6 to 14-Bit Delta Sigma ADC Datasheet DelSigMultiV 1.30

Modulator Order	Decimation Rate	Resolution	Sample Rate (CLK= 2 MHz)	Sample Rate (CLK= 8 MHz)	No. Decimators	SC Blocks	Flash	RAM	Channels (I/O Pins)
CY8C28x45, CY8C28x43, CY8C28x52, CY8C28x33, CY8C28x23									
1	32	6	15625.0	62500.0	2	2	148	3	2
1	64	7.5	7812.5	31250.0	2	2	156	3	2
1	128	9	3906.3	15625.0	2	2	185	5	2
1	256	10.5	1953.1	7812.5	2	2	185	5	2
2	32	8	15625.0	62500.0	2	2	187	5	2
2	64	10	7812.5	31250.0	2	2	216	7	2
2	128	12	3906.3	15625.0	2	2	216	7	2
2	256	14	1953.1	7812.5	2	2	216	7	2
1	32	6	15625.0	62500.0	3	6	180	6	3
1	64	7.5	7812.5	31250.0	3	6	192	6	3
1	128	9	3906.3	15625.0	3	6	234	9	3
1	256	10.5	1953.1	7812.5	3	6	234	9	3
2	32	8	15625.0	62500.0	3	6	215	6	3
2	64	10	7812.5	31250.0	3	6	257	9	3
2	128	12	3906.3	15625.0	3	6	257	9	3
2	256	14	1953.1	7812.5	3	6	257	9	3
1	32	6	15625.0	62500.0	4	8	200	7	4
1	64	7.5	7812.5	31250.0	4	8	216	7	4
1	128	9	3906.3	15625.0	4	8	271	11	4
1	256	10.5	1953.1	7812.5	4	8	271	11	4
2	32	8	15625.0	62500.0	4	8	243	7	4
2	64	10	7812.5	31250.0	4	8	298	11	4
2	128	12	3906.3	15625.0	4	8	298	11	4
2	256	14	1953.1	7812.5	4	8	298	11	4

See the application note “Analog - ADC Selection” AN2239 for other converters.

Features and Overview

- 6-bit to 14-bit resolution
- Two to four channels of synchronized sampling
- Data in unsigned or signed 2’s complement formats
- Maximum sample rates of 65,500 sps at 6 bit resolution, 7812 sps at 14-bit resolution
- Sinc^2 filter fully implemented in hardware reduces CPU overhead and anti-alias requirements
- First order or second order modulator for improved signal-to-noise ratio, user selectable
- Input range defined by internal and external reference options
- Requires no digital blocks
- Configuration wizard enables you to easily select between two, three, or four channels of delta-sigma ADC measurements that are all synchronized with each other
- The internal timer of the decimators allows no digital block use

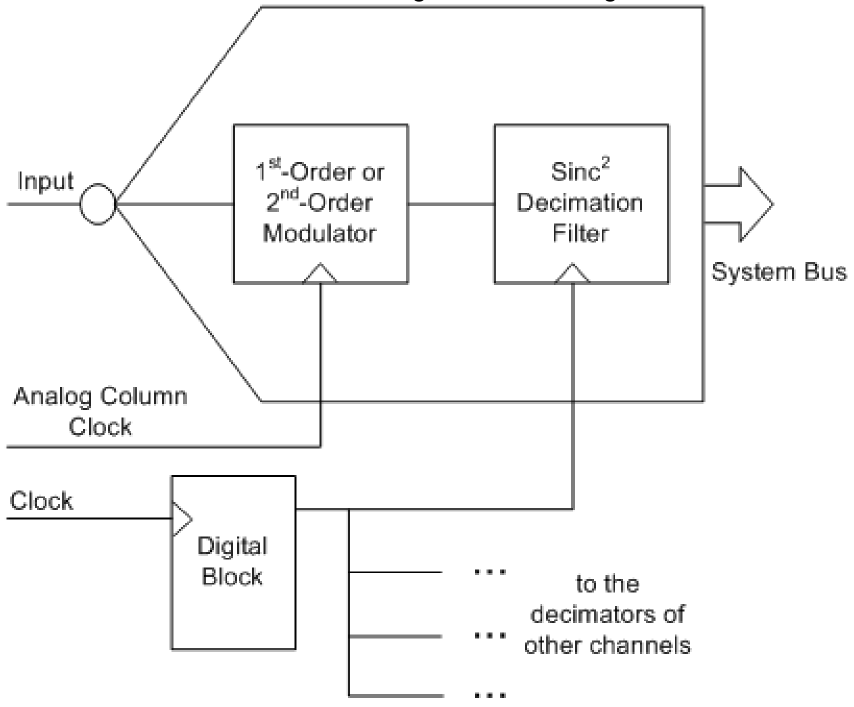
The DelSigMulti User Module is an integrating converter, requiring from 32 to 256 integration cycles to generate a single output sample. Changing multiplexed inputs invalidates the first two samples after the change. This DelSigMulti User Module supports up to four channels of simultaneous, synchronized delta-sigma ADC sampling.

A configuration wizard allows you to easily select the number of analog blocks that are used by each channel and the decimator oversample rate of each channel.

Read the Parameters section before placing the module.

Note The regular "DelSig" or "DelSigPlus" User Module should be used when only one channel or multiple unsynchronized channels are required.

