アナログ スイッチド キャパシタ PSoC ブロック データシート SCBLOCK V 2.4

Copyright © 2002-2011 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	フラッシュ	RAM	
CY8C29/27/26/25/24/23/22xxx, CY8CLED04/08/16, CY8CLED0xD, CY8CLED0xG, CY8CTST120, CY8CTMG120, CY8CTMA120, CY8C28x45, CY8CPLC20, CY8CLED16P01, CY8C28x43, CY8C28x52						
	0	0	1	20	0	

このユーザ モジュールを使用する、ひとつまたは複数の完全設定されたサンプルプロジェクトについては、以下を参照してください。www.cypress.com/psocexampleprojects

特性および概要

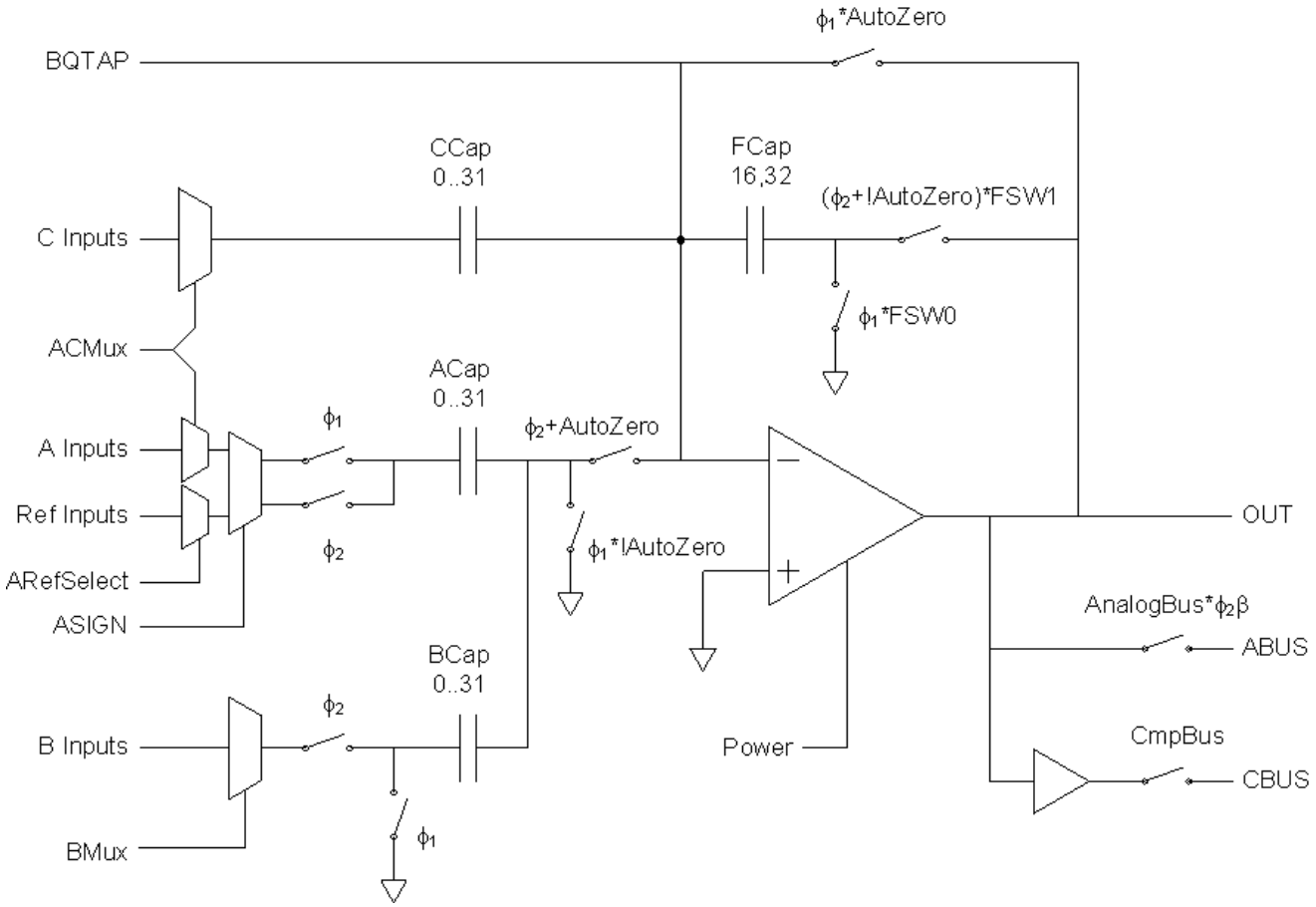
- 完全パラメータ化によりカスタム開発に対応
- プロトタイプ用カスタムブロック
- 選択可能な出力（消費電力）設定

SCBLOCK ユーザ モジュールは、完全にパラメータ化されたアナログ スイッチド キャパシタ（SC）PSoC ブロックで、スイッチド キャパシタ機能のカスタム メードが可能です。SCBLOCK 電源管理のために、アプリケーション プログラミング インタフェース（API）が含まれています。

CY8C26/25xxx デバイスファミリ向けの場合、SCBLOCK は「A」または「B」タイプの SC PSoC ブロックで構成されます。PSoC デバイス向けの場合、SCBLOCK は「C」または「D」タイプの SC PSoC ブロックで構成されます。

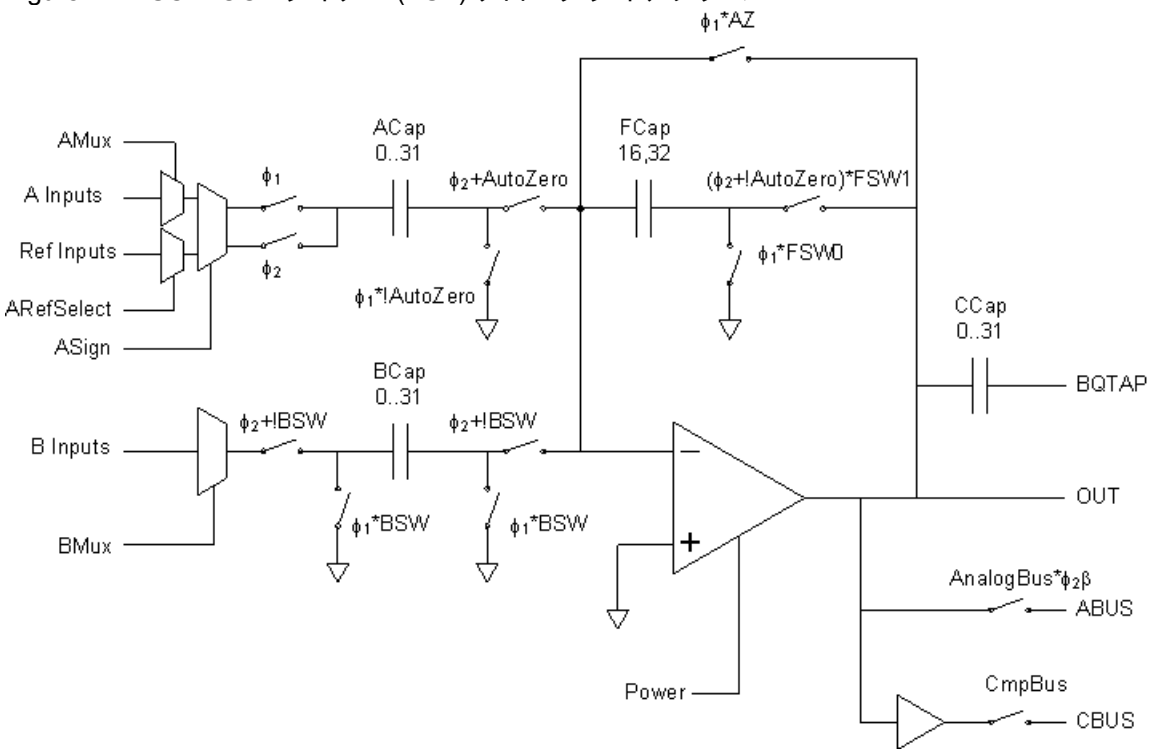
次にこれらのタイプを示します。

Figure 1. SCBLOCK タイプ A (ASA) ブロック ダイアグラム



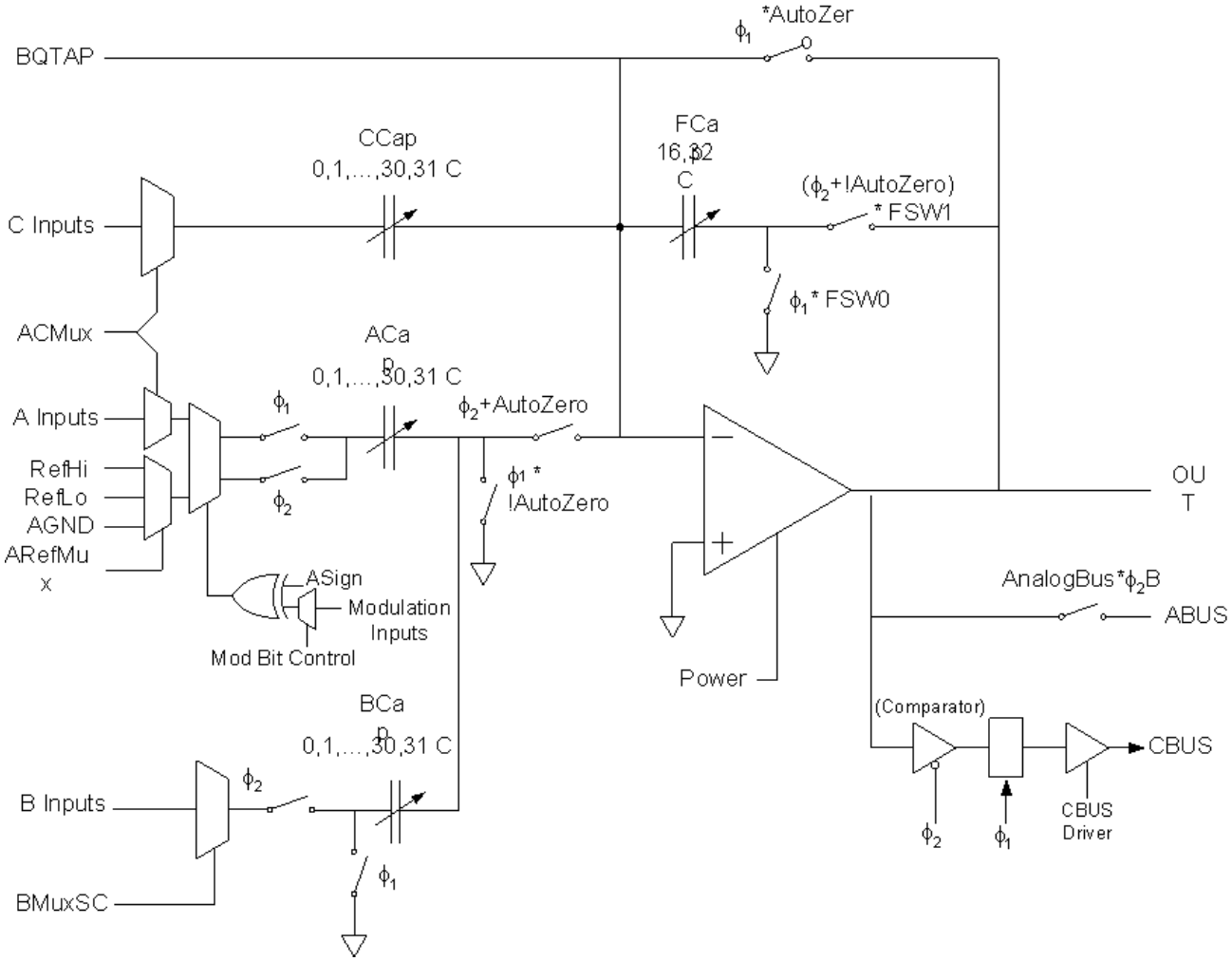
[+] Feedback

Figure 2. SCBLOCK タイプ B (ASB) ブロック ダイアグラム



