デルタ シグマ ADC データシート DelSigPlus V 1.0

Copyright © 2008-2011 Cypress Semiconductor Corporation. All Rights Reserved.

リソース	PSoC® ブロック				API メモリ （バイト数）		ピン （外部入出力ごと）
	デジタル	アナログ CT	アナログ SC	デシメータコラム	フラッシュ	RAM	
CYC8C24x94, CY8CLED0xD, CY8CLED0xG, CY8C28x45, CY8C28x43							
6, 1 次 , 32	0	1	0	1	84	2	1
7.5, 1 次 , 64	0	1	0	1	88	2	1
9, 1 次 , 128	0	1	0	1	107	3	1
10.5, 1 次 , 256	0	1	0	1	109	3	1
8, 2 次 , 32	0	2	0	1	99	2	1
10, 2 次 , 64	0	2	0	1	118	3	1
12, 2 次 , 128	0	2	0	1	118	3	1
14, 2 次 , 256	0	2	0	1	118	3	1

Note 「デシメータコラム」リソースは、CY8C28x45 デバイス用としてのみ利用可能です。

その他のコンバータについては、アプリケーション ノート「アナログ - ADC の選択」AN2239 を参照してください。

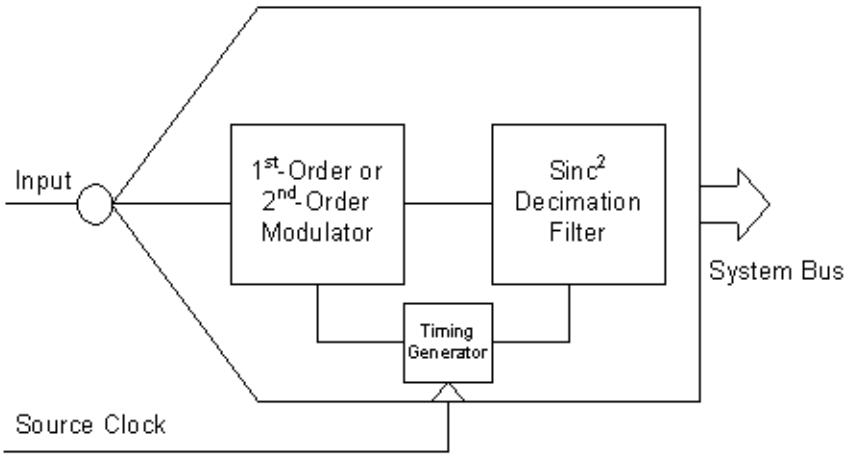
当ユーザ モジュールを使用した、構成済みの機能的なプロジェクト例は下記のウェブサイトでご覧ください。www.cypress.com/psocexampleprojects

特性および概要

- 6-bit ~ 14-bit 分解能
- 符号なしまたは符号付き 2 の補数の形式によるデータ
- 最大サンプリング速度は 6 ビット分解能で 65,500 sps、14-bit 分解能で 7812 sps
- ハードウェアに完全実装された $Sinc^N$ フィルタが、CPU オーバヘッドとアンチエイリアスの要件を軽減
- 信号対雑音比を改善する 1 次または 2 次変調器。ユーザ選択可
- 入力範囲は内部及び外部リファレンス オプションによって選択可能
- デジタルブロックが不要

DelSigPlus ユーザ モジュールは、単一の出力サンプルを生成するために 32 ～ 256 の積分サイクルを必要とする、積分型変換器です。マルチプレックス入力を変更すると、変更後最初の 2 つのサンプルが無効になります。モジュールを配置する前に、「パラメータ」のセクションを参照してください。

Figure 1. DelSigPlus ブロック ダイアグラム



機能説明

ブロック ダイアグラムに示すように、DelSigPlus ユーザ モジュールは、変調器、$Sinc^2$ デシメーション フィルタ、タイミング信号発生器という 3 つの主要機能から構成されています。各コンポーネントは、アプリケーションにおいてパフォーマンスとリソース利用の適正なバランスを実現するために選択・調整できるオプションを提供します。