Figure 1. One Channel of the DelSigMulti Block Diagram



Functional Description

As shown in Figure 1, the DelSigMulti User Module is composed of three primary functions:

- A modulator
- A Sinc^2 decimation filter
- A timing generator

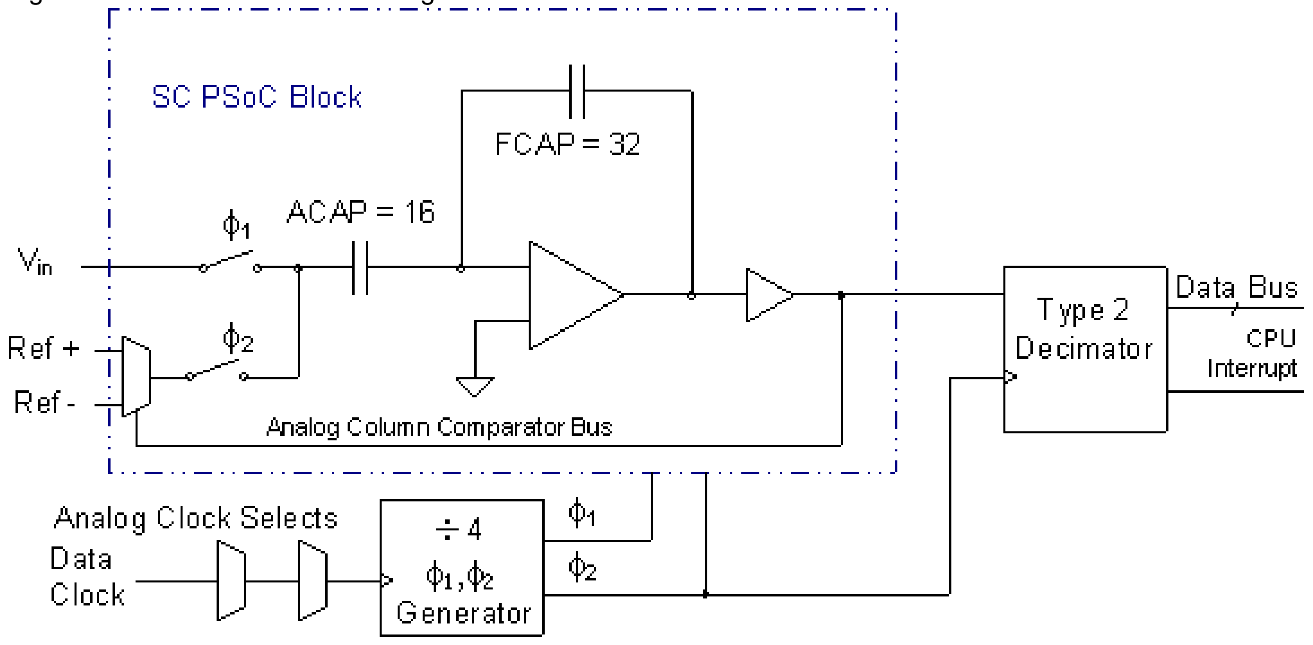
Each component offers options that may be tailored to find the right balance between performance and resource use for a given application.

Modulator

The modulator is a 1-bit over-sampling circuit that represents the input voltage in terms of the density of 1's and 0's that it produces. The modulator output is reduced to the final sample rate by the low-pass decimation filter that converts multiple 1-bit samples into samples of higher resolution. In general, higher decimation rates (that is, higher oversample rates) can produce higher resolution results, but other factors such as the order of the modulator, also matter.

A key benefit of delta-sigma converters is the "noise shaping" given by the modulator. Normally, the quantization noise inherent in sampling a signal is more or less evenly distributed ("white") in frequency between "DC" and one-half the sample frequency or Nyquist frequency. Simply put, the delta-sigma modulator shifts some of the quantization noise from lower to higher frequencies that are later attenuated by the decimation filter. A second order modulator that requires two switched-capacitor analog PSoC blocks does a better job of noise shaping than the first order modulator that only requires one analog PSoC block. At the highest decimation rate of 256X, a second order modulator accounts for a 3.5-bit increase in the effective resolution compared to a first order modulator.

Figure 2. Schematic of the DelSigMulti First Order Modulator



The analog block is configured as an integrator. The output polarity of the comparator configures reference multiplexer so the reference voltage is either added or subtracted 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 also fed into the decimator sinc^2 filter.

Note that the 1-bit oversample rate is determined by the divide-by-four generator that produces the φ_1 and φ_2 clocks that control the switched-capacitor (SC) PSoC block. The output rate is determined by dividing the data by 4 to get the 1-bit oversample rate and further dividing by the decimation rate to get the final sample rate.

