6-Bit 電圧出力 DAC データシート DAC6 V 4.3

Copyright © 2001-2010 Cypress Semiconductor Corporation. All Rights Reserved.

リソース	PSoC® Block			API メモリ (バイト)		ピン (外部 I/O)
	デジタル	アナログ CT	アナログ SC	フラッシュ	RAM	
CY8C29/27/24/23/22xxx, CY8CLED04/08/16, CY8CLED0xD, CY8CLED0xG, CY8CTST120, CY8CTMG120, CY8CTMA120, CY8C28x45, CY8CPLC20, CY8CLED16P01, CY8C28x43, CY8C28x52						
	0	0	1	61	0	1
CY8C26/25xxx	0	0	1	61	0	1

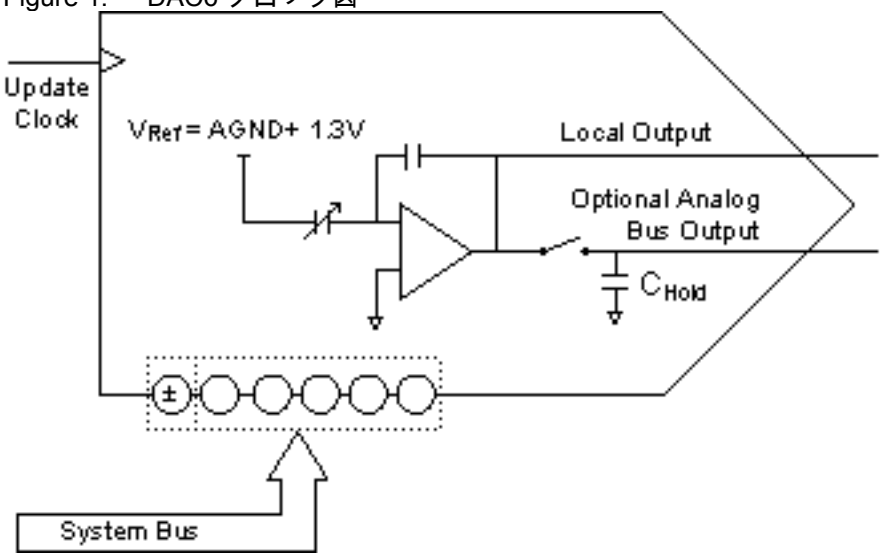
このユーザモジュールを使用したサンプルプロジェクトは、以下を参照してください。
www.cypress.com/psocexampleprojects.

特徴及び概要

- 6-bit 解像度
- 電圧出力
- 2 の補数、オフセットバイナリ、及び記号 / サイズの入力データフォーマット
- アナログバスと外部出力用 sample and hold 回路
- 250 ksps アップデートレート

DAC6 ユーザーのモジュールは、デジタルコードを出力電圧に変換します。DAC6 は毎秒 250k サンプルまでの更新レートでのデジタルコードを出力電圧に変換します。アプリケーションプログラミングインターフェイス (API) は、最大の自由度のためにオフセットーバイナリ、コード、及びサイズ、2 の補数データフォーマットをサポートしています。エラーを最小限にするためのオフセット補正機能が使用されています。

Figure 1. DAC6 ブロック図

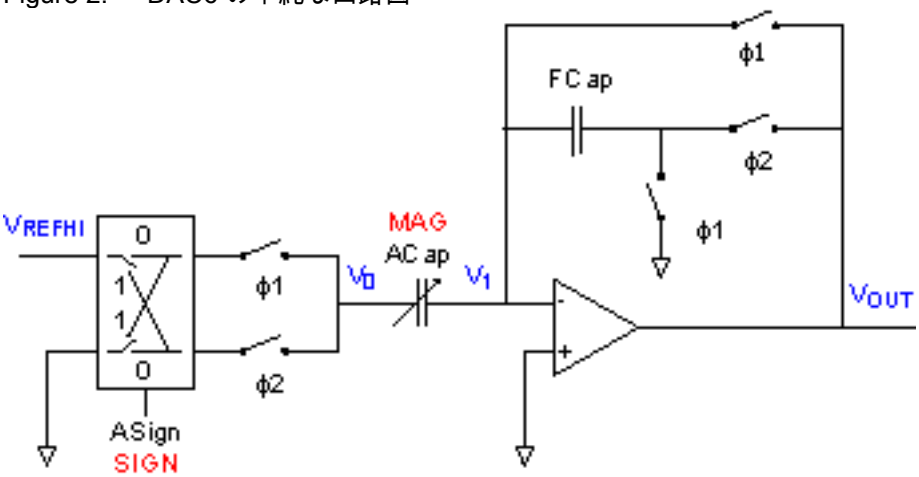


機能説明

DAC6 ユーザーのモジュールは、デジタルコードをアナログ出力電圧に変換します。デジタルコードは、-31 ~ +31 の範囲を持つ 2 の補完やコード、及びサイズのフォームに数値として表示されます。代替的な方法で、入力のコードを 0 ~ 62 までの範囲の数値でオフセットーバイナリ形式に表示することができます。これは、出力電圧のオンーストップを変更する値が、より一般的な 64 番目の変更の値というよりも、全体の尺度、出力範囲の 63 番目の変更の値を表すことを意味します。コード、及びサイズのフォームで構成される入力コード '-0' は、ユーザーのモジュール API によって '+0' に変換されます。システムレベルのバラメータ、REFMux のために選択した値に応じて、出力電圧範囲が異なる場合があります。

