Delta Sigma ADC Datasheet DelSigPlus V 1.10

Copyright © 2008-2013 Cypress Semiconductor Corporation. All Rights Reserved.

Resources	PSoC® Blocks				API Memory (Bytes)		Pins (per External I/O)
	Digital	Analog CT	Analog SC	Decimator Column	Flash	RAM	
CYC8C24x94, CY8CLED0xD, CY8CLED0xG, CY8C28x45, CY8C28x43							
6, First Order, 32	0	1	0	1	84	2	1
7.5, First Order, 64	0	1	0	1	88	2	1
9, First Order, 128	0	1	0	1	107	3	1
10.5, First Order, 256	0	1	0	1	109	3	1
8, Second Order, 32	0	2	0	1	99	2	1
10, Second Order, 64	0	2	0	1	118	3	1
12, Second Order, 128	0	2	0	1	118	3	1
14, Second Order, 256	0	2	0	1	118	3	1

Note “Decimator Column” resource is available only for CY8C28x45 device.

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

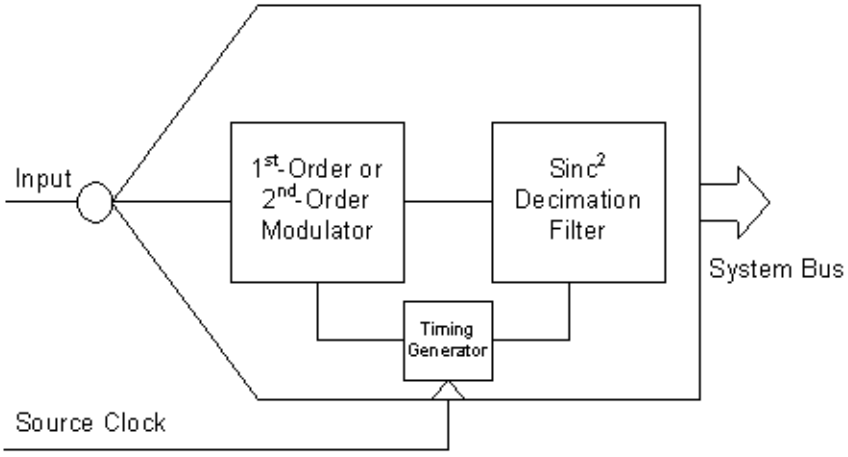
For one or more fully configured, functional example projects that use this user module go to www.cypress.com/psocexampleprojects.

Features and Overview

- 6-bit to 14-bit resolution
- 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

The DelSigPlus 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 following the change. Refer to the Parameters section prior to module placement.

Figure 1. DelSigPlus Block Diagram



Functional Description

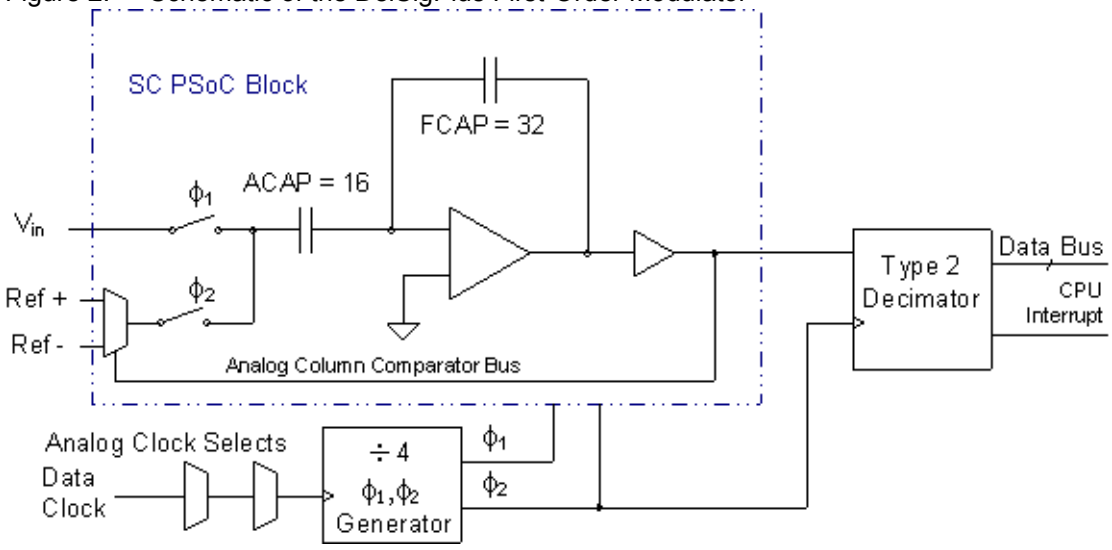
As shown in the block diagram, the DelSigPlus User Module is composed of three primary functions: a modulator, a Sinc^2 decimation filter, and a timing generator. Each component offers options that may be tailored to find the right balance between performance and resource use for a given application.