Equation 1

$$SampleRate = \frac{DataClockFrequency}{4 \times DecimationRate} \text{ samples per second}$$

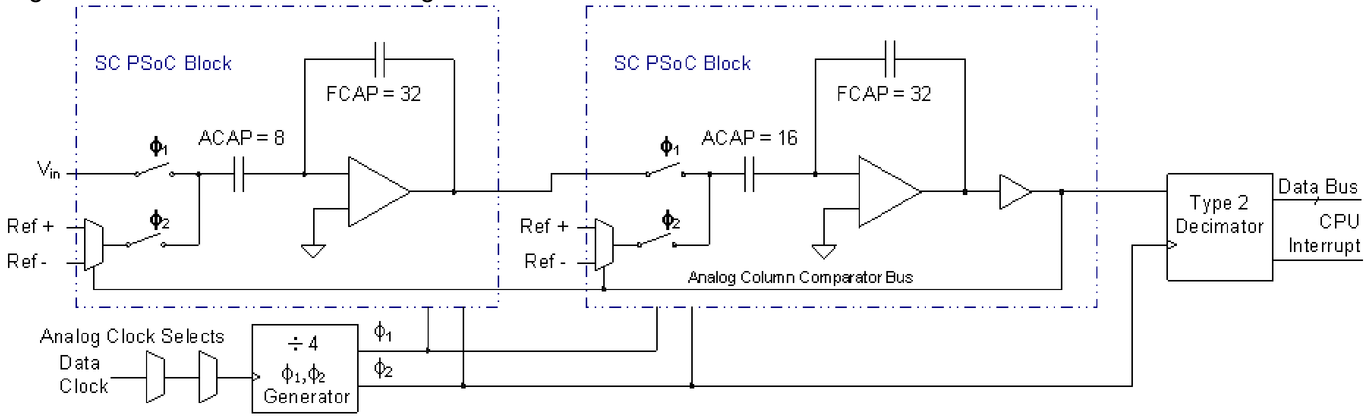
The highest data clock frequency that can be used is given in the following specification tables. For a data clock of 8 MHz, and a decimation rate of 256, the sample rate is:

Equation 2

$$8 \times 10^6/(4 \times 256) = 7812.5sps$$

A second order modulator is constructed by feeding the analog output of a first order modulator into a similar PSoC block and modifying the feedback arrangement so that the 1-bit comparator output of the second block back into both blocks as illustrated here:

Figure 3. Schematic of the DelSigMulti Second Order Modulator



Because the analog comparator buses run vertically in the columns of the analog PSoC block array, the blocks of a second order modulator must be positioned one above the other.

The range of the DelSigMulti is established by $\pm V_{Ref}$. You set V_{Ref} in the Global Resources window of PSoC Designer. For fixed scale, V_{Ref} is set to $\pm V_{Bandgap}$ or, for the CY8C29x66family of PSoC Devices, $\pm 1.6\ V_{Bandgap}$. For adjustable scale, V_{Ref} is set to ±Port 2[6]. To supply a ratiometric scale, V_{Ref} is set to $\pm V_{DD}/2$. The complete list of options is given in Table 1:

Table 1. Input Voltage Ranges for the Ref Mux Global Parameter Setting

RefMux Setting	V_{DD} = 5 Volts	V_{DD} = 3.3 Volts
(V_{DD}/2) ± BandGap	$1.2 < V_{in} < 3.8$	$0.35 < V_{in} < 2.95$
(V_{DD}/2) ± (Vdd/2)	$0 < V_{in} < 5$	$0 < V_{in} < 3.3$
BandGap ± BandGap	$0 < V_{in} < 2.6$	$0 < V_{in} < 2.6$
(1.6*BandGap) ± (1.6*BandGap)	$0 < V_{in} < 4.16$	NA

RefMux Setting	V_{DD} = 5 Volts	V_{DD} = 3.3 Volts
(2*BandGap) ± BandGap	$1.3 < V_{in} < 3.9$	NA
(2*BandGap) ± P2[6]	$(2.6 - V_{P2[6]}) < V_{in} < (2.6 + V_{P2[6]})$	NA
P2[4] ± BandGap	$(V_{P2[4]} - 1.3) < V_{in} < (V_{P2[4]} + 1.3)$	$(V_{P2[4]} - 1.3) < V_{in} < (V_{P2[4]} + 1.3)$
P2[4] ± P2[6]	$(V_{P2[4]}-V_{P2[6]}) < V_{in} < (V_{P2[4]}+V_{P2[6]})$	$(V_{P2[4]}-V_{P2[6]}) < V_{in} < (V_{P2[4]}+V_{P2[6]})$

Sinc^2 Decimation Filter

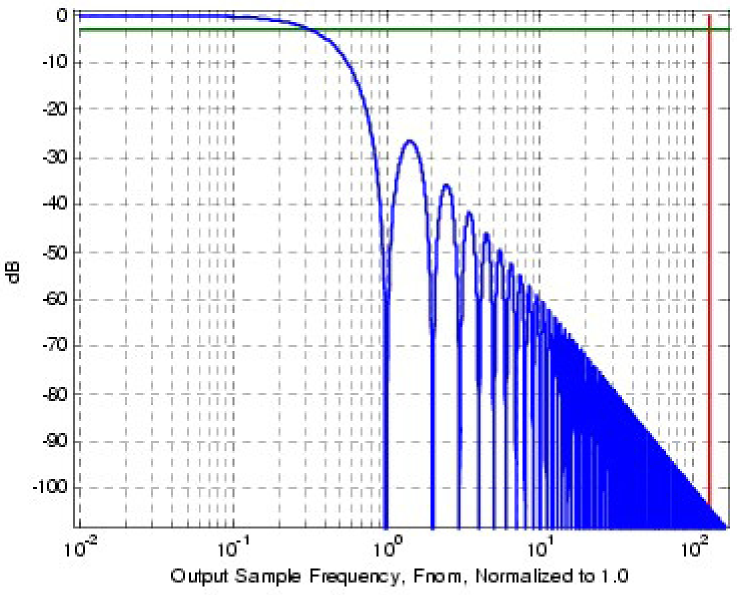
The response of the decimation filter is given by this z-domain relation:

Equation 3

$$H(z) = \left[\frac{1 - z^{-n}}{1 - z^{-1}}\right]^2, \text{ where n is the decimation level.}$$

The frequency domain transfer function plotted in this section normalizes the frequency so the output sample rate, F_{nom}, equals 1.0. The -3 dB point occurs just above $0.318 \times F_{nom}$ and zeros of the function occur at each integer multiple of F_{nom}. Since the 1-bit sample rate is 32 to 256 higher than the nominal output rate, the Nyquist limit is 4 to 7 octaves above F_{nom}, significantly reducing the requirements for an anti-alias filter. The 1-bit Nyquist frequency for a decimation rate of 256 is shown by the heavy vertical line at the right of the graph. Though higher decimation rates are possible, they contribute little additional benefit because of the noise floor of the device. In the case of the 14-bit topology, a second order modulator with a decimation rate of 256, the resolution is limited by the signal-to-noise ratio. To obtain repeatable 14-bit resolution in the measurement of DC or slow-moving signals, it is necessary to average multiple output samples or apply more sophisticated signal processing techniques.

Figure 4. Sinc^2 Decimation Filter Magnitude Response, with -3 dB point and Nyquist Frequency



Unlike the earlier DELSIG8 and DELSIG11, this user module implements both the numerator and denominator of the transfer function entirely in hardware. This requires the improved "Type 2" decimator. It is used for both the first and second order modulator topologies. The decimator implements the denominator of the transfer function by a double integrator operating at the 1-bit sample rate. The numerator is implemented by a double differentiator (second difference operator) that runs at the nominal output sample rate. The CPU overhead and interrupt latency consumed by the DelSigMulti User Module is limited to the approximately 80 cycles or less required to retrieve the sample data from the decimator registers in I/O space. The Type 2 decimator natively produces an unsigned value ranging from 0 to 2^n-1 for an n-bit converter. The interrupt service routine can be configured to convert this into a 2's complement value ranging from -2^{n-1} to $+2^{n-1}-1$.

Table 2. Features Table of Delta Sigma ADCs

Feature	Delta Sigma ADC			
	DELSIG8, DELSIG11	**DelSig**	**DelSigPlus**	**DelSigMulti**
Resolution	8, 11	6-14	6-14	6-14
Digital blocks	1	1-2	0	0
Analog blocks	1-2	1-2	1-2	2-8
Supported parts	CY8C24/27/29, not CY8C24x94, CY8C28x45	CY8C24x94, CY8C29xxx, CY8C28x45	CY8C24x94, CY8C28x45	CY8C28x45
The CPU overhead and interrupt latency	high	low	low	low

Timing Generator and Requirements

The divide-by-four clock generator that supplies the φ_1 and φ_2 clocks to the analog modulator also gives a bit-clock to the decimator. The decimation factor corresponding output sample rate is determined by a word clock. The word clock is generated by decimator internal timer.

The type2 decimator is a fully hardware version of a $Sinc^2$ filter. Its architecture allows the you the option of using an internal timer for decimation and interrupt purposes. To compute the effective resolution the following equations are used:

Single Modulator: (log2(DecimatorRate) - 1) × 1.5

Double Modulator: (log2(DecimatorRate) - 1) × 2

DataFormat bit can be weighted as signed (2s complement output) or unsigned (offset binary data).

Table 3. Decimator Data Output Shift

Decimation Rate	Modulator Type	Effective Resolution	Shift
32	Single	6	4
32	Double	8	2
64	Single	8 (7.5)	4
64	Double	10	2

Decimation Rate	Modulator Type	Effective Resolution	Shift
128	Single	9	5
128	Double	12	2
256	Single	11(10.5)	5
256	Double	14	2

DC and AC Electrical Characteristics

The following values are indicative of expected performance and based on initial characterization data. Unless otherwise specified TA = -40, 25, 85, and 125 °C, Vdd = 5.0 V.

Table 4. 5.0 V Results Summary

Parameter	Typ	Limit	Units	Remarks
8 bits, 24 MHz CPU Clk, 1 MHz data clock, High Power				
Gain	-2.6482	2	%FSR	
Offset	-47.0072	13	mV	
DNL	0.161	<1	LSB	
INL	0.27	-	LSB	
SNR	45.86	-	dB	
8 bits, 24 MHz CPU Clk, 2 MHz data clock, High Power				
Gain	-2.3168	2	%FSR	
Offset	-62.3507	13	mV	
DNL	0.069	<1	LSB	
INL	0.172	-	LSB	
SNR	45.86	-	dB	

The following values are indicative of expected performance and based on initial characterization data. Unless otherwise specified, TA = -40, 25, 85, and 125 °C, Vdd = 3.3 V.