内部的に見ると、これらの作業は、コード、及びサイズのフォーマットに基づいています。5 つのサイズのビットは、以下の単純な図式で提示されたバイナリ加重コンデンサアレイである ACap 値を設定します。ACap は、0 ~ 31 単位の値を想定しています。ASign ビットに変換可能な基準の電圧は、32 単位の公称容量を持つフィードバッグコンデンサ、FCap に対する ACap の比率によって、出力装置に表示されます。

Figure 2. DAC6 の単純な回路図



コードビットは、クロック信号の Φ_1 及び Φ_2 によって制御されるスイッチと一緒にクロスバースイッチを介して極性を制御します。これは、周期が同じで、位相が逆です。Φ_1 と Φ_2 が " 下にラップ " されます。つまり、一つアクティブではない時に、個々のパルスの間に短い周期が存在します。これは Φ_1 と Φ_2 が形を作成する前にブロックされる関連のスイッチを開いて閉じます。個々のアナログコラムのクロックジェネレータは、ソースクロックを 24 MHz オシレータに同期して、これを生成するために 4 分周までします。

入力コードが正の値 (SIGN ビットが 0) のとき、クロスバーは V_{REFHI} を ACap に接続し、Φ_1 が有効にされている間、ACap を基準電圧マイナス AGND に充電します。Φ_1 が有効に移動し、Φ_2 を有効にした後、ACap の入力側が V_{REFHI} で AGND に変換され、検出状態を AGND に関連して、効果的に変換されます。ACap と FCap の間では、充電値が共有される場合、opamp が反対側の充電値を提供します。

したがって、正の値の入力コードの場合、2 つの電圧閾値の現象が発生します。ACap スイッチのソース端子が AGND に転換されるとき、最初の変換作業が発生します。ACap が変換入力装置に接続されるため、opamp が 2 番目の変換処理を強制します。負の値の入力コードの分析は同様です。大事な違いは、V_{REFHI} が Φ_1 というよりも Φ_2 で直接適用されるということです。これらの作業は、ACap の効果的な充電値への変換作業が発生せず、唯一の変換作業が opamp により提供されるように実行されます。

ハードウェアがそれぞれの更新プログラムの周期でオフセット補正作業を実行します。Φ_1 と Φ_2 によって制御されるスイッチは、Φ_1 間、反転入力の利点フォロワー opamp を構成します。このような構成では、オフセット電圧はサミングノードに現れ、ACap と FCap の両方を充電します。Φ_2 に再構成されたように、回路がこのようなコンデンサのオフセットの充電値を変換し、オフセット電圧を効果的にオフにします。

すべてのアップデートサイクルでは、V_{out} がオフセット電 (Φ_1) から目的の電圧 (Φ_2) まで変化する。直接の結果としてオフセットの補償となる。精度を向上させるために、これらの価格を軽減させることができる 1 つの方法は、出力バスに関連する sample-and-hold 回路を使用することです。Vout は、第 Φ_2 の半分の間、load 及び hold コンデンサ (DAC6 ブロック図で CHold) を充電します。CHold は、該当のサイクルが終了する時点で、opamp 出力装置から分離されます。個々のアナログ出力バスは、非常に高い入力インピーダンスを持つアナログ出力デバイスのバッファによって使用されます。

Equation 1

$$V_{Out} = (V_{REFHI} - AGND)\frac{ACap}{FCap} + AGND = 1.3V\left(\frac{MAG}{32}\right) + 2.6V, 0 \le MAG \le 31$$

デバイスエディタによって REFMux パラメータが 2*BandGap ± BandGap で構成されている場合、次のような等式が適用されます。

例題

DAC の入力コードのために指定された 16 に対する値は、式 2 に提示されたように、結果的に発生する出力電圧が 3.25V になると予想することができます。

Equation 2

$$V_{Out} = 1.3Volts\left(\frac{16}{32}\right) + 2.6Volts = 3.25Volts$$

計算された値は、実際的な値であり、これは、システムのノイズやチップのオフセット値に応じて大幅に異なることがあります。

DC 及び AC 電気の特性

以下の値は、初期の特性データを元に予測される性能を示しています。次の表に明記されていない場合は、T_A = 25°C 及び Vdd = 5V 等式が成立されます。また、f_{clock} = 125 kHz, 外部 AGND 2.50V, 外部 V_{Ref} 1.23V, REFPWR = HIGH, SCPOWER = ON の場合には、PSoC の電源 HIGH をブロックします。