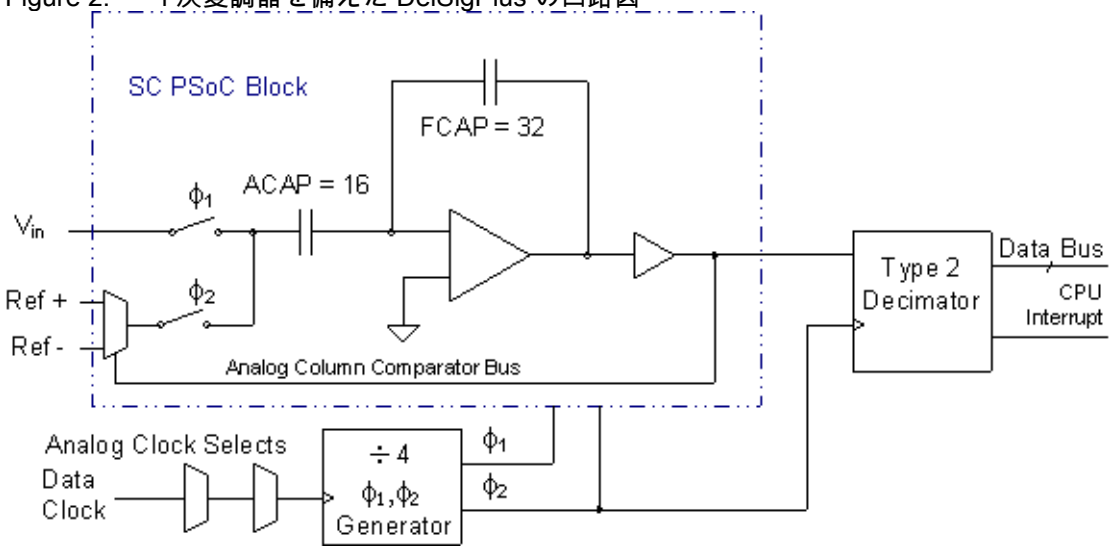
変調器

変調器は 1-bit オーバーサンプリング回路で、それが生成する 1 及び 0 の密度を単位として入力電圧を表します。変調器の出力は、複数の 1-bit サンプルをより高い分解能のサンプルに変換するローパス デシメーション フィルタにより、最終のサンプリング速度に低減されます。一般に、より高い復号化レート (つまりより高いオーバーサンプリング速度) は、より高い分解能の結果を生成することができますが、変調器の次数などその他の要因も関係します。

デルタシグマ変換器の主な利点は、変調器が行う「ノイズ シェーピング」です。通常、信号サンプリングに特有の量子化ノイズは、「DC」とサンプル周波数の半分またはナイキスト周波数の間の周波数で、ほぼ均等に分散しています (「白」)。簡単に言うと、デルタシグマ変調器は、量子化ノイズの一部を低周波数から高周波数にシフトします。これは後で、デシメーション フィルタによって減衰されます。2 つのスイッチド キャパシタ アナログ PSoC ブロックを必要とする 2 次変調器は、1 つのアナログ PSoC ブロックのみを必要とする 1 次変調器よりも、ノイズ シェーピングで優れています。256X のより高い復号化レートで 2 次変調器は、1 次変調器に比べ、実効分解能において 3.5-bit の増加が見られます。

[+] Feedback

Figure 2. 1 次変調器を備えた DelSigPlus の回路図



アナログ ブロックは、積分器として構成されます。コンパレータの出力極性はリファレンスマルチプレクサを構成し、それによってリファレンス電圧が入力に加算または入力から減算され、積分器に置かれるように構成されます。リファレンス制御は、積分器の出力をゼロに向かって引き戻そうとします。単一ビットコンパレータの出力もまた、デシメータ $sinc^2$ フィルタに入力されます。

注：1-bit オーバーサンプリング速度は、スイッチド キャパシタ PSoC ブロックを制御する φ1 と φ2 s を生成する四分割信号発生器によって決定します。出力レートは、1-bit オーバーサンプリング速度を得るためにデータを 4 で割り、最終サンプリング速度を得るためにさらにデシメーションレートで割ることによって決定します。サンプリング速度は次の式で求められます。

Equation 1

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

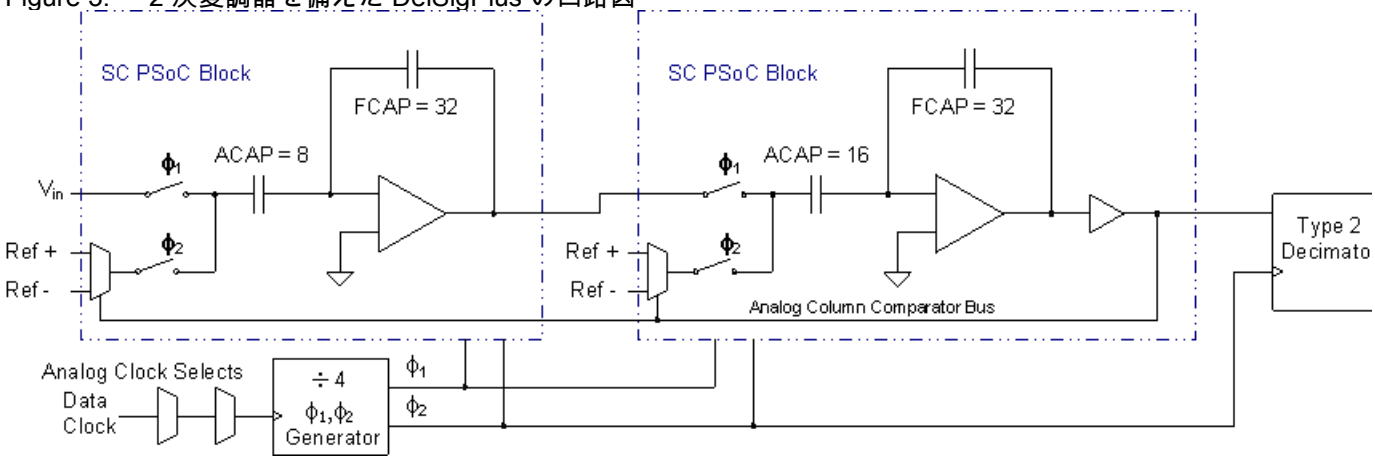
使用できるもっとも高いデータクロック周波数は、下記の仕様表で示されています。データクロックが 8MHz で、デシメーションレートが 256 の場合、サンプリング速度は次のようになります。