Table 5. 3.3 V Results Summary

Parameter	Typ	Limit	Units	Remarks
8 bits, 24 MHz CPU Clk, 1 MHz data clock, High Power				
Gain	-2.7182	2	%FSR	
Offset	-40.1334	5	mV	
DNL	-	<1	LSB	
INL	-	-	LSB	
SNR	-	-	dB	
8 bits, 24 MHz CPU Clk, 2 MHz data clock, High Power				
Gain	-2.8219	2	%FSR	
Offset	-42.8073	5	mV	
DNL	0.064	<1	LSB	
INL	0.161	-	LSB	
SNR	46.02	-	dB	

Placement

When the DelSigMulti User Module is selected in the tool bar or by double clicking its icon in the selector view, a selection window opens that gives guidance in selecting the appropriate topology. The topology may be changed at any later time by right clicking on the user module in the placement view and choosing "User Module Selection Options..." from the context menu.

The first order modulator design requires one PSoC analog block. The analog block, named "ADC" may be placed in any switched capacitor PSoC block.

The second order modulator design uses two switched capacitor PSoC blocks, ADC0 and ADC1. Because the analog comparator bus that connects them runs vertically in each column of the analog array, the switched capacitor PSoC blocks must be placed vertically, one above the other.

Although there are a number of placements possible for the analog blocks, the DelSigMulti also uses the PSoC device's only hardware decimation filter. The decimator is automatically allocated when the analog blocks are placed; no additional action is necessary. Because of this, only one instance of the DelSigMulti User Module may be placed in a given configuration. With dynamic reconfiguration it is possible to load (activate) more than one configuration at a time and there is no check performed that would prevent two DelSigMulti User Modules from operating at the same time. If this occurs, both instances may appear to work; however, only the instance most recently loaded controls the decimation filter. Both interrupts may still operate, possibly interfering.

Parameters and Resources

After a DelSigMulti instance is placed, several parameters must be configured for proper operation: the Input Signal Multiplexer selection, the Clock Phase, and the Polling selection.

DataFormat0/1/2/3

This parameter may take the values of Unsigned (default) or Signed. Unsigned data takes values from zero to 2^n-1 for n-bits of resolution. Signed data ranges in value from -2^{n-1} to $+2^{n-1}$-1. 0/1/2/3 stands for the number of delta-sigma ADC channel used.

Clock Phase0/1/2/3

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 (φ_1, φ_2) to acquire and transfer signals. Normally, the input to the DelSigMulti is sampled on φ_1. A problem arises in that many of the user modules autozero their output during φ_1 and only give a valid output during φ_2. If such a module's output is fed to the DelSigMulti's input, the DelSigMulti samples an indeterminate value. The Clock Phase selection allows the phases to be swapped, so that the input signal is acquired during φ_2. 0/1/2/3 stands for the number of delta-sigma ADC channels used.

PosInput0/1/2/3

This parameter determines the signal source for singe-ended inputs, or the noninverting input for differential inputs. 0/1/2/3 stands for the number of delta-sigma ADC channels used.

NegInput0/1/2/3 and NegInputGain0/1/2/3

NegInput selects the source for the inverting input of a differential signal pair. When a single-ended input is used, this parameter may be set to any legal value. It is disconnected from the converter by setting the NegInputGain parameter to "Disconnected" (zero gain).

NegInputGain adjusts the gain of the inverting input (see NegInput parameter, above) relative to the noninverting input. For a single-ended input, this parameter should take the value "Disconnected. For differential inputs the NegInputGain can be set to 1.000. If desired, the gain applied to the inverting input can also be adjusted in 1/16th increments between 0.0625 and 1.9375 relative to the noninverting input. 0/1/2/3 stands for the number of delta-sigma ADC channel used.

Interrupt Generation Control

There are two additional parameters that become 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:

- Interrupt API
- IntDispatchMode

InterruptAPI

The InterruptAPI parameter allows conditional generation of a user module's interrupt handler and interrupt vector table entry. Select "Enable" to generate the interrupt handler and interrupt vector table entry. Select "Disable" to bypass the generation of the interrupt handler and interrupt vector table entry.

Pay particular attention to this if your project has multiple overlays where a single block resource is used by the different overlays. Choose to generate interrupts only for the overlays that actually need them to conserve code space.

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. When you select ActiveStatus the firmware tests 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. When you selecting OffsetPreCalc the firmware calculates 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 given 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 given by the "include" files.

Each time a user module is placed, it is assigned an instance name. By default, PSoC Designer assigns the DelSigMulti_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 DelSigMulti for simplicity.

Note

In this, as in all user module APIs, you can modify the values of the A and X registers 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.

DelSigMulti_Start

Description:

Performs all required initialization for this user module and sets the power level for the switched capacitor PSoC block. All channels are configured at the same power level.

C Prototype:

```
void  DelSigMulti_Start(BYTE bfPowerSetting)
```

Assembly:

```
mov   A, bfPowerSetting
lcall  DelSigMulti_Start
```

Parameters:

bPowerSetting:One byte that has four 2-bit bitfields that each individually specifies the power level of each channel. Following reset and configuration, the analog PSoC blocks assigned to DelSigMulti are powered down. Symbolic names given in C and assembly, and their associated values are listed in the following table.