Table 1. 5.0V DAC6 DC 及び AC 電気特性、CY8C29/27/24/22xxx PSoC デバイスファミリ

パラメータ	一般	限界	単位	条件及び注意事項
解像度	--	6	ビット	
リニア				
DNL	0.09	--	LSB	
INL	0.07	--	LSB	
単調な	可能	--		
利得エラー				
基準利得エラーを含む	3.4	--	%FSR	
基準利得エラーを除く [3]	0.45	--	%FSR	
V_{OS}, オフセット電圧	±7.5	--	mV	
出力ノイズ	4.6	--	mV rms	0 ~ 300 kHz
f_{clock}, アナログコラムクロック [1]				
低電力	0.128 ~ 0.5 kHz	--	MHz	
中間電力	0.128 ~ 2.0 kHz	--	MHz	
高電力	0.128 ~ 3.2 kHz	--	MHz	
動作電流 [2]				
低電力	155	--	µA	
中間電力	585	--	µA	
高電力	2225	--	µA	

以下の値は、初期の特性データを元に予測される性能を示しています。次の表に特に明記されていない場合は、T_A = 25°C と Vdd = 3.3V 等式が成立されます。また、f_{clock} = 125 kHz, 外部 AGND 1.50V, 外部 V_{Ref} 0.8V, REFPWR = HIGH, SCPOWER = ON の場合には、PSoC が電源 HIGH をブロックします。

Table 2. 3.3V DAC6 DC 及び AC 電気特性、CY8C29/27/24/22xxx PSoC デバイスファミリ

パラメータ	一般	限界	単位	条件及び注意事項
解像度	--	6	ビット	
リニア				
DNL	0.09	--	LSB	
INL	0.07	--	LSB	
単調な	可能	--		
利得エラー				
基準利得エラーを含む	2.9	--	%FSR	
基準利得エラーを除く [3]	0.3	--	%FSR	
V_{OS}, オフセット電圧	±7.5	--	mV	
出力ノイズ	2.1	--	mV rms	0 ~ 300 kHz
f_{clock}, アナログコラムクロック [1]				
低電力	0.128 ~ 0.5 kHz	--	MHz	
中間電力	0.128 ~ 2.0 kHz	--	MHz	
高電力	0.128 ~ 3.2 kHz	--	MHz	
動作電流 [2]				
低電力	150	--	µA	
中間電力	560	--	µA	
高電力	2150	--	µA	

以下の値は、初期の特性データを元に予測される性能を示しています。次の表に特に明記されていない場合は、T_A = 25°C と Vdd = 2.7V 等式が成立されます。また、f_{clock} = 125 kHz, 外部 AGND 1.50V, 外部 V_{Ref} 0.8V, REFPWR = HIGH, SCPOWER = ON の場合には、PSoC の電源 HIGH をブロックします。

Table 3. 2.7V DAC6 DC 及び AC 電気特性、CY8C29/27/24/22xxx PSoC デバイスファミリ

パラメータ	一般	限界	単位	条件及び注意事項
解像度	--	6	ビット	
リニア				
DNL	0.09	--	LSB	
INL	0.07	--	LSB	
単調な	可能	--		
利得エラー				
基準利得エラーを含む	2.9	--	%FSR	
基準利得エラーを除く [3]	0.3	--	%FSR	
V_{OS}, オフセット電圧	±7.5	--	mV	
出力ノイズ	2.1	--	mV rms	0 ~ 300 kHz
f_{clock}, アナログコラムクロック [1]				
低電力	0.128 ~ 0.5 kHz	--	MHz	
中間電力	0.128 ~ 2.0 kHz	--	MHz	
高電力	0.128 ~ 3.2 kHz	--	MHz	
動作電流 [2]				
低電力	150	--	µA	
中間電力	560	--	µA	
高電力	2150	--	µA	

電気特性注意事項

1. 広帯域ノイズの増加のため指定された 3dB 範囲の上部の先端。たれた < 1 LSB の下部の先端。DAC によって選択されたアナログコラムクロックは、更新プログラムのサイクルを管轄する位相のクロックレートの 4 倍以上高速です。タイミングの説明については下記を参照してください。
2. 基準ブロックの電源が含まれておらず、すべてのアナログ信号が遮断されています (PSoC ファミリのデータシートを参照)。
3. ピンのテストのマルチプレクサ及びバックアウトを介して送信される $V_{RefHigh}$ と V_{RefLow} と外部基準電圧を比較して測定された基準利得エラー。

4. 次の表に特に明記されていない場合は、すべての限界値が T_A = 25°C と Vdd = 5V 等式を保障します。また、f_{clock} = 125 kHz, 外部 AGND 2.50V, 外部 V_{Ref} 1.23V, REFPWR = HIGH, SCPOWER = ON の場合には、PSoC が電源 HIGH を遮断します。