Modulator

The modulator is a 1-bit oversampling 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" provided 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 into 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 DelSigPlus 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 φ1 and φ2 s 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

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

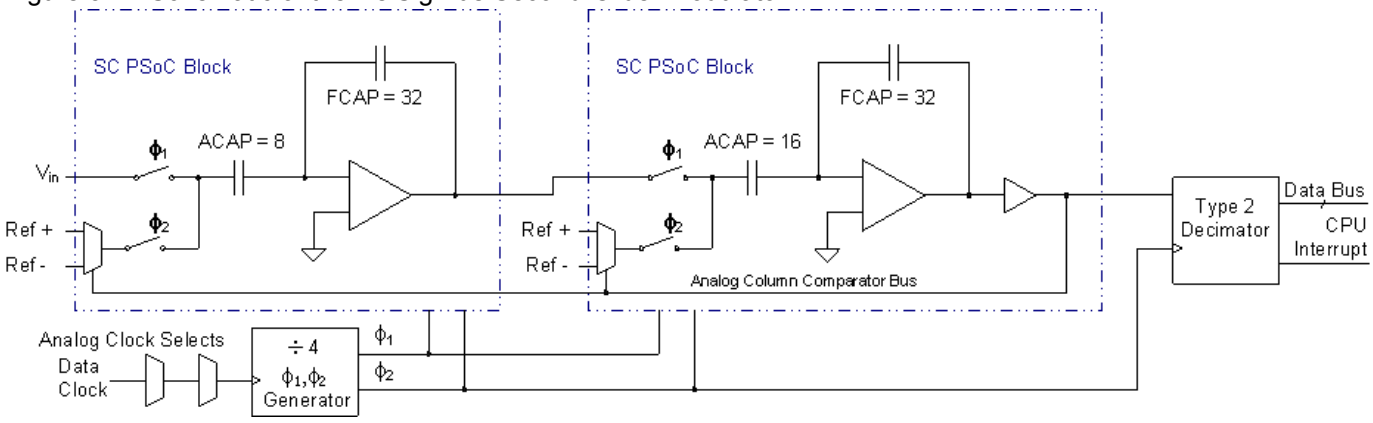
The highest data clock frequency that can be used is given in the following specification tables. For a data clock of 8 MHz, and a decimation rate of 256, the sample rate is:

Equation 2

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

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 below.

Figure 3. Schematic of the DelSigPlus 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 DelSigPlus 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 the following table.