Symbolic Name	Mask
DelSigMulti_CH0_OFF	00h
DelSigMulti_CH0_LOWPOWER	01h
DelSigMulti_CH0_MEDPOWER	02h
DelSigMulti_CH0_HIGHPOWER	03h
DelSigMulti_CH1_OFF	00h
DelSigMulti_CH1_LOWPOWER	04h
DelSigMulti_CH1_MEDPOWER	08h
DelSigMulti_CH1_HIGHPOWER	0Ch
DelSigMulti_CH2_OFF	00h
DelSigMulti_CH2_LOWPOWER	10h
DelSigMulti_CH2_MEDPOWER	20h
DelSigMulti_CH2_HIGHPOWER	30h
DelSigMulti_CH3_OFF	00h
DelSigMulti_CH3_LOWPOWER	40h
DelSigMulti_CH3_MEDPOWER	80h
DelSigMulti_CH3_HIGHPOWER	C0h

Return Value:

None

Side Effects:

You can modify the A and X registers by this or future implementations of this function. The same is true for all RAM page pointer registers in the Large Memory Model. When necessary, it is the calling function's responsibility to preserve the values across calls to fastcall16 functions.

DelSigMulti_Stop

Description:

Sets the power level to the switched capacitor PSoC blocks to OFF.

C Prototype:

```
void  DelSigMulti_Stop(void)
```

Assembly:

```
lcall  DelSigMulti_Stop
```

Parameters:

None

Return Value:

None

Side Effects:

You can modify the A and X registers by this or future implementations of this function. The same is true for all RAM page pointer registers in the Large Memory Model. When necessary, it is the calling function's responsibility to preserve the values across calls to fastcall16 functions.

DelSigMulti_SetPower

Description:

Sets the power level for the switched capacitor PSoC blocks for each channel of the UM.

C Prototype:

```
void  DelSigMulti_SetPower(BYTE bPowerSetting)
```

Assembly:

```
mov   A, bPowerSetting
lcall  DelSigMulti_SetPower
```

Parameters:

bPowerSetting: One byte that has four 2-bit bitfields that each individually specifies the power level of each channel. Following reset and configuration, the analog PSoC blocks assigned to DelSigMulti are powered down. Symbolic names given in C and assembly, and their associated values are listed in the following table.

Symbolic Name	Mask
DelSigMulti_CH0_OFF	00h
DelSigMulti_CH0_LOWPOWER	01h
DelSigMulti_CH0_MEDPOWER	02h
DelSigMulti_CH0_HIGHPOWER	03h
DelSigMulti_CH1_OFF	00h
DelSigMulti_CH1_LOWPOWER	04h
DelSigMulti_CH1_MEDPOWER	08h
DelSigMulti_CH1_HIGHPOWER	0Ch
DelSigMulti_CH2_OFF	00h
DelSigMulti_CH2_LOWPOWER	10h
DelSigMulti_CH2_MEDPOWER	20h
DelSigMulti_CH2_HIGHPOWER	30h

Symbolic Name	Mask
DelSigMulti_CH3_OFF	00h
DelSigMulti_CH3_LOWPOWER	40h
DelSigMulti_CH3_MEDPOWER	80h
DelSigMulti_CH3_HIGHPOWER	C0h

Return Value:

None

Side Effects:

You can modify the A and X registers by this or future implementations of this function. The same is true for all RAM page pointer registers in the Large Memory Model. When necessary, it is the calling function's responsibility to preserve the values across calls to fastcall16 functions.

DelSigMulti_StartAD

Description:

Activates interrupts for this user module and begins sampling.

C Prototype:

```
void  DelSigMulti_StartAD(void)
```

Assembly

```
lcall  DelSigMulti_StartAD
```

Parameters:

None

Return Value:

None

Side Effects:

You can modify the A and X registers by this or future implementations of this function. The same is true for all RAM page pointer registers in the Large Memory Model. When necessary, it is the calling function's responsibility to preserve the values across calls to fastcall16 functions.

DelSigMulti_StopAD

Description:

Shuts down the A/D by interrupt disabling. Analog power is still supplied to the analog block.

C Prototype:

```
void  DelSigMulti_StopAD(void)
```

Assembly:

```
lcall  DelSigMulti_StopAD
```

Parameters:

None

Return Value:

None

Side Effects:

You can modify the A and X registers by this or future implementations of this function. The same is true for all RAM page pointer registers in the Large Memory Model. When necessary, it is the calling function's responsibility to preserve the values across calls to fastcall16 functions.

DelSigMulti_fIsDataAvailable

Description:

Checks whether new ADC sampled data is ready. A flag is set when new results from ADC are ready. This API function allows the user to check the flag. The data availability is checked for the whole UM. All channels are synchronized, so all data is new or not new at the same time. Therefore, only one check is made that applies to all sampled data of all channels. The user must clear this flag using another API function.

C Prototype:

```
BYTE  DelSigMulti_fIsDataAvailable(void)
```

Assembly:

```
lcall  DelSigMulti_fIsDataAvailable
cmp   A, 0
jz    .DataNotAvailable
```

Parameters:

None

Return Value:

Returns a nonzero value if each channel’s data has been converted and is ready to read. Returns a zero if each channel’s data has not been converted and is not ready.

Side Effects:

You can modify the A and X registers by this or future implementations of this function. The same is true for all RAM page pointer registers in the Large Memory Model. 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.