Table 4. 5.0V DAC6 DC 及び AC 電気の特性 CY8C26/25xxx PSoC デバイスファミリ

パラメータ	一般 [1]	限界 [2]	単位	条件及び注意事項
解像度	--	6	ビット	
リニア				
DNL	.02	.05	LSB	
INL	.03	.08	LSB	
単調な	--	½	ビット	
利得のエラー	1.0	2.5	%FSR	
V_{OS}, オフセット電圧 [3]	8	43	mV	
出力ノイズ				
帯域の限界	.3	1	mV rms	0 ~ 10 kHz
広帯域	7	10	mV rms	0 ~ 300 kHz
f_{clock}, アナログコラムクロック [4]	--	0.128 ~ 1.33 kHz	MHz	
動作電流 [5]				
低電力	125	--	?A	
中間電力	280	--	?A	
高電力	780	1000	?A	

次の表に特に明記されていない場合は、すべての限界値が T_A = 25°C と Vdd = 3.3V 等式を保障します。また、f_{clock} = 125 kHz, 外部 AGND 1.50V, 外部 V_{Ref} 0.80V, REFPWR = HIGH, SCPOWER = ON の場合には、PSoC が電源 HIGH を遮断します。

Table 5. 3.3V DAC6 DC 及び AC 電気特性、CY8C26/25xxx PSoC デバイスファミリ

パラメータ	一般 [1]	限界 [2]	単位	条件及び注意事項 [5]
解像度	--	6	ビット	
リニア				
DNL	.02	.04	LSB	
INL	.04	.09	LSB	
単調な	--	½	ビット	
利得のエラー	1.0	2.5	%FSR	

パラメータ	一般 [1]	限界 [2]	単位	条件及び注意事項 [5]
V_{OS}, オフセット電圧 [3]	7	31	mV	
出力ノイズ				
帯域限界	.3	1	mV rms	0 ~ 10 kHz
広帯域	7	10	mV rms	0 ~ 300 kHz
f_{clock}, アナログコラムクロック [4]	--	0.128 ~ 1.33 kHz	MHz	
動作電流 [5]				
低電力	100	--	?A	
中間電力	250	--	?A	
高電力	640	900	?A	

電気特性注意事項

1. 一般的な値は、統計平均値 + 1 を意味します。
2. 試験または統計分析により、限界値が保障されます。
3. 外部 AGND に、2 の補数ゼロスケールオフセット値には、アナログ出力バッファのオフセットエラーが含まれていません。
4. 広帯域ノイズの増加のために、指定された 3dB 範囲の上部の先端。たれた < 1 LSB の下部の先端。DAC によって選択されたアナログコラムクロックは、更新プログラムを管轄する位相のクロックの 4 倍以上の高速です。タイミングの説明については下記を参照してください。
5. PSoC 遮断電流は基準電流を除外するよう求めています。

交換

DAC6 ブロックが、そのデバイス上で変換されたコンデンサ PSoC ブロックで自由にマップされます。しかし、DAC6 出力デバイスがアナログ出力バス上で活性化されている場合、他のユーザーのモジュールが同じバスを駆動させないように注意を払う必要があります。交換する場所を選択するときに、追加的に考慮しなければならない事項は、DAC6 ブロックが使用されるクロックが同じ PSoC ブロックのコラムにマップされた他のユーザーのモジュールのブロックとの交換性が必要であることです。

パラメータ、及びリソース

DAC6 例を生成するには、デバイスのエディタで、ユーザーのモジュールを選択し、該当する場合には、名前を変更し、装置内に変換されたすべてのコンデンサ PSoC ブロックにマップしてください。交換時に考慮すべき注意点としては、信号がオフのチップに駆動される場合、そしてコラムクロックリソースの他のユーザーのモジュールとの相互依存的な場合には、アナログコラム出力バスの可能性を含むことができます。交換された場合には、ユーザーモジュール、これらのブロックを “DAC" という象徴的な名前を付与してパラメータを表示します。

データフォーマット

DAC6 ユーザーモジュールの API は、3 種類の異なるデータ形式、つまり、オフセットバイナリ、2 の補数と符号とサイズのフォーマットを扱っています。API セクション (下) の WriteBlind エントリポイントには、このような習慣と、それぞれに関連する値の範囲が説明されています。

アナログバス

DAC ブロックは、隣接している、アナログ PSoC ブロック側の出力デバイスを発信しています。アナログバスのオプションのいずれかを選択する場合は、アナログ出力バッファのいずれかを使用して、DAC 出力のデバイスが外部と接続されています。特定のコラムでバスを選択した場合、アレイの上部にある PSoC ブロックとの追加的な接続機能が提供されます。スイッチドキャパシター PSoC ブロックには、Φ_2 の半分の間 DAC 出力デバイスを見本として抽出する一つの sample and hold 回路が組み込まれています。これは、自動的にゼロ作業の期間中発生する電圧のスイングから外部出力を分離します。