Equation 2

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

2 次変調器は、1 次変調器のアナログ出力を同様の PSoC ブロックにフィードし、2 つ目のブロックの 1-bit コンパレータ出力が下に示すように両方のブロックに戻るよう、フィードバック構成を変更することによって作成します。

Figure 3. 2 次変調器を備えた DelSigPlus の回路図



アナログ コンパレータのバスは、アナログ PSoC ブロック アレイのコラムで垂直に走るため、2 次変調器のブロックは、ほかのものより 1 つ上に配置する必要があります。

DelSigPlus の範囲は、$\pm V_{Ref}$ によって規定されます。V_{Ref} は PSoC Designer のグローバル リソース ウィンドウで設定します。固定スケールの場合、V_{Ref} は $\pm V_{Bandgap}$ に設定されます。PSoC デバイスの CY8C29x66 ファミリでは $\pm 1.6\ V_{Bandgap}$ です。調整可能なスケールの場合、V_{Ref} は ±Port 2[6] に設定されます。レシオメトリック スケールでは、V_{Ref} は $\pm V_{DD}/2$ に設定されます。オプションの完全なリストを以下の表に示します。

Table 1. Ref Mux グローバル パラメータ設定の入力電圧範囲

RefMux の設定	Vdd = 5 ボルト	Vdd = 3.3 ボルト
(Vdd/2) ± BandGap	$1.2 < V_{in} < 3.8$	$0.35 < V_{in} < 2.95$
(Vdd/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)	$0 < V_{in} < 4.16$	なし
(2*BandGap) ± BandGap	$1.3 < V_{in} < 3.9$	なし
(2*BandGap) ± P2[6]	$(2.6 - V_{P2[6]}) < V_{in} < (2.6 + V_{P2[6]})$	なし
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 デシメーション フィルタ