DelSigMulti_GetAllDataClearFlag

Description:

This function is used to retrieve channel’s data with one function. A RAM array pointer is passed into the function and the function places the ADC results in this RAM array. The user must ensure that their RAM array is the correct size for the number and resolution of their ADC channels. This function also clears the flag for those signals for which the ADC data is ready. DelSigMulti_fIsDataAvailable() may be called to verify that the data sample is ready before the function is called.

C Prototype:

```
void DelSigMulti_GetAllDataClearFlag(BYTE* pbRamBuffer)
```

Assembly:

```
mov   A, >pbRamBuffer
mov   X, <pbRamBuffer
lcall  DelSigMulti_GetAllDataClearFlag
```

Parameters:

pbRamBuffer: This parameter determines the RAM location to which the ADC sample data is copied. The data that is copied are the ADC results. The user must ensure that the RAM array type matches the data type of the ADC results. The ADC channel data is placed in the user's array starting with the channel 0 data first and the byte order is MSB first. For example, if the UM has three channels of unsigned 10-bit data, this array copies six bytes to the user's array in the following order: Ch0_Result_MSB, Ch0_Result_LSB, Ch1_Result_MSB, Ch1_Result_LSB, Ch2_Result_MSB, Ch2_Result_LSB. If the user has an array type that is different than the BYTE type, the user should cast the array pointer passed into the function to be a BYTE pointer.

Return Value:

None

Side Effects:

You can modify the A and X registers by this or future implementations of this function. The same is true for all RAM page pointer registers in the Large Memory Model. 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.

DelSigMulti_cGetData

DelSigMulti_iGetData

Description:

Returns converted data as a signed 8-bit or 16-bit 2's complement format. Note that the user module DataFormat parameter determines the underlying representation. Calling a signed format function does not change the value of the data when the underlying representation is unsigned. DelSigMulti_fIsDataAvailable() may be called to verify that the data sample is ready. The channel number is passed into this function.

C Prototypes:

```
CHAR DelSigMulti_cGetData(BYTE bChannelNumber) // use for 8-bit resolution or lower
INT  DelSigMulti_iGetData(BYTE bChannelNumber) // use for 9-bit resolution or higher
```

Assembly:

```
mov   A, [bChannelNumber]
lcall  DelSigMulti_cGetData      ; Result will be in A
- or -
mov   A, [bChannelNumber]
lcall  DelSigMulti_iGetData      ; LSB will be in A, MSB in X upon return
```

Parameters:

bChannelNumber: This parameter determines which channel's data is returned.

Return Value:

Returns the converted data sample in 8-bit or 16-bit 2's complement format.

Side Effects:

You can modify the A and X registers by this or future implementations of this function. The same is true for all RAM page pointer registers in the Large Memory Model. 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.

DelSigMulti_bGetData

DelSigMulti_wGetData

Description:

Returns converted data as an 8-bit or 16-bit unsigned format. Note that the user module DataFormat parameter determines the underlying representation. Calling an unsigned format function does not change the value of the data when the underlying representation is signed. DelSigMulti_fIsDataAvailable() may be called to verify that the data sample is ready. The channel number is passed into this function.

C Prototypes:

```
BYTE DelSigMulti_bGetData(BYTE bChannelNumber) // use for 8-bit resolution or lower
WORD DelSigMulti_wGetData(BYTE bChannelNumber) // use for 9-bit resolution or higher
```

Assembly:

```
mov   A, [bChannelNumber]
lcall  DelSigMulti_bGetData       ; Result will be in A
- or -
mov   A, [bChannelNumber]
lcall  DelSigMulti_wGetData       ; LSB will be in A, MSB in X upon return
```

Parameters:

bChannelNumber: This parameter determines which channel's data is returned.

Return Value:

Returns the converted data sample in 8-bit or 16-bit unsigned format according to the function.

Side Effects:

You can modify the A and X registers by this or future implementations of this function. The same is true for all RAM page pointer registers in the Large Memory Model. 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.

DelSigMulti_ClearFlag

Description:

Resets the Data Available flag.

C Prototype:

```
void  DelSigMulti_ClearFlag(void)
```

Assembly:

```
lcall  DelSigMulti_ClearFlag
```

Parameters:

None

Return Value:

None

Side Effect:

You can modify the A and X registers by this or future implementations of this function. The same is true for all RAM page pointer registers in the Large Memory Model. 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

Note The DelSigMulti User Module should be configured as a two-channel with resolution not more than 8 bits for every channel.

Here is an assembly language example:

```
include "m8c.inc"       ; part specific constants and macros
include "memory.inc"    ; Constants & macros for SMM/LMM and Compiler
include "PSoCAPI.inc"   ; PSoC API definitions for all User Modules

    ; two bytes are required for this example code
BUFF_SIZE: equ 2;

AREA userRAM(RAM,REL)
    ; reserve a 2-bytes area in RAM to load data in
RamBuffer: blk BUFF_SIZE

AREA text(ROM, REL)
export _main

_main:
   M8C_EnableGInt                     ; enable global interrupts
   mov   A,DelSigMulti_CH0_HIGHPOWER | DelSigMulti_CH1_HIGHPOWER        ; Establish
power setting...
   call  DelSigMulti_Start                 ; and initialize
   call  DelSigMulti_StartAD               ; Commence sampling process
mainloop:
   call  DelSigMulti_fIsDataAvailable   ; Retrieve the status byte
   cmp   A, 0
   jz    mainloop                     ; spin lock until(data is Available)
   mov   A, >RamBuffer
   mov   X, <RamBuffer
   call  DelSigMulti_GetAllDataClearFlag   ; fastcall convention places ADC results in
RAM array defined by RamBuffer pointer
   call  ProcessSample                ; pass the sample to the dummy fcn
   jmp   mainloop

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

The equivalent code in C:

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

BYTE RamBuffer[2];

void main(void)
{
    M8C_EnableGInt;
        // start both channels of DelSigMulti ADC
   DelSigMulti_Start(DelSigMulti_CH0_HIGHPOWER|DelSigMulti_CH1_HIGHPOWER);
   DelSigMulti_StartAD();

    while(1)
    {       //  check if data sample is available
        if( DelSigMulti_fIsDataAvailable() )
        {
            DelSigMulti_GetAllDataClearFlag(RamBuffer); // copy ADC data to buffer
before processing
}
   }
}
```