クロック位相

このパラメータは、次のクロックとタイミングのセクションに議論されているコラムクリック分周器によって生成された位相のクロック、 Φ_1 と Φ_2 の役割を判別します。 *一般*が選択された場合、Φ_1 の間の自動ゼロサイクルが発生し、DAC 出力デバイスが Φ_2 で有効に機能します。クロック位相を *Swapped* に設定する場合は、これらの役割が逆になります。これは、DAC を Φ_1 からの入力デバイスを見本抽出する別の周辺機器に接続するときに便利です。

Note アナログバスの Sample and hold 機能はクロックフェーズ設定により無効になる。アナログバスのパラメータを有効に設定する場合には、バスの出力がローカル PSoC ブロックの出力を表示し、Φ_1 の間 AGND(+ オフセット電圧) と Φ_2 の間、必要な出力デバイスを変更します。

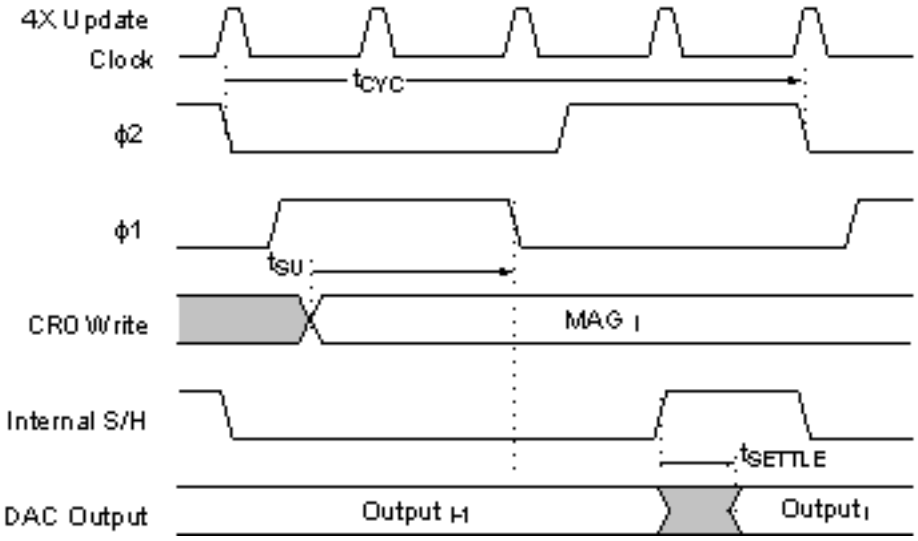
アナログコラムのクロック

DAC は、適切な WriteBlind と WriteStall API 機能を呼び出すことにより、新しい値を " 作成 " するよう命令を受け取ったかどうかに関係なく、出力デバイスを連続的に更新します。アナログコラムクロックマルチプレクサは、これらの更新作業を制御する位相クロック、Φ_1 と Φ_2 を生成するために使用するソースクロックを選択します。位相クロック発生器は、Φ_1 と Φ_2 と生成するために、コラムのクロックを 4 つまで分割するのでコラムのクロック周波数は、実際のアナログ出力更新レートよりも 4 倍以上高速です。2 つのレベルの多重化を介して、すべてのデジタルブロックと、システムクロック分周期が含まれているコラムのクロックを選択することができます。上記の電気特性のセクションには、コラムのクロック周波数の下部と上部の限界値が指定されています。

正の出力電圧 (Vout > AGND) と一般的な位相の場合、上記の単純なグラフに示されたように Φ_1 の間に基準電圧が A Cap に保存されます。A Cap を完全に充電するには、値の変化が設定時間 t_{SU} で Φ_1 下がりエッジまでの要件を満たしている必要があります。次の図に示されているこれらの設定時間は基準の電源レベルに応じて異なります。設定時間が特徴化されていないにもかかわらず、ハードウェアのストールの構造を使用して、これらの要求を満たすように保障することができます。不適切な設定時間の入力により、A Cap が完全に充電されていない場合、この設定時間が校正されている f1 全体のサイクルの次の位相クロックサイクルまで出力デバイスが校正されません。f2 の強いエッジに関連する同様の設定時間は、負の出力電圧 (Vout < AGND) の動作を管轄しています。

