¹</td>
<td>3</td>
<td>---</td>
<td>pF</td>
<td></td>
</tr>
<tr>
<td>Input Impedance</td>
<td>1/(C*clk)</td>
<td>---</td>
<td>Ω</td>
<td></td>
</tr>
<tr>
<td>Resolution</td>
<td>---</td>
<td>11</td>
<td>Bits</td>
<td></td>
</tr>
<tr>
<td>Sample Rate</td>
<td>---</td>
<td>125 to 7,800</td>
<td>sps</td>
<td></td>
</tr>
<tr>
<td>SNR</td>
<td>65</td>
<td>---</td>
<td>dB</td>
<td></td>
</tr>
<tr>
<td>DC Accuracy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DNL</td>
<td>1.4</td>
<td>---</td>
<td>LSB</td>
<td></td>
</tr>
<tr>
<td>INL</td>
<td>4.6</td>
<td>---</td>
<td>LSB</td>
<td></td>
</tr>
<tr>
<td>Offset Error</td>
<td>5.3</td>
<td>---</td>
<td>mV</td>
<td></td>
</tr>
<tr>
<td>Gain Error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Including Reference Gain Error</td>
<td>0.8</td>
<td>---</td>
<td>% FSR</td>
<td></td>
</tr>
<tr>
<td>Operating Current</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low Power</td>
<td>110</td>
<td>---</td>
<td>uA</td>
<td></td>
</tr>
<tr>
<td>Med Power</td>
<td>332</td>
<td>---</td>
<td>uA</td>
<td></td>
</tr>
<tr>
<td>High Power</td>
<td>1272</td>
<td>---</td>
<td>uA</td>
<td></td>
</tr>
<tr>
<td>Data Clock</td>
<td>---</td>
<td>0.032 to 8.0</td>
<td>MHz</td>
<td></td>
</tr>
</tbody>
</table>

Electrical Characteristics Notes

1. Typical values represent parametric norm at +25°C.
2. Input voltages above the maximum generate a maximum positive reading. Input voltages below the minimum generate a maximum negative reading.
3. User module only, not including I/O pin.
4. The input capacitance or impedance is only applicable when input to analog block is directly to a pin.
5. C = input capacitance, clk = data clock (Analog Column Clock).
6. DataClock = SampleRate * 1024.
7. SNR = Ratio of power of full scale single tone divided by total noise integrated to F_{sample}/2.
CY8C29/24xxx Typical Performance

These graphs are restricted to a subset of the input range that best displays the dominant error. Operation with the user module set to LOWPOWER is not recommended.

Figure 5. Typical DNL as a Function of Power Level, CY8C24xxx, 3.3V, 25°C

![Figure 5](image1)

Figure 6. Typical DNL as a Function of Power Level, First Order, CY8C24xxx, 5.0V, 25°C

![Figure 6](image2)
Figure 7. Typical DNL as a Function of Power Level, Second Order, CY8C24xxx, 3.3V, 25°C

![DNL Graph 3.3V](image)

Figure 8. Typical DNL as a Function of Power Level, Second Order, CY8C24xxx, 5.0V, 25°C

![DNL Graph 5.0V](image)
Figure 9. Typical DNL as a Function of Power Level, First Order, CY8C29xxx, 3.3V, 25°C

Figure 10. Typical DNL as a Function of Power Level, First Order, CY8C29xxx, 5.0V, 25°C
Placement

The first order modulator design requires two PSoC blocks, one digital and one analog. No inherent limitations govern placement of the analog block; the only considerations are input and clock availability. The digital block, however, must be able to feed the hardware decimator. In the CY8C27xxx (A-silicon) family, the qualified digital blocks are DBB01, DCB02, DBB11 and DCB12. In the CY8C29/24/22xxx and CY8C27xxx (B-silicon, with ‘X’ in suffix for Pb-free package) device families, any of the digital blocks can be used. As noted later, both blocks must utilize the same source clock.

Placement for second order modulator design differs from the first order design in that there is a second switched capacitor PSoC block. Both analog blocks must lie in the same column so they can share the column comparator bus. The digital block is subject to the same restrictions for both first- and second order modulators.
Although there are a number of placements possible for the analog and digital blocks, the DELSIG11 also uses the PSoC device’s only hardware decimator. Only one DELSIG11 instance may be placed for a given configuration. With dynamic re-configuration it is possible to load more than one configuration as long as the blocks do not overlap. Though both instances appear to work, only the output of the one most recently loaded would be correct.