Configuration Registers

The following section details the registers altered by this user module.

Analog Registers, First Order Modulator

Table 6. Registers used by the "ADC" Analog Switched Capacitor PSoC Block

Register	7	6	5	4	3	2	1	0
ADC_CH0/1/2/3_CR0	1	0	0	1	0	0	0	0
ADC_CH0/1/2/3_CR1	PosInput0/1/2/3			InvertingGain0/1/2/3				
ADC_CH0/1/2/3_CR2	0	1	0	0	0	0	0	0
ADC_CH0/1/2/3_CR3	1	1	1	0	NegInput0/1/2/3		Power	

PosInput0/1/2/3 selects the single-ended input signal or the noninverting input of a differential input signal. NegInput0/1/2/3 selects the inverting input of a differential input. The inverting input is disconnected when ever the InvertingGain0/1/2/3 field is set to zero. Power is set by the DelSigMulti_Start and DelSigMulti_SetPower API functions. 0/1/2/3 stands for the number of delta-sigma ADC channel used.

Analog Registers, Second Order Modulator

Table 7. Registers used by the "ADC0" and "ADC1" Analog Switched Capacitor PSoC Block

Register	7	6	5	4	3	2	1	0
ADC0_CH0/1/2/3_CR0	1	0	0	0	1	0	0	0
ADC0_CH0/1/2/3_CR1	PosInput0/1/2/3			InvertingGain0/1/2/3				
ADC0_CH0/1/2/3_CR2	0	1	0	0	0	0	0	0
ADC0_CH0/1/2/3_CR3	1	1	1	0	NegInput0/1/2/3		Power	
ADC1_CH0/1/2/3_CR0	1	0	0	1	0	0	0	0
ADC1_CH0/1/2/3_CR1	LinkToADC0			0	0	0	0	0
ADC1_CH0/1/2/3_CR2	0	0	0	0	0	0	0	0
ADC1_CH0/1/2/3_CR3	1	1	1	0	0	0	Power	

PosInput0/1/2/3 selects the single-ended input signal or the noninverting input of a differential input signal. NegInput0/1/2/3 selects the inverting input of a differential input. The inverting input is disconnected when ever the InvertingGain0/1/2/3 field is set to zero. LinktoADC0 is determined by block placement and connects the output of the ADC0 block to the "A" input capacitor of the ADC1 PSoC block. Power is set by the DelSigMulti_Start and DelSigMulti_SetPower API functions. 0/1/2/3 stands for the number of delta-sigma ADC channel used.

Decimator Control Registers

Table 8. Decimation Control Registers

Bit	7	6	5	4	3	2	1	0
DECx_DH	Data High Byte							
DECx_DL	Data Low Byte							
DEC_CR0	ACC_IGEN				ICLKS	ACE_IGEN		DCLKS0
DEC_CR1	0	IDEC	ICLKS3	ICLKS2	ICLKS1	DCLKS3	DCLKS2	DCLKS1
DECx_CR0	POL	GOOO	GOOE	0		DATA_IN		
DEC_CR3	DEC1_EN	CLK_IN1			DEC0_EN	CLK_IN0		
DEC_CR4	DEC3_EN	CLK_IN3			DEC2_EN	CLK_IN2		
DEC_CR5	0				DSCLK			
DECx_CR	Mode		Data Out Shift		Data Format	Decimation Rate		

The decimator is dedicated hardware used to implement a Sinc2 filter. It consists of three control registers and two data output registers. 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. Shift, in DEC_CR2, is set according to the decimation rate, also specified in DEC_CR2, to minimize the data aligned that must be accomplished in software.

Digital Block Registers

Table 9. Digital Block Control Registers

Bit	7	6	5	4	3	2	1	0
DxCxxCR0								Enable

This register is used in configuration with addition digital block to start generation clock for this UM by calling function DelSigMulti_StartAD. Also this register is used for stopping generation clock by calling function DelSigMulti _StopAD.

Version History

Version	Originator	Description
1.2	DHA	Added DRC to check if the source clock is different in digital and analog resources. Improved GetAllDataClearFlagAPI realization. Improved clock selection DRC.
1.2.b	DHA	Added help file to wizard. Updated Sample Code in user module datasheet.
1.30	MYKZ	1. Implemented the ability to place analog user module blocks in non-adjacent columns. 2. Added design rules check for the situation when the ADC clock is faster than 8 MHz.

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.