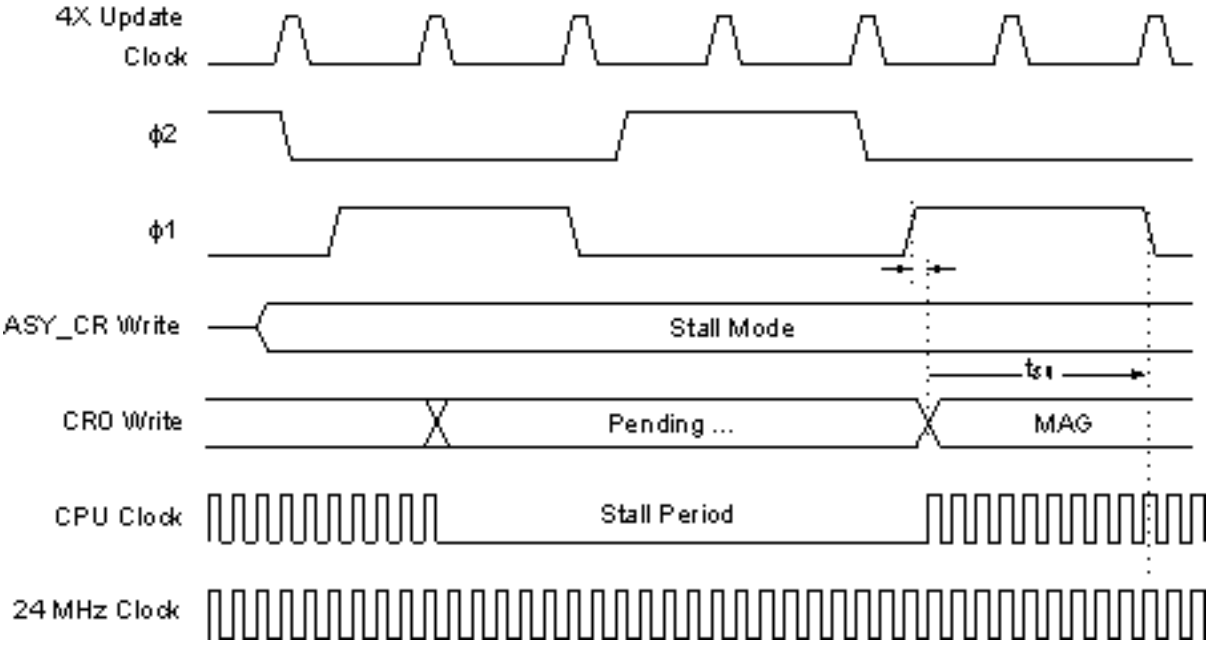
[+] Feedback

Figure 3. V_{OUT} > AGND の更新タイミング



より大きい等級のアプリケーションのため、瞬間的な (1 つの更新プログラムのサイクル) の偏差が許容されています。他のアプリケーションは、より厳しい要求条件が付与されることができます。ハードウェアの同期機能は、A Cap 値を変更するレジスタをライトするタイミングを制御するために使用することができます。この機能は、入力内容によって WriteStall API 内で直接サポートされます。呼び出された場合、該当のハードウェアが PSoC ブロックのレジスタの作成の命令を認識し、CPU クロックを停止させ、Φ_1 の昇降のエッジまでの作成命令が完了されるのを留保します。ASY_CR レジスタは、次の操作を制御します。

Figure 4. ハードウェアの同期機能、CPU クロックのストール



CPU ストール時に、すべてのアナログ及びデジタル PSoC が機能を正常にブロックします。この周期の間 DAC の CR0 レジスタに書く MOV 命令が、単に中断して、割り込みされるか、係留されている状態のままになります。同じストール時に CPU 周期数が失われるため、このとき、次の関係を使用して、周期を計算することができます。

Equation 3

$$CPU\,Cycles \leq \frac{F_{CPU}}{F_{\phi_1}} = \frac{4 \times F_{CPU}}{F_{ColClock}}$$

はっきり言って、ストール側の CPU サイクルの損失を最小限に抑えるには、コラムのクロックを実用的に最も高い周波数で駆動する必要があります。外部出力の周期が、以前の出力周期を単純に繰り返すため、出力電圧が実際に変更するときに必要な値よりもはるか高くなることがあります。より迅速なコラムクロックは、出力機能の呼び出す時間から出力を変更する時間までの遅れを最小限に抑えます。

しかし、コラムのクロック周波数の実際的な限界が存在します。DAC は、AGND の出力電圧の各位相のクロック周期で変化するため、コラムクロックが出力デバイスが安定するように許可される周波数に制限されています。アナログ出力バスの見本抽出及び固定機能を使用する場合、opamp 出力デバイスが 2 番目の Φ2 の半分の周期で標本抽出のウインドウが表示されている、バスを駆動します。Opamp が出力電圧を変化させ安定化されるに十分にコラムクロックが速い場合は、これは、出力信号のノイズで観察することができます。極端な例では、出力デバイスは、見本抽出ウィンドウが閉まる前に最終的に出力電圧の一部としてのみ変化します。これは、出力の全範囲の終わりで最も明らかにされる深刻な非直線入力の圧縮値を観察することができます。これらの現象が発生することを防止するアナログコラムクロックのハイレベルの限界は、上記の電気特性表に示されています。

アプリケーションプログラミングインターフェイス

アプリケーションプログラミングインターフェイス (API) ルーチンは、ユーザーのモジュール一部として提供されるため、設計者はより高いレベルでのモジュールを処理することができます。このセクションでは、“include" ファイルが提供した関連の定数と一緒に、個々の機能へのインターフェイスの機能が指定されています。