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

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

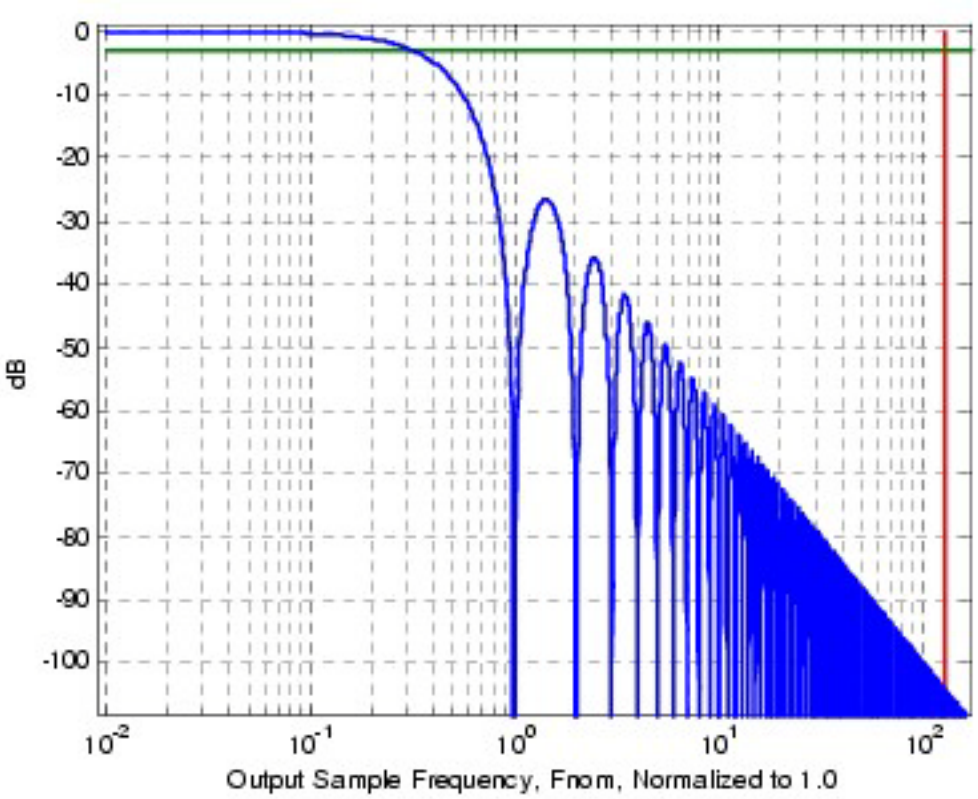
The response of the decimation filter is given by the following z-domain relation.

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 the following figure 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 DelSigPlus 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
Resolution	8, 11	6-14	6-14
Digital blocks	1	1-2	0
Analog bocks	1-2	1-2	1-2
Supported parts	CY8C24/27/29, not CY8C24x94	CY8C24x94, CY8C29xxx	CY8C24x94
The CPU overhead and interrupt latency	high	low	low

Timing Generator and Requirements

The divide-by-four clock generator that supplies the φ1 and φ2 clocks to the analog modulator also provides 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 Sinc2 filter. It’s 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 below TA = -40, 25, 85, and 125 °C, Vdd = 5.0 V.

Table 4. 5.0V 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 below, 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 DelSigPlus User Module is selected in the tool bar or by double-clicking its icon in the selector view, a selection window opens that provides 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, ADC1 and ADC2. 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 DelSigPlus also utilizes 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 DelSigPlus 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 DelSigPlus User Modules from operating at the same time. If this occurs, both instances may appear to work; however, only the instance most recently loaded will control the decimation filter. Both interrupts may still operate, possibly interfering.

Parameters and Resources

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

DataFormat

This parameter may take the values of Signed (default) or Unsigned. Unsigned data will take values from zero to 2^n-1 for n-bits of resolution. Signed data will range in value from -2^{n-1} to $+2^{n-1}$-1

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 DelSigPlus is sampled on φ1. A problem arises in that many of the user modules auto-zero their output during φ1 and only provide a valid output during φ2. If such a module's output is fed to the DelSigPlus's input, the DelSigPlus will sample an indeterminate value. The Clock Phase selection allows the phases to be swapped, so that the input signal is acquired during φ2.

PosInput

This parameter determines the signal source for singe-ended inputs, or the non-inverting input for differential inputs.

NegInput and NegInputGain

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 non-inverting 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 non-inverting input.

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 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.

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

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 prior to 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 will 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.

DelSigPlus_Start

Description:

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

C Prototype:

```
void  DelSigPlus_Start (BYTE bPowerSetting)
```

Assembly:

```
mov   A, bPowerSetting
lcall  DelSigPlus_Start
```

Parameters:

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

Symbolic Name	Value
DelSigPlus_OFF	0
DelSigPlus_LOWPOWER	1
DelSigPlus_MEDPOWER	2
DelSigPlus_HIGHPOWER	3

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.

DelSigPlus_Stop

Description:

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

C Prototype:

```
void  DelSigPlus_Stop (void)
```

Assembly:

```
lcall  DelSigPlus_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.

DelSigPlus_SetPower

Description:

Sets the power level for the switched capacitor PSoC block.

C Prototype:

```
void  DelSigPlus_SetPower(BYTE bPowerSetting)
```

Assembly:

```
mov   A, bPowerSetting
lcall  DelSigPlus_SetPower
```

Parameters:

bPowerSetting: bPowerSetting:One byte that specifies the power level. Symbolic names provided in C and assembly, and their associated values are given in the following table.

Symbolic Name	Value
DelSigPlus_OFF	0
DelSigPlus_LOWPOWER	1
DelSigPlus_MEDPOWER	2
DelSigPlus_HIGHPOWER	3

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.

DelSigPlus_StartAD

Description:

Activates interrupts for this user module and begins sampling.

C Prototype:

```
void  DelSigPlus_StartAD (void)
```

Assembly

```
lcall  DelSigPlus_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.

DelSigPlus_StopAD

Description:

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

C Prototype:

```
void  DelSigPlus_StopAD(void)
```

Assembly:

```
lcall  DelSigPlus_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.

DelSigPlus_fIsDataAvailable

Description:

Checks the availability of sampled data.

C Prototype:

```
BYTE  DelSigPlus_fIsDataAvailable (void)
```

Assembly:

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

Parameters:

None

Return Value:

Returns a non-zero value if data has been converted and is ready to read.

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.

DelSigPlus_cGetData

DelSigPlus_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. DelSigPlus_fIsDataAvailable() may be called to verify that the data sample is ready.

C Prototypes:

```
CHAR DelSigPlus_cGetData (void)      // use for 8-bit resolution or lower
INT  DelSigPlus_iGetData (void)      // use for 9-bit resolution or higher
```

Assembly:

```
lcall  DelSigPlus_cGetData      ; Result will be in A
- or -
lcall  DelSigPlus_iGetData      ; LSB will be in A, MSB in X upon return
```

Parameters:

None

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.

DelSigPlus_bGetData

DelSigPlus_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. DelSigPlus_fIsDataAvailable() may be called to verify that the data sample is ready.

C Prototypes:

```
BYTE DelSigPlus_bGetData(void)      // use for 8-bit resolution or lower
WORD DelSigPlus_wGetData(void)      // use for 9-bit resolution or higher
```

Assembly:

```
lcall  DelSigPlus_bGetData       ; Result will be in A
- or -
lcall  DelSigPlus_wGetData       ; LSB will be in A, MSB in X upon return
```

Parameters:

None

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.

DelSigPlus_ClearFlag

Description:

Resets the Data Available flag.

C Prototype:

```
void  DelSigPlus_ClearFlag(void)
```

Assembly:

```
lcall  DelSigPlus_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.

DelSigPlus_cGetDataClearFlag

DelSigPlus_iGetDataClearFlag

Description:

Returns converted data 8-bit or 16-bit 2’s complement format and resets the Data Available flag. 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. DelSigPlus_fIsDataAvailable() may be called to verify that the data sample is ready.

C Prototype:

```
CHAR DelSigPlus_cGetDataClearFlag(void) //for 8-bit resolution or lower
INT  DelSigPlus_iGetDataClearFlag(void) //for 9-bit resolution or higher
```

Assembly:

```
lcall  DelSigPlus_cGetDataClearFlag  ;Result will be in A
- or -
lcall  DelSigPlus_iGetDataClearFlag  ;LSB will be in A, MSB in X upon return
```

Parameters:

None

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.

DelSigPlus_bGetDataClearFlag

DelSigPlus_wGetDataClearFlag

Description:

Returns converted data in 8-bit or 16-bit unsigned format and resets the Data Available flag. 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. DelSigPlus_fIsDataAvailable() may be called to verify that the data sample is ready.

C Prototype:

```
BYTE DelSigPlus_bGetDataClearFlag(void)  //for 8-bit resolution or lower
WORD DelSigPlus_wGetDataClearFlag(void)  //for 9-bit resolution or higher
```

Assembly:

```
lcall  DelSigPlus_bGetDataClearFlag  ;Result will be in A
- or -
lcall  DelSigPlus_wGetDataClearFlag  ;LSB will be in A, MSB in X upon return
```

Parameters:

None

Return Value:

Returns the converted data sample in 8-bit or 16-bit unsigned 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.

Sample Firmware Source Code

In this example, polling is used to determine when samples are available. A dummy routine is called to represent further handling of the sample obtained from the DelSigPlus ADC.

Here is an assembly language example:

```
include "DelSigPlus.inc"
include "m8c.inc"

export _main

_main:
   M8C_EnableGInt                  ; enable global interrupts
   mov   A,DelSigPlus_HIGHPOWER        ; Establish power setting...
   call  DelSigPlus_Start              ; and initialize
   call  DelSigPlus_StartAD            ; Commence sampling process
mainloop:
   call  DelSigPlus_fIsDataAvailable   ; Retrieve the status byte
   cmp   A, 0
   jz    mainloop                  ; spin lock until(data is Available)
   call  DelSigPlus_iGetDataClearFlag   ; 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
```

The equivalent code in 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;
   DelSigPlus_Start( DelSigPlus_HIGHPOWER );
   DelSigPlus_StartAD();
   while (1) {
      if ( DelSigPlus_fIsDataAvailable() ) {
         ProcessSample( DelSigPlus_iGetDataClearFlag() );
      }
   }
}
```

Configuration Registers

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
CR0	1	0	0	1	0	0	0	0
CR1	PosInput			InvertingGain				
CR2	0	1	0	0	0	0	0	0
CR3	1	1	0	0	NegInput		Power	

PosInput selects the single-ended input signal or the non-inverting input of a differential input signal. NegInput selects the inverting input of a differential input. The inverting input is disconnected when ever the InvertingGain field is set to zero. Power is set by the DelSigPlus_Start and DelSigPlus_SetPower API functions.

Analog Registers, Second-Order Modulator

Table 7. Registers used by the “ADC1” and “ADC2” Analog Switched Capacitor PSoC Block

Register	7	6	5	4	3	2	1	0
ADC1CR0	1	0	0	0	1	0	0	0
ADC1CR1	PosInput			InvertingGain				
ADC1CR2	0	1	0	0	0	0	0	0
ADC1CR3	1	1	0	0	NegInput		Power	
ADC2CR0	1	0	0	1	0	0	0	0
ADC2CR1	LinkToADC1			0	0	0	0	0
ADC2CR2	0	1	0	0	0	0	0	0
ADC2CR3	1	1	0	0	0	0	Power	

PosInput selects the single-ended input signal or the non-inverting input of a differential input signal. NegInput selects the inverting input of a differential input. The inverting input is disconnected when ever the InvertingGain field is set to zero. LinktoADC1 is determined by block placement and connects the output of the ADC1 block to the “A" input capacitor of the ADC2 PSoC block. Power is set by the DelSigPlus_Start and DelSigPlus_SetPower API functions.

Decimator Control Registers

Table 8. Decimation Control Registers

Bit	7	6	5	4	3	2	1	0
DEC_CR0	0	0	0	0	0	DCol		DCLKSEL
DEC_CR1	0	1	0	0	0	DCLKSEL		
DEC_CR2	1	0	Shift		1	DecimationRate		
DEC_DH	High Byte Output of Decimator							
DEC_DL	Low Byte Output of Decimator							

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.

Version History

Version	Originator	Description
1.0	DHA	Initial version
1.10	MYKZ	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 to document high level descriptions of the differences between the current and previous user module versions.

Copyright © 2008-2013 Cypress Semiconductor Corporation. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

PSoC Designer™ and Programmable System-on-Chip™ are trademarks and PSoC® is a registered trademark of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

Any Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.