**Parameters and Resources**

Once a DELSIG11 instance is placed, four parameters must be configured for proper operation: the Input, Clock Phase, Data Clock, and Polling selection.

**Data Position**

The ADC generates 11 bits that are positioned in two 8-bit registers. The data is either right justified or left justified. Right-justified data has the lower 8 bits stored in the least significant byte and the upper 3 bits stored in the lower 3 bits of the most significant byte. The upper 5 bits are a sign extension, to preserve the 2’s complement format of the data. Left-justified data is stored with the upper 8 bits in the most significant byte and the lower 3 bits in the upper 3 bits of the least significant byte. The lower 5 bits contain residue of the decimator.

**Input**

This parameter determines the signal source for the input to the ADC.

**Clock Phase**

The selection of the Clock Phase is used to synchronize the output of one analog PSoC block to the input of another. The switched capacitor analog PSoC blocks use a two-phase clock (φ1, φ2) to acquire and transfer signals. Normally, the input to the DELSIG11 is sampled on φ1. A problem arises in that many of the user modules autozero their output during φ1 and only provide a valid output during φ2. If such a module’s output is fed to the DELSIG11’s input, the DELSIG11 acquires an autozeroed output instead of a valid signal. The Clock Phase selection allows the phases to be swapped, so that the input signal is acquired during φ2.

**TMR Clock**

The TMR Clock determines the sample rate. This clock goes to both PSoC blocks of the first order modulator design and to all three PSoC blocks of the second order design.

*Note** CAUTION: It is imperative that the same clock is selected for both the digital block and the analog column clock or this user module does not function correctly.

*Note** The TMR Clock is called the "data clock" in other PSoC user module datasheets.

The timer is set to provide an interrupt every 256 counts of the TMR clock and it takes 4 interrupts to process the data. The sample rate is thus defined as follows.

\[
\text{Sample Rate} = \frac{\text{Data Clock}}{1024}
\]
Example 1

If a sample rate of 5 ksp is desired, the TMR clock must be:

\[ \text{DataClock} = \text{SampleRate} \times 1024 = 5\text{ksps} \times 1024 = 5.12\text{MHz} \]  

Equation 9

The CPU overhead is also dependent on the data clock. The interrupt routine that calculates the ADC value must service 4 interrupts for each output sample. Each of the 4 executes a different section of code within the service routine. The first and third sections require 90 CPU cycles, the second section requires 154 CPU cycles, and the fourth section requires 289 CPU cycles. Overall, 623 CPU cycles are required to process the ADC data. Averaged over one sample period, total CPU use by the 4 sections is:

\[ \text{AverageCPUOverhead} = \frac{623 \times \text{SampleRate}}{\text{CPUClock}} \]  

Equation 10

During any given interrupt period, however, CPU use varies according to the number of cycles required to service that particular interrupt. The overhead for each section may be computed as:

\[ \text{FirstSectionCPUOverhead} = \frac{90 \cdot 4 \cdot \text{SampleRate}}{\text{CPUClock}} \]  

Equation 11

\[ \text{SecondSectionCPUOverhead} = \frac{154 \cdot 4 \cdot \text{SampleRate}}{\text{CPUClock}} \]  

Equation 12

\[ \text{ThirdSectionCPUOverhead} = \frac{90 \cdot 4 \cdot \text{SampleRate}}{\text{CPUClock}} \]  

Equation 13

\[ \text{FourthSectionCPUOverhead} = \frac{289 \cdot 4 \cdot \text{SampleRate}}{\text{CPUClock}} \]  

Equation 14

Example Overhead Calculation

If the sample rate is 5 ksp and the CPU clock is 24 MHz, the CPU overhead is:

\[ \text{TotalCPUOverhead} = \frac{623 \times 5\text{ksps}}{24\text{MHz}} = 13\% \]  