Note ここでは、API 機能を呼び出して A 及び X レジスタの値を、すべてのユーザーモジュールの API に変更することができます。呼び出された後、それらの値が必要な場合は、呼び出しの前に A と X の値を保持する責任は、呼び出し機能を持つデバイスにあります。この “registers are volatile(レジスタは、揮発性である)" という政策は、効率の理由のために選択され、1.0 の PSoC Designer のために効力が発生するようになりました。C コンパイラは、これらの要求条件を自動的に観察します。アセンブリ言語のプログラマは、該当のコードが、この方針を遵守しているかを確認する必要があります。数個のユーザーモジュールの API 機能で、A と X が変更されていない状態で残されている可能性があるにもかかわらず、今後もそのまま残されていない可能性があります。

DAC6 ユーザーモジュールを初期化して、更新プログラムの値を作成し、ユーザーのモジュールを無効にするために、入力内容が提供されています。

DAC6_Start

説明：

このユーザーのモジュールに必要なすべての初期化処理を実行して、スイッチドキャパシター PSoC ブロックに電力レベルを設定します。

C プロトタイプ：

```
void  DAC6_Start(BYTE bPowerSetting)
```

アセンブラ：

```
mov   A, bPowerSetting
lcall  DAC6_Start
```

パラメータ：

bPowerSetting: 電源レベルを指定する 1 バイト。リセット及び構成されてから、DAC ブロックに割り当てられている PSoC ブロックの電源が遮断されています。C 及びアセンブリとして提供され、象徴的な名前と関連する値は、次の表に記載されています。

象徴的な名前	値
DAC6_OFF	0
DAC6_LOWPOWER	1
DAC6_MEDPOWER	2
DAC6_FULLPOWER	3

戻り値：

なし

副作用：

DAC 出力が駆動されます。デフォルトで、初期値は AGND です。電源が供給されるとき、一部の他の出力値が必要な場合は、" スタート " を起動する前に、作動ルーチンのいずれかを呼び出します。A 及び X レジスタが、この機能によって変更されることがあります。

DAC6_SetPower

説明：

DAC の スイッチドキャパシター PSoC ブロックの電源レベルを設定します。ブロックを遮断するときとつけるときに使用することができます。

C プロトタイプ：

```
void  DAC6_SetPower(BYTE bPowerSetting)
```

アセンブラ：

```
mov   A, bPowerSetting
lcall  DAC6_SetPower
```

パラメータ：

bPowerSetting: スタート入力内容のために使用される PowerSetting パラメータと同じです。

戻り値：

なし

副作用：

DAC 出力が駆動されます。デフォルトで、初期値は AGND です。電源が供給されるとき、一部の他の出力値が必要な場合は、" スタート " を起動する前に、作動ルーチンのいずれかを呼び出します。A 及び X レジスタが、この機能によって変更されることがあります。

DAC6_WriteBlind

説明：

出力電圧を指示された値を直ちに更新します。

C プロトタイプ：

```
void  DAC6_WriteBlind(CHAR cOutputValue)
```

アセンブラ：

```
mov   A, cOutputValue
lcall  DAC6_WriteBlind
```

パラメータ：

cOutputValue: 出力電圧を指定する 1 バイト。次の表に示すように、許可された値は、選択された DataFormat の値と同じ範囲に置かれています。

データフォーマット	最小値	最大値
OffsetBinary	0	62
TwosComplement	-31	31
SignAndMagnitude	-31	31

2 の補数、及び OffsetBinary は、マイクロコントローラの本来の 2 の補数形式を使用します。オフセットーバイナリ値はゼロまで表示される最小の出力電圧を持って 62 まで表示される最大の出力電圧を持つ正の数である。mmmmm が大きさであり、s が符号である場合、SignAndMagnitude になったのはバイトは、バイナリフォーム ” 00smmmmm” を持つ必要があります。正の数の場合は、0 を使用して暗号化して、負の場合、 1 を使用して符号化します。

戻り値：

なし

副作用：

このユーザーのモジュールのタイミングのセクションで説明している理由のため、出力デバイスに障害が発生する可能性があります。A 及び X レジスタが、この機能によって変更されることがあります。

注意事項：OFFSET_BINARY を選択する場合は、範囲外の入力値が API 内の 2 つのバックアップデータに自動変更されます。これは、可能な限り許可されたオフセットバイナリ値を超える範囲の値が小さい養成出力 (Agnd 付近) の値に変換されることを意味します。

DAC6_WriteStall

説明：

Φ_1 が起動されるまでのマイクロプロセッサーをストールした後、出力電圧を指示された値で更新します API は、割り込みが無効になったり、最大の障害遅延が ACLKi 以下という事実を前提としていることが注意されます。