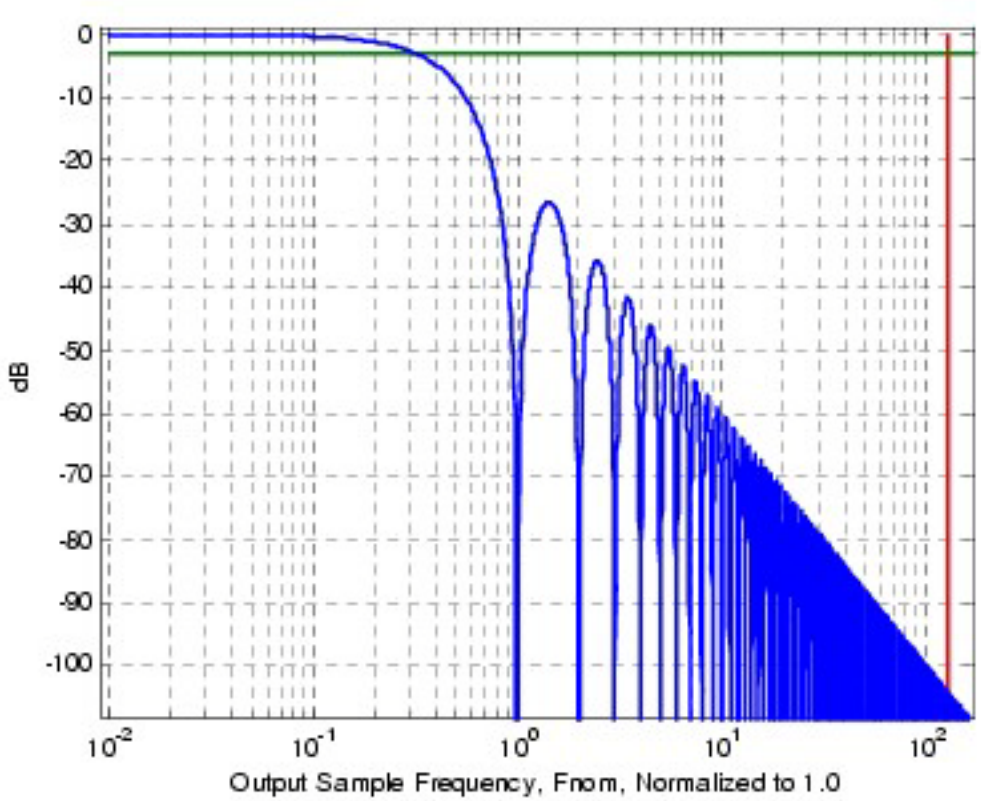
デシメーション フィルタの応答は、以下の z ドメイン関係によって得られます。

Equation 3

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

以下にプロットした周波数ドメイン伝達関数は、出力サンプリング速度 F_{nom} が 1.0 になるように正規化されます。-3 dB のポイントは 0.318×F_{nom} のすぐ上で発生し、関数は F_{nom} の各整数倍でゼロになります。1-bit サンプリング速度は定格出力速度よりも 32 ～ 256 高いため、ナイキスト制限は F_{nom} の 4 ～ 7 オクターブ上となり、アンチエイリアス フィルタの要件を著しく低減します。256 の復号化レートに対する 1-bit ナイキスト周波数は、グラフの右側に太い垂直線で示されています。さらに高い復号化レートも可能ですが、デバイスのノイズフロアのため、利点はほとんどありません。14-bit トポロジ、256 の復号化レートを持つ 2 次変調器の場合は、分解能が信号対雑音比により制限されます。DC 測定または移動速度の遅い信号でリピート可能な 14-bit 分解能を得るには、複数の出力サンプルの平均を取るか、高度な信号処理技法を採用することが必要となります。

Figure 4. -3 dB 及びナイキスト周波数を使った Sinc^2 デシメーション フィルタの振幅特性



以前の DELSIG8 と DELSIG11 とは異なり、このユーザ モジュールは伝達関数の分子と分母をすべてハードウェアで実行します。これには、改良型「タイプ 2」のデシメータが必要です。これは 1 次および 2 次の変調器トポロジに使用されます。デシメータは、1-bit サンプリング速度で動作する二重積分器により、伝達関数の分母を実装します。分子は、定格出力サンプリング速度で動作する二重微分器 (2 つ目の差分演算子) によって実装されます。DelSigPlus ユーザ モジュールが消費する CPU のオーバーヘッドと割り込みレイテンシは、I/O スペースのデシメータレジスタからサンプルデータを回復するために必要な約 80 サイクル以下に限定されています。タイプ 2 のデシメータは、n ビットコンバータ用に 0 ~ 2^n-1 の範囲の符号無しの値を生成します。割込みサービス ルーチンは、これを -2^{n-1} ~ $+2^{n-1}$-1 の範囲の 2 の補数に変換するように構成できます。

Table 2. デルタ シグマ ADC の機能表

機能	デルタ シグマ ADC		
	DELSIG8, DELSIG11	DelSig	DelSigPlus
分解能	8, 11	6-14	6-14
デジタルブロック	1	1-2	0
アナログブロック	1-2	1-2	1-2
サポートされているコンポーネント	CY8C24/27/29。 CY8C24x94 は未サポート	CY8C24x94, CY8C29xxx	CY8C24x94
CPU オーバーヘットと割り込みレイテンシ	高	低	低

タイミング信号発生器と要件

アナログ変調器に φ1 と φ2 クロックを供給する四分周信号発生器は、デシメータにビットクロックも供給します。出力サンプリング速度に対応するデシメーションファクタは、ワードクロックによって決定します。ワードクロックは、デシメータの内部タイマによって生成されます。

タイプ 2 デシメータは、Sinc2 フィルタの完全にハードウェア化されたバージョンです。そのアーキテクチャによって、デシメーションと割り込みの目的に使用できる内部タイマのオプションが可能になります。有効な分解能の計算には、次の式を使用します。

一段変調器：(log2(DecimatorRate) - 1)*1,5

二段変調器：(log2(DecimatorRate) - 1)*2

DataFormat ビットは、符号付き（2 の補正出力）と符号無し（オフセットバイナリデータ）を選択することが出来ます。

Table 3. デシメータデータの出力シフト

デシメーションレート	変調器タイプ	有効な分解能	シフト
32	一段	6	4
32	二段	8	2
64	一段	8 (7.5)	4
64	二段	10	2

[+] Feedback

デシメーションレート	変調器タイプ	有効な分解能	シフト
128	一段	9	5
128	二段	12	2
256	一段	11(10.5)	5
256	二段	14	2

DC 電気的特性と AC 電気的特性

以下の値は初期の特性データを基に、予測される性能を入れておきます。別途記載がない限り、下記では T_A = -40、25、85、125°C および V_{dd} = 5.0V を想定しています。

Table 4. 5.0V 結果の概要

パラメータ	標準値	制限	単位	備考
8 ビット、24 MHz CPU Clk、1 MHz データクロック、High Power				
ゲイン	-2.6482	2	%FSR	
オフセット	-47.0072	13	mV	
DNL	0.161	<1	LSB	
INL	0.27	--	LSB	
SNR	45.86	--	dB	
8 ビット、24 MHz CPU Clk、2 MHz データクロック、High Power				
ゲイン	-2.3168	2	%FSR	
オフセット	-62.3507	13	mV	
DNL	0.069	<1	LSB	
INL	0.172	--	LSB	
SNR	45.86	--	dB	

以下の値は初期の特性データを基に、予測される性能を入れておきます。別途記載がない限り、下記では T_A = -40、25、85、125°C および V_{dd} = 3.3V を想定しています。