Equation 15
The overhead for each section is:

\[
\begin{align*}
\text{First section} &= \frac{90 \cdot (4 \cdot 5 \text{ksps})}{24\text{MHz}} = 7.5\% \\
\text{Second section} &= \frac{154 \cdot (4 \cdot 5 \text{ksps})}{24\text{MHz}} = 12.8\% \\
\text{Third section} &= \frac{90 \cdot (4 \cdot 5 \text{ksps})}{24\text{MHz}} = 7.5\% \\
\text{Fourth section} &= \frac{289 \cdot (4 \cdot 5 \text{ksps})}{24\text{MHz}} = 24.1\%
\end{align*}
\]

**Polling**

The conversion time for the ADC can be as little as 130 μs, so fast retrieval of the data may be important. The data produced by the sinc2 filter, is computed under control of the digital block’s Interrupt Service Routine (ISR). Two options are available for accessing this data controlled by the Polling property which may be set to Enable or Disable.

When Polling is set to Disable, retrieving the data is accomplished by inserting the code to do so into the assembly language interrupt routine DELSIG11_ADConversion_ISR, located in the assembly file delsig11INT.asm. The point to insert code is clearly marked.

When Polling is set to Enable, two additional RAM variables are allocated, one to store a copy of the data and one to describe its availability. These variables and associated polling functions are described in the next section.

**Interrupt Generation Control**

There is an additional parameter that becomes available when the Enable interrupt generation control check box in PSoC Designer is checked. This is available under Project > Settings > Chip Editor. Interrupt Generation Control is important when multiple overlays are used with interrupts shared by multiple user modules across overlays:

**IntDispatchMode**

The IntDispatchMode parameter is used to specify how an interrupt request is handled for interrupts shared by multiple user modules existing in the same block but in different overlays. Selecting “ActiveStatus” causes firmware to test which overlay is active before servicing the shared interrupt request. This test occurs every time the shared interrupt is requested. This adds latency and also produces a nondeterministic procedure of servicing shared interrupt requests, but does not require any RAM. Selecting “OffsetPreCalc” causes firmware to calculate the source of a shared interrupt request only when an overlay is initially loaded. This calculation decreases interrupt latency and produces a deterministic procedure for servicing shared interrupt requests, but at the expense of a byte of RAM.
Application Programming Interface

The Application Programming Interface (API) routines are provided as part of the user module to allow the
designer to deal with the module at a higher level. This section specifies the interface to each function
together with related constants provided by the “include” files.

Note

In this, as in all user module APIs, the values of the A and X register may be altered by calling an API
function. It is the responsibility of the calling function to preserve the values of A and X before the call if
those values are required after the call. This “registers are volatile” policy was selected for efficiency
reasons and has been in force since version 1.0 of PSoC Designer. The C compiler automatically takes
care of this requirement. Assembly language programmers must ensure their code observes the policy,
too. Though some user module API function may leave A and X unchanged, there is no guarantee they
may do so in the future.

For Large Memory Model devices, it is also the caller's responsibility to preserve any value in the
CUR_PP, IDX_PP, MVR_PP, and MVW_PP registers. Even though some of these registers may not be
modified now, there is no guarantee that will remain the case in future releases.

Each time a user module is placed, it is assigned an instance name. By default, PSoC Designer assigns
the DELSIG11_1 to the first instance of this user module in a given project. It can be changed to any
unique value that follows the syntactic rules for identifiers. The assigned instance name becomes the
prefix of every global function name, variable and constant symbol. In the following descriptions the
instance name has been shortened to DELSIG11 for simplicity.

Global Variable DELSIG11_bfStatus

Description:

This variable is available only if the value of the Polling parameter is set to Enable. It is set to a
nonzero value when the global variable DELSIG11_iResult contains valid data. This variable may be
accessed directly or indirectly through the API functions DELSIG11_ClearFlag,
DELSIG11_fIsDataAvailable, and DELSIG11_GetDataClearFlag.

C Prototype:

BOOL DELSIG11_bfStatus;

Global Variable DELSIG11_iResult

Description:

This variable is available only if the value of the Polling parameter is set to Enable. Each sample of
data converted by the DELSIG11 is placed in this variable and the value of the related boolean vari-
able, DELSIG11_bfStatus, is set to a nonzero value. The contents of this variable may be accessed
directly or indirectly through the API functions DELSIG11_GetData and
DELSIG11_GetDataClearFlag. Also see the API function DELSIG11_fIsDataAvailable.