C プロトタイプ：

```
void  DAC6_WriteStall (CHAR cOutputValue)
```

アセンブラ：

```
mov   A, cOutputValue
lcall  DAC6_WriteStall
```

パラメータ：

cOutputValue: WriteBlind 入力事項に記載された cOutputValue パラメータと同じフォーマットと同じ範囲の値

戻り値：

なし

副作用：

ACLKi が無効 ('i' は、アナログ PSoC ブロックがマップされる列にある) になる場合、Φ_2 が無効になるまで、つまり、3/4 の更新プログラムのサイクルの間 (+ 2 つの CPU クロック) マイクロプロセッサーの CPU クロックが不能の状態がされています。ストールの間隔の間の割り込みが確認されないことに注意してください。A 及び X レジスタが、この機能によって変更されることがあります。

DAC6_Stop

説明：

ユーザーモジュールの電源を遮断します。

C プロトタイプ：

```
void  DAC6_Stop(void)
```

アセンブリ：

```
lcall  DAC6_Stop
```

パラメータ：

なし

戻り値：

なし

副作用：

出力が駆動しません。A 及び X レジスタが、この機能によって変更されることがあります。

サンプルファームウェアのソースコード

次のサンプルコードは、一つのサイクルを発生させ、のこぎり波を下に下げます。

```
//-------------------------------------------------------------------------
// This C sample code for the DAC6 user module creates a periodic signal
// that ramps down.
//-------------------------------------------------------------------------

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


#define DAC_MAX (62)                    // Define max DAC value as 62

unsigned char bDACValue = 0;            // Variable for the DAC value
unsigned char i;                        // Variable for an index

void main(void)
{
    DAC6_Start(DAC6_HIGHPOWER);         // Start DAC6 in HIGH power mode

    while(1)                            // Repeat forever
    {
        if(bDACValue == 0)
        {
           bDACValue = DAC_MAX;        // Reset DAC value to the max if it reached zero
        }

        DAC6_WriteStall(bDACValue--);   // Write value to DAC and decrement

        for(i = 0xFF; i != 0; i--);     // Delay loop
    }
}
```

次のアセンブリのサンプルコードは、C のサンプルコードと同じ機能を持ちます。

```
;-------------------------------------------------------------------------
; This sample code for the DAC6 user module generates a periodic signal that
; ramps down
;-------------------------------------------------------------------------

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

export _main

DAC_MAX: equ 62             ; This is the maximum DAC value

area bss (RAM, REL, CON)
    bDACValue: blk 1        ; Variable to hold the DAC value


area text (ROM, REL, CON)
```

```
_main:
    mov A, DAC6_HIGHPOWER    ; Start DAC with HIGH power setting
    call DAC6_Start

Init:
    mov [bDACValue], DAC_MAX ; Initialize DAC value to hold the maximum
    mov X, 0xFF              ; Initialize X register to hold 0xFF

RampDown:
    mov A, [bDACValue]       ; Move DAC value into A register
    call DAC6_WriteStall     ; Write the value in A to the DAC

Delay:
    dec X                    ; Decrement X register
    jnz Delay                ; Keep delaying if it hasn't reached zero yet

    dec [bDACValue]          ; Decrement DAC value variable
    jnz RampDown             ; If it is not zero, keep ramping down
    jmp Init                 ; If it is zero, restart the ramp down
```

設定レジスタ

API は、DAC6 ユーザーモジュールの完全なインターフェースを提供します。設定レジスタに直接記述する場合、出力の代わりに更新することができます。どちらも出力デバイスの障害を防止するために理解する必要があるタイミングの考慮事項が存在します。以下のレジスタが DAC6 スイッチドキャパシタ DAC ブロックのために使用されます。

Table 6. ブロック DAC ASAxxCR0 または ASBxxCR0: レジスタ CR0

ビット	7	6	5	4	3	2	1	0
値	1	0	コードとサイズ					

コードとサイズは、リセットして再構成した後、中間目盛 (AGND) に設定されます。これは 、API で、"Write" を呼び出すことによって変更されます。

Table 7. ブロック DAC ASACR1xx または ASBxxCR1: レジスタ CR1

ビット	7	6	5	4	3	2	1	0
値	0	1	0	0	0	0	0	0

Table 8. ブロック DAC ASACR2xx または ASBxxCR2: レジスタ CR2

ビット	7	6	5	4	3	2	1	0
値	アナログバス	0	1	0	0	0	0	0

アナログバスは、DAC PSoC ブロックがバスを駆動させるかどうかを判別します。ビットフィールドの値は、デバイスエディタのユーザーモジュールの交換モードでは、同じ名前のパラメータを選択することにより決定されます。

Table 9. ブロック DAC ASACR3xx または ASBxxCR3: レジスタ CR3

ビット	7	6	5	4	3	2	1	0
値	0	0	1	1	0	0	電源	

電源は、デバイスのリセット、及び構成の後にオフに設定されます。これは、API の Start, SetPower または Stop 、入力内容が変更されます。

Table 10. グローバルレジスタ ASY_CR

ビット	7	6	5	4	3	2	1	0
値	0	0	0	0	0	0	0	1

API は、必要に応じて、これらのレジスタを作成して、出力の更新タイミングの条件を保障するために、CPU をストールします。

Copyright © 2001-2010 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.