Table 5. 3.3V 結果の概要

パラメータ	標準値	制限	単位	備考
8 ビット、24MHz CPU Clk、1 MHz データクロック、High Power				
ゲイン	-2.7182	2	%FSR	
オフセット	-40.1334	5	mV	
DNL		<1	LSB	
INL		--	LSB	
SNR		--	dB	
8 ビット、24MHz CPU Clk、2 MHz データクロック、High Power				
ゲイン	-2.8219	2	%FSR	
オフセット	-42.8073	5	mV	
DNL	0.064	<1	LSB	
INL	0.161	--	LSB	
SNR	46.02	--	dB	

配置

ツールバーで DelSigPlus ユーザ モジュールを選択するか、セレクタービューでそのアイコンをダブルクロックすると、選択ウィンドウが開き、適切なトポロジを選択するガイダンスを提供します。トポロジは、配置ビューでユーザ モジュールを右クリックして コンテンツメニューで [User Module Selection Options...] （ユーザ モジュール選択オプション）を選択すると変更できます

1 次変調器を設計するには、1 つの PSoC アナログブロックが必要です。アナログ ブロック 「ADC」は、任意のスイッチド キャパシタ PSoC ブロックに配置できます。

2 次変調器デザインは、2 つのスイッチド キャパシタ PSoC ブロック、ADC1、ADC2 を利用しています。これらを連結するアナログコンパレータバスがアナログアレイの各コラムを縦に通っているため、スイッチド キャパシタ PSoC ブロックは縦に上下させて並べなければなりません。

アナログ ブロックでは多数の配置が可能ですが、DelSigPlus は、PSoC デバイス唯一のハードウェア デシメータフィルタも使用します。デシメータは、アナログブロックが配置されたときに自動的に割り当てられます。ユーザによる操作は不要です。このため、DelSigPlus ユーザ モジュールのインスタンスは 1 つだけ配置できます。ダイナミックな再構成の場合、一度に 2 つ以上の構成を読み込む（有効にする）ことが可能ですが、2 つの DelSigPlus ユーザ モジュールが同時に稼動することを抑止するためのチェックは行われません。この状態では、両方のインスタンスが稼動しているように見えますが、一番最近読み込んだインスタンスがデシメーションフィルタを制御します。ただし、両方の割り込みが実行され、干渉をきたす場合もあります。

パラメータおよびリソース

DelSigPlus のインスタンスが配置されると、適正に作動するように幾つかのパラメータを構成する必要がでてきます (入力信号マルチプレクサ選択、クロック位相、ポーリング選択)。

データ 形式

このパラメータは、符号無し (デフォルト) または符号付きの値をとることがあります。n ビットの分解能の場合、符号無しデータはゼロから 2^n-1 の値です。符号付きデータは、-2^{n-1} ~ $+2^{n-1}$-1 の間の値です。

クロック位相

クロック位相の選択は、あるアナログ PSoC ブロックの出力を別のアナログ PSoC ブロックの入力と同期させるために使用します。スイッチド キャパシタ アナログ PSoC ブロックは、2 相クロック (φ1、φ2) を使用して、信号を取得および転送します。通常、DelSigPlus への入力は、φ1 でサンプリングされます。ユーザ モジュールの多くは、φφ1 中に出力を自動的にゼロに設定し、φφ2 中しか有効な出力を供給しないため、問題が発生します。そのようなモジュールの出力が DelSigPlus 入力に配信された場合、DelSigPlus は不定値をサンプリングします。クロック位相を選択することで、入力信号を φ2 中に取得するように、位相を交換できます。

PosInput

このパラメータは、シングルエンド入力の信号ソース、または差動入力の非反転入力を決定します。

NegInput および NegInputGain

NegInput は、差動信号ペアの反転入力用ソースを選択します。シングルエンドの入力を使用した場合、このパラメータには任意の値を設定できます。NegInputGain パラメータを [Disconnected] (接続解除) (ゼロゲイン) に設定すると、コンバータからの接続を解除できます。

NegInputGain は非反転入力と比例して、反転入力のゲインを調整します (上述の 「NegInput」セクションを参照してください)。シングルエンドの入力の場合、このパラメータは [Disconnected] (接続解除) となります。差動入力では、NegInputGain は 1.000 に設定できます。必要に応じて、反転入力に適用されたゲインは、NegInputGain は非反転入力と比例して、0.0625 ~ 1.9375 の範囲で 1/16th の増分を用いて調整できます。

割り込み生成制御

PSoC Designer で、**[Enable interrupt generation control (割り込み生成の制御を有効にする)]** チェックボックスが選択されている場合は、さらに 2 つのパラメータが利用できるようになります。これは **[Project (プロジェクト)] > > [Settings (設定)] > > [Chip Editor (チップ エディタ)] >** を開いてアクセスします。「割り込み生成の制御」は、オーバーレイ全体で複数のユーザ モジュールにより共有される割り込みとともに、複数のオーバーレイが使用される場合に重要です。

- 割り込み API
- IntDispatchMode

InterruptAPI

InterruptAPI パラメータを使うと、ユーザ モジュールの割り込みハンドラと割り込みベクトル テーブル エントリの状況に応じた生成が可能になります。「Enable (有効)」を選択すると、割り込みハンドラと割り込みベクトル テーブル エントリが生成されます。「Disable (無効)」を選択すると、割り込みハンドラと割り込みベクトル テーブル エントリがバイパスされます。