C Prototype:

int DELSIG11_iResult;

DELSIG11_Start

Description: Performs all required initialization for this user module and sets the power level for the
switched capacitor PSoC block
C Prototype:

```c
void DELSIG11_Start (BYTE bPowerSetting)
```

Assembly:

```assembly
mov A, bPowerSetting
lcall DELSIG11_Start
```

**Parameters:**

- `bPowerSetting`: One byte that specifies the power level. Following reset and configuration, the analog PSoC block assigned to DELSIG11 is powered down. Symbolic names provided in C and assembly, and their associated values are given in the following table.

<table>
<thead>
<tr>
<th>Symbolic Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELSIG11_OFF</td>
<td>0</td>
</tr>
<tr>
<td>DELSIG11_LOWPOWER</td>
<td>1</td>
</tr>
<tr>
<td>DELSIG11_MEDPOWER</td>
<td>2</td>
</tr>
<tr>
<td>DELSIG11_HIGHPOWER</td>
<td>3</td>
</tr>
</tbody>
</table>

**Return Value:**

None

**Side Effects:**

The A and X registers may be modified by this or future implementations of this function. The same is true for all RAM page pointer registers in the Large Memory Model (CY8C29xxx). When necessary, it is the calling function’s responsibility to preserve the values across calls to fastcall16 functions.

### DELSIG11_SetPower

**Description:**

Sets the power level for the switched capacitor PSoC block.

**C Prototype:**

```c
void DELSIG11_SetPower (BYTE bPowerSetting)
```

**Assembly:**

```assembly
mov A, bPowerSetting
lcall DELSIG11_SetPower
```

**Parameters:**

- `bPowerSetting`: Same as the `bPowerSetting` parameter used for the Start entry point.

**Return Value:**

None

**Side Effects:**

The A and X registers may be modified by this or future implementations of this function. The same is true for all RAM page pointer registers in the Large Memory Model (CY8C29xxx). When necessary, it is the calling function’s responsibility to preserve the values across calls to fastcall16 functions.
DELSIG11_Stop

Description:
Sets the power level to the switched capacitor PSoC block to OFF.

C Prototype:
void DELSIG11_Stop (void)

Assembly:
1call DELSIG11_Stop

Parameters:
None

Return Value:
None

Side Effects:
The A and X registers may be modified by this or future implementations of this function. The same
is true for all RAM page pointer registers in the Large Memory Model (CY8C29xxx). When necessary,
it is the calling function's responsibility to preserve the values across calls to fastcall16 functions.

DELSIG11_StartAD

Description:
Enables the timer and the integrator.

C Prototype:
void DELSIG11_StartAD (void)

Assembly
1call DELSIG11_StartAD

Parameters:
None

Return Value:
None

Side Effects:
The A and X registers may be modified by this or future implementations of this function. The same
is true for all RAM page pointer registers in the Large Memory Model (CY8C29xxx). When necessary,
it is the calling function's responsibility to preserve the values across calls to fastcall16 functions.
Currently, only the CUR_PP page pointer register is modified.
DELSIG11_StopAD

Description:
Disables the timer and resets the integrator.

C Prototype:
void DELSIG11_StopAD (void)

Assembly:
lcall DELSIG11_StopAD

Parameters:
None

Return Value:
None

Side Effects:
The A and X registers may be modified by this or future implementations of this function. The same
is true for all RAM page pointer registers in the Large Memory Model (CY8C29xxx). When necessary,
it is the calling function's responsibility to preserve the values across calls to fastcall16 functions.

DELSIG11_fIsDataAvailable

Description:
Checks the availability of sampled data.

C Prototype:
BYTE DELSIG11_fIsDataAvailable(void)

Assembly:
lcall DELSIG11_fIsDataAvailable
cmp A, 0
jz .DataNotAvailable

Parameters:
None

Return Value:
Returns a nonzero value if data has been converted and is ready to read.

Side Effects:
The A and X registers may be modified by this or future implementations of this function. The same
is true for all RAM page pointer registers in the Large Memory Model (CY8C29xxx). When necessary,
it is the calling function's responsibility to preserve the values across calls to fastcall16 functions.
Currently, only the CUR_PP page pointer register is modified.
DELSIG11_iGetData

Description:
Returns converted data. DELSIG11_fIsDataAvailable() should be called to verify that the data sample is ready. There is a possibility that the returned data is corrupted if the call to this function is done exactly at the end of an integration period. It is therefore highly recommended that the data retrieval be done at a higher frequency than the sampling rate, or if that cannot be guaranteed that interrupts be turned off before calling this function.