複数のオーバーレイがあり、1 つのブロックリソースが異なるオーバーレイで使用されているプロジェクトの場合、特に注意が必要です。コードスペースを節約するために実際にこれを必要とするオーバーレイの場合のみ、割り込みを生成するよう選択します。

IntDispatchMode

IntDispatchMode パラメータを使用して、同一ブロック内の異なるオーバレイ内に存在する複数のユーザ モジュールで共用している割り込みについて、割り込みをどのように処理するかを指定します。ActiveStatus を選択すると、ファームウェアが、共有される割り込みリクエストに応答する前に、どちらのオーバーレイがアクティブであるかをテストします。このテストは、共用割り込みが要求されるたびに行われます。このためにレイテンシが付加され、共用割り込み要求を処理する非決定性のプロセスも生じますが、RAM は不要です。OffsetPreCalc を選択すると、ファームウェアが、オーバーレイが最初にロードされるときだけ、共有割り込みリクエストのソースを計算するようになります。この計算によって割り込みレイテンシは減少し、共用割り込み要求を処理する決定性のプロセスが生じますが、これは RAM のバイトを消費します。

アプリケーション プログラミング インタフェース

アプリケーション プログラミング インタフェース (API) ルーチンは設計者がより高度なレベルでモジュールを処理できるようにユーザ モジュールの一部として提供されます。このセクションでは、「include」ファイルによって提供される各機能に対するインタフェースおよび定数を示します。

ユーザ モジュールを配置するたびに、インスタンス名が割り当てられます。デフォルトでは、PSoC Designer プロジェクトで、このユーザ モジュールの最初のインスタンスに DelSigPlus_1 を割り当てます。これは識別子の構文ルールに従った一意の値に変更できます。割り当てたインスタンス名が、全てのグローバル関数名、変数、および定数記号の接頭語になります。次の説明では、簡単にするために、インスタンス名は省略されて単に 「DelSigPlus」となっています。

Note ここでは、全てのユーザ モジュール API と同じように、API 関数を呼び出すことで A と X レジスタの値が変更されることがあります。API をコールした後でも A と X の値を保持したいときは、API をコールするファンクションで A と X の値を保持する必要があります。PSoC Designer のバージョン 1.0 以降、効率性の観点から、この「registers are volatile（レジスタの揮発性）」ポリシーが採用されています。C コンパイラは自動的にこの条件を処理します。アセンブラ言語のプログラマは、コードがこのポリシーを遵守していることも確認しなければなりません。一部のユーザーモジュール API 関数では A と X は変更されないこともありますが、将来も変更されないという保証はありません。

DelSigPlus_Start

説明：

このユーザ モジュールで必要な全ての初期化を実行し、スイッチド キャパシタ PSoC ブロックの出力レベルを設定します。

C プロトタイプ：

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

アセンブラ：

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

パラメータ：

bPowerSetting: 出力レベルを指定する 1 バイト。リセットとコンフィグレーションの後、ADCINC に割り当てられたアナログ PSoC ブロックの電力が遮断されます。C およびアセンブリで用意されたシンボル名、およびそれらに関連付けられた値は、以下の表で示されます。

記号名	値
DelSigPlus_OFF	0
DelSigPlus_LOWPOWER	1
DelSigPlus_MEDPOWER	2
DelSigPlus_HIGHPOWER	3

戻り値：

なし

副作用

A および X レジスタは、今回、または今後、この関数を実装することによって変更される可能性があります。大容量メモリ モデル（c）のすべての RAM ページ ポインタ レジスタにも同じことが言えます。必要に応じて、fastcall16 関数の呼び出しでこれらの値を保存してください。

DelSigPlus_Stop

説明：

スイッチド キャパシタ PSoC ブロックへの出力レベルをオフに設定します。

C プロトタイプ：

```
void  DelSigPlus_Stop (void)
```

アセンブラ：

```
lcall  DelSigPlus_Stop
```

パラメータ：

なし

戻り値：

なし

副作用

A および X レジスタは、今回、または今後、この関数を実装することによって変更される可能性があります。大容量メモリ モデル（c）のすべての RAM ページ ポインタ レジスタにも同じことが言えます。必要に応じて、fastcall16 関数の呼び出しでこれらの値を保存してください。

DelSigPlus_SetPower

説明：

スイッチド キャパシタ PSoC ブロックの出力レベルを設定します。

C プロトタイプ：

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

アセンブラ：

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

パラメータ：

bPowerSetting: bPowerSetting: 出力レベルを指定する 1 バイト。C およびアセンブリで用意されたシンボル名、およびそれらに関連付けられた値は、以下の表で示されます。

記号名	値
DelSigPlus_OFF	0
DelSigPlus_LOWPOWER	1
DelSigPlus_MEDPOWER	2
DelSigPlus_HIGHPOWER	3

戻り値：

なし

副作用

A および X レジスタは、今回、または今後、この関数を実装することによって変更される可能性があります。大容量メモリ モデル（c）のすべての RAM ページ ポインタ レジスタにも同じことが言えます。必要に応じて、fastcall16 関数の呼び出しでこれらの値を保存してください。

DelSigPlus_StartAD

説明：

このユーザ モジュール用に割り込みを有効にし、サンプリングを開始します。

C プロトタイプ：

```
void  DelSigPlus_StartAD (void)
```

アセンブラ：

```
lcall  DelSigPlus_StartAD
```

パラメータ：

なし

戻り値：

なし

副作用

A および X レジスタは、今回、または今後、この関数を実装することによって変更される可能性があります。大容量メモリ モデル（c）のすべての RAM ページ ポインタ レジスタにも同じことが言えます。必要に応じて、fastcall16 関数の呼び出しでこれらの値を保存してください。

DelSigPlus_StopAD

説明：

割り込みの無効化により、A/D をシャットダウンします。ただし、アナログブロックへのアナログ出力供給は続行されます。

C プロトタイプ：

```
void  DelSigPlus_StopAD(void)
```

アセンブラ：

```
lcall  DelSigPlus_StopAD
```

パラメータ：

なし

戻り値：

なし

副作用

A および X レジスタは、今回、または今後、この関数を実装することによって変更される可能性があります。大容量メモリ モデル（c）のすべての RAM ページ ポインタ レジスタにも同じことが言えます。必要に応じて、fastcall16 関数の呼び出しでこれらの値を保存してください。

DelSigPlus_fIsDataAvailable

説明：

サンプリングしたデータが使用できるかどうかチェックします。

C プロトタイプ：

```
BYTE  DelSigPlus_fIsDataAvailable (void)
```

アセンブラ：

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

パラメータ：

なし

戻り値：

データが変換され、読み取れる状態の場合は、ゼロ以外の値を返します。

副作用

A および X レジスタは、今回、または今後、この関数を実装することによって変更される可能性があります。大容量メモリ モデル（c）のすべての RAM ページ ポインタ レジスタにも同じことが言

えます。必要に応じて、fastcall16 関数の呼び出しでこれらの値を保存してください。現時点では , CUR_PP ページ ポインタ レジスタのみが変更されています。

DelSigPlus_cGetData

DelSigPlus_iGetData

説明 :

変換されたデータを符号付き 8-bit または 16-bit2 の補数形式で返します。注 : ユーザ モジュールの DataFormat パラメータが内部表現を決定します。内部表現が符号無しの無しの場合、符号付きの形式関数を呼び出しても、データの値は変りません。データサンプルの準備ができていることを確認するために、DelSigPlus_fIsDataAvailable() が呼び出されることもあります。

C プロトタイプ :

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

アセンブラ :

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

パラメータ :

なし

戻り値 :

変換したデータ サンプルを 8-bit または 16-bit の 2 の補数の形で返します。

副作用

A および X レジスタは、今回、または今後、この関数を実装することによって変更される可能性があります。大容量メモリ モデル (c) のすべての RAM ページ ポインタ レジスタにも同じことが言えます。必要に応じて、fastcall16 関数の呼び出しでこれらの値を保存してください。現時点では , CUR_PP ページ ポインタ レジスタのみが変更されています。

DelSigPlus_bGetData

DelSigPlus_wGetData

説明 :

変換されたデータを 8-bit または 16-bit 符号無し形式で返します。注 : ユーザ モジュールの DataFormat パラメータが内部表現を決定します。内部表現が符号無しの無しの場合、符号付きの形式関数を呼び出しても、データの値は変りません。データサンプルの準備ができていることを確認するために、DelSigPlus_fIsDataAvailable() が呼び出されることもあります。

C プロトタイプ :

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

アセンブラ :

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

パラメータ :

なし

戻り値 :

関数に従い、8-bit または 16-bit の変換されたデータ サンプルを符号なしの形式で返します。

副作用

A および X レジスタは、今回、または今後、この関数を実装することによって変更される可能性があります。大容量メモリ モデル (c) のすべての RAM ページ ポインタ レジスタにも同じことが言えます。必要に応じて、fastcall16 関数の呼び出しでこれらの値を保存してください。現時点では , CUR_PP ページ ポインタ レジスタのみが変更されています。

DelSigPlus_ClearFlag

説明 :

データ利用可能フラグをリセットします。

C プロトタイプ :

```
void  DelSigPlus_ClearFlag(void)
```

アセンブラ :

```
lcall  DelSigPlus_ClearFlag
```

パラメータ :

なし

戻り値 :

なし

副作用

A および X レジスタは、今回、または今後、この関数を実装することによって変更される可能性があります。大容量メモリ モデル (c) のすべての RAM ページ ポインタ レジスタにも同じことが言えます。必要に応じて、fastcall16 関数の呼び出しでこれらの値を保存してください。現時点では , CUR_PP ページ ポインタ レジスタのみが変更されています。

DelSigPlus_cGetDataClearFlag

DelSigPlus_iGetDataClearFlag

説明 :