C Prototype:
INT DELSIG11_iGetData(void)

Assembly:
lcall DELSIG11_iGetData ; LSB will be in A, MSB in X upon return

Parameters:
None

Return Value:
Returns the converted data sample in 8-bit 2's complement format.

Side Effects:
The A and X registers may be modified by this or future implementations of this function. The same is true for all RAM page pointer registers in the Large Memory Model (CY8C29xxx). When necessary, it is the calling function's responsibility to preserve the values across calls to fastcall16 functions. Currently, only the CUR_PP page pointer register is modified.

DELSIG11_ClearFlag

Description:
Resets the data available flag.

C Prototype:
void DELSIG11_ClearFlag(void)

Assembly:
lcall DELSIG11_ClearFlag

Parameters:
None

Return Value:
None

Side Effect:
The global variable DELSIG11_bfStatus is set to zero. The A and X registers may be modified by this or future implementations of this function. The same is true for all RAM page pointer registers in the Large Memory Model (CY8C29xxx). When necessary, it is the calling function's responsibility to preserve the values across calls to fastcall16 functions. Currently, only the CUR_PP page pointer register is modified.
DELSIG11_iGetDataClearFlag

Description:

Returns converted data and resets the data available flag. DELSIG11_fIsDataAvailable() should be called to verify that the data sample is ready. There is a possibility that the returned data is corrupted if the call to this function is done exactly at the end of an integration period. It is therefore highly recommended that the data retrieval be done at a higher frequency than the sampling rate, or if that cannot be guaranteed that interrupts be turned off before calling this function.

C Prototype:

INT DELSIG11_iGetDataClearFlag(void)

Assembly:

lcall DELSIG11_iGetDataClearFlag ; LSB will be in A, MSB in X upon return

Parameters:

None

Return Value:

Returns the converted data sample in 8-bit 2’s complement format.

Side Effects:

The global variable DELSIG11_bfStatus is set to zero. The A and X registers may be modified by this or future implementations of this function. The same is true for all RAM page pointer registers in the Large Memory Model (CY8C29xxx). When necessary, it is the calling function’s responsibility to preserve the values across calls to fastcall16 functions. Currently, only the CUR_PP page pointer register is modified.
Sample Firmware Source Code

Example 1: Polling Enabled, Direct Access

In this example, polling is used to determine when samples become available. Direct access to the global variables is used for maximum speed. A dummy routine is called to represent further handling of the sample obtained from the DELSIG11 ADC.

The following is an assembly language example.

```assembly
include "m8c.inc" ; part specific constants and macros
include "PSoCAPI.inc" ; PSoC API definitions for all User Modules

export _main

_main:

M8C_EnableGInt ; enable global interrupts
mov A,DELSIG11_HIGHPOWER ; Establish power setting...
call DELSIG11_Start ; and initialize
call DELSIG11_StartAD ; Commence sampling process

mainloop:
cmp [DELSIG11_bfStatus], 0
jz mainloop ; spin lock until(data is Available)
mov [DELSIG11_bfStatus], 0 ; reset the data available flag
mov X, [DELSIG11_iResult+0] ; grab valid data and pass using fast-
mov A, [DELSIG11_iResult+1] ; call convention (LSB in A, MSB in X)
call ProcessSample ; pass the sample in A to the dummy fcn
jmp mainloop

ProcessSample:
    ... ; (do something useful with the data)
    ret

Here is the equivalent code in C:

```c
#include <m8c.h>        // part specific constants and macros
#include "PSoCAPI.h"    // PSoC API definitions for all User Modules

void ProcessSample( int iSample )
{
    ; // (Do something useful with the data)
}

void main(void)
{
    extern BYTE DELSIG11_bfStatus;
    extern int  DELSIG11_iResult;
    M8C_EnableGInt;
    DELSIG11_Start( DELSIG11_HIGHPOWER );
    DELSIG11_StartAD();
    while (1) {
        if ( DELSIG11_bfStatus ) {
            DELSIG11_bfStatus = 0;
            ... ; (do something useful with the data)
        }
    }
```
Example 2: Polling Enabled, Indirect Access

This example repeats the previous scenario but uses API functions rather than direct references to the global variables.

The following is an assembly language example.

```asm
include "DELSIG11.inc"
include "m8c.inc"

main:
   M8C_EnableGInt ; enable global interrupts
   mov A,DELSIG11_HIGHPOWER ; Establish power setting...
   call DELSIG11_Start ; and initialize
   call DELSIG11_StartAD ; Commence sampling process

mainloop:
   call DELSIG11_fIsDataAvailable ; Retrieve the status byte
   cmp A, 0
   jz mainloop ; spin lock until(data is Available)
   call DELSIG11_GetDataClearFlag ; fastcall convention puts data in X, A
   call ProcessSample ; pass the sample to the dummy fcn
   jmp mainloop

ProcessSample:
   ... ; (do something useful with the data)
   ret
```

Again, the equivalent code in C is:

```c
#include <m8c.h>        // part specific constants and macros
#include "PSoCAPI.h"    // PSoC API definitions for all User Modules

void ProcessSample( int iSample )
{
   ; // (Do something useful with the data)
}

void main(void)
{
   M8C_EnableGInt;
   DELSIG11_Start( DELSIG11_HIGHPOWER );
   DELSIG11_StartAD();
   while (1) {
      if ( DELSIG11_fIsDataAvailable() ) {
         ProcessSample( DELSIG11_iGetDataClearFlag() );
      }
   }
}
Configuration Registers

Table 7. Registers used by the "ADC" Analog Switched Capacitor PSoC Block

<table>
<thead>
<tr>
<th>Register</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>CR0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CR1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CR2</td>
<td>0</td>
<td>1</td>
<td>AZ</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CR3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>FSW0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The ADC is a switched capacitor PSoC block. It is configured to make an analog modulator. To build the modulator, the block is configured to be an integrator with reference feedback that converts the input value into a digital pulse stream. The input multiplexer determines what signal is digitized.

InputSource field selects the input signal digitized by the converter. This parameter is set in the Device Editor.

The AZ and FSW0 are used by the TMR interrupt handler and various APIs to reset the integrator.

Table 8. Registers used by the TMR Digital PSoC Block

<table>
<thead>
<tr>
<th>Register</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>Function</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Input</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Clock</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DR0</td>
<td>Timer Down Count Value (Never Accessed by the API)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DR1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>DR2</td>
<td>Not Used</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CR0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Enable</td>
</tr>
</tbody>
</table>

The TMR is a digital PSoC block configured to have a timer with a period of 256 counts. At the interrupt, the decimator is read and the ADC value is calculated.

Clock selects the input clock from one of 16 sources. This parameter is set in the Device Editor. Note that the source chosen must also be used to control the analog clock for the column in with the ADC block resides.

Enable empowers the TMR when set. It is modified and controlled by the DELSIG11 API.

Table 9. Decimation Control Registers

<table>
<thead>
<tr>
<th>Bit</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>DEC_CR</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>DCol</td>
<td>DCLKSEL</td>
</tr>
<tr>
<td>DEC_DH</td>
<td>High Byte Output of Decimator</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEC_DL</td>
<td>Low Byte Output of Decimator</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The decimator is dedicated hardware used to implement a Sinc2 filter needed for a ΔΣADC. It consists of a control register and two data output registers. When the value in DR0 counts down to terminal count, an
interrupt is called to decrement a higher value software counter and CNT reloads from DR1. The data is output through DR2.

DCol selects which column comparator is connected. DCLKSEL selects which digital block is used to control the decimator timing. Both parameters are set in Device Editor.

**Version History**

<table>
<thead>
<tr>
<th>Version</th>
<th>Originator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2</td>
<td>DHA</td>
<td>Added DRC to check if:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. The source clock is different between digital and analog resources.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. The ADC Clock is higher than CPU Clock.</td>
</tr>
<tr>
<td>3.2.b</td>
<td>MYKZ</td>
<td>Added design rules check for the situation when the ADC clock is faster than 8 MHz.</td>
</tr>
</tbody>
</table>

**Note**  
PSoC Designer™ 5.1 introduces a Version History in all user module datasheets. This section documents high level descriptions of the differences between the current and previous user module versions.