変換したデータ サンプルを 8-bit または 16-bit の 2 の補数の形で返し、データ使用可能フラグをリセットします。注 : ユーザ モジュールの DataFormat パラメータが内部表現を決定します。内部表現が符号無しの無しの場合、符号付きの形式関数を呼び出しても、データの値は変りません。データサンプルの準備ができていることを確認するために、DelSigPlus_fIsDataAvailable() が呼び出されることもあります。

C プロトタイプ :

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

アセンブラ :

```
lcall  DelSigPlus_cGetDataClearFlag  ;Result will be in A
```

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

パラメータ :

なし

戻り値 :

変換したデータ サンプルを 8-bit または 16-bit の 2 の補数の形で返します。

副作用

A および X レジスタは、今回、または今後、この関数を実装することによって変更される可能性があります。大容量メモリ モデル (c) のすべての RAM ページ ポインタ レジスタにも同じことが言えます。必要に応じて、fastcall16 関数の呼び出しでこれらの値を保存してください。現時点では , CUR_PP ページ ポインタ レジスタのみが変更されています。

DelSigPlus_bGetDataClearFlag

DelSigPlus_wGetDataClearFlag

説明 :

変換したデータ サンプルを 8-bit または 16-bit の符号無しの形で返し、データ使用可能フラグをリセットします。注 : ユーザ モジュールの DataFormat パラメータが内部表現を決定します。内部表現が符号無しの場合、符号付きの形式関数を呼び出しても、データの値は変りません。データサンプルの準備ができていることを確認するために、DelSigPlus_fIsDataAvailable() が呼び出されることもあります。

C プロトタイプ :

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

アセンブラ :

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

パラメータ :

なし

戻り値 :

8-bit または 16-bit 符号なし整数の形式で、変換されたデータ サンプルを返します。

副作用

A および X レジスタは、今回、または今後、この関数を実装することによって変更される可能性があります。大容量メモリ モデル (c) のすべての RAM ページ ポインタ レジスタにも同じことが言えます。必要に応じて、fastcall16 関数の呼び出しでこれらの値を保存してください。現時点では , CUR_PP ページ ポインタ レジスタのみが変更されています。

ファームウェア ソースコードの例

この例では前のシナリオを繰り返しますが、API 変数ではなく、グローバル変数への直接リファレンスを使用しています。

アセンブリ言語の例をここに示します。

```
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
```

同じコードを C 言語で示します。

```
#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() );
      }
   }
}
```

コンフィグレーション レジスタ

アナログレジスタ、1 次変調器

Table 6. 「ADC」アナログ スイッチド キャパシタ PSoC ブロックによって使用されるレジスタ

レジスタ	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 (出力)	

PosInput はシングルエンドの入力信号または差動入力信号の非反転入力を選択します。NegInput は、差動信号の反転入力を選択します。InvertingGain フィールドがゼロに設定されているとき、反転入力は接続が解除されます。出力は、DelSigPlus_Start と DelSigPlus_SetPowerAPI 関数によって設定されます。

アナログレジスタ、2 次変調器

Table 7. 「ADC1」と「ADC2」アナログ スイッチド キャパシタ PSoC ブロックによって使用されるレジスタ

レジスタ	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 (出力)	
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 (出力)	

PosInput はシングルエンドの入力信号または差動入力信号の非反転入力を選択します。NegInput は、差動信号の反転入力を選択します。InvertingGain フィールドがゼロに設定されているとき、反転入力は接続が解除されます。LinktoADC1 は、ブロック配置によって決定し、ADC1 ブロックの出力を ADC2 PSoC ブロックの 「A」入力キャパシタに接続します。出力は、DelSigPlus_Start と DelSigPlus_SetPowerAPI 関数によって設定されます。

デシメーション制御レジスタ

Table 8. デシメーション制御レジスタ

ビット	7	6	5	4	3	2	1	0
DEC_CR0	0	0	0	0	0	DCol		DCLKSEL

ビット	7	6	5	4	3	2	1	0
DEC_CR1	0	1	0	0	0	DCLKSEL		
DEC_CR2	1	0	シフト		1	DecimationRate		
DEC_DH	デシメータの上位バイト出力							
DEC_DL	デシメータの下位バイト出力							

デシメータは、Sinc2 フィルタの実装に使用される専用ハードウェアです。3 つの制御レジスタと 2 つのデータ出力レジスタから構成されています。DCol は接続するコラムコンパレータを選択します。DCLKSEL は、デシメータタイミングを制御するデジタルブロックを選択します。どちらのパラメータも、デバイス エディタで設定します。DEC_CR2 のシフトは、ソフトウェアで達成しなければならないアラインされたデータを最小化するために、デシメーションレートに基づいて設定します。また DEC_CR2 でも指定できます。

バージョン ヒストリー

バージョン	著者	説明
1.0	DHA	初期バージョン

Note PSoC Designer 5.1 ではすべてのユーザ モジュール データシートにおいてバージョン ヒストリーを導入し、ユーザー モジュールの過去のバージョンと現在のバージョンとの違いに関して高度な解説を掲載しています。

Copyright © 2008-2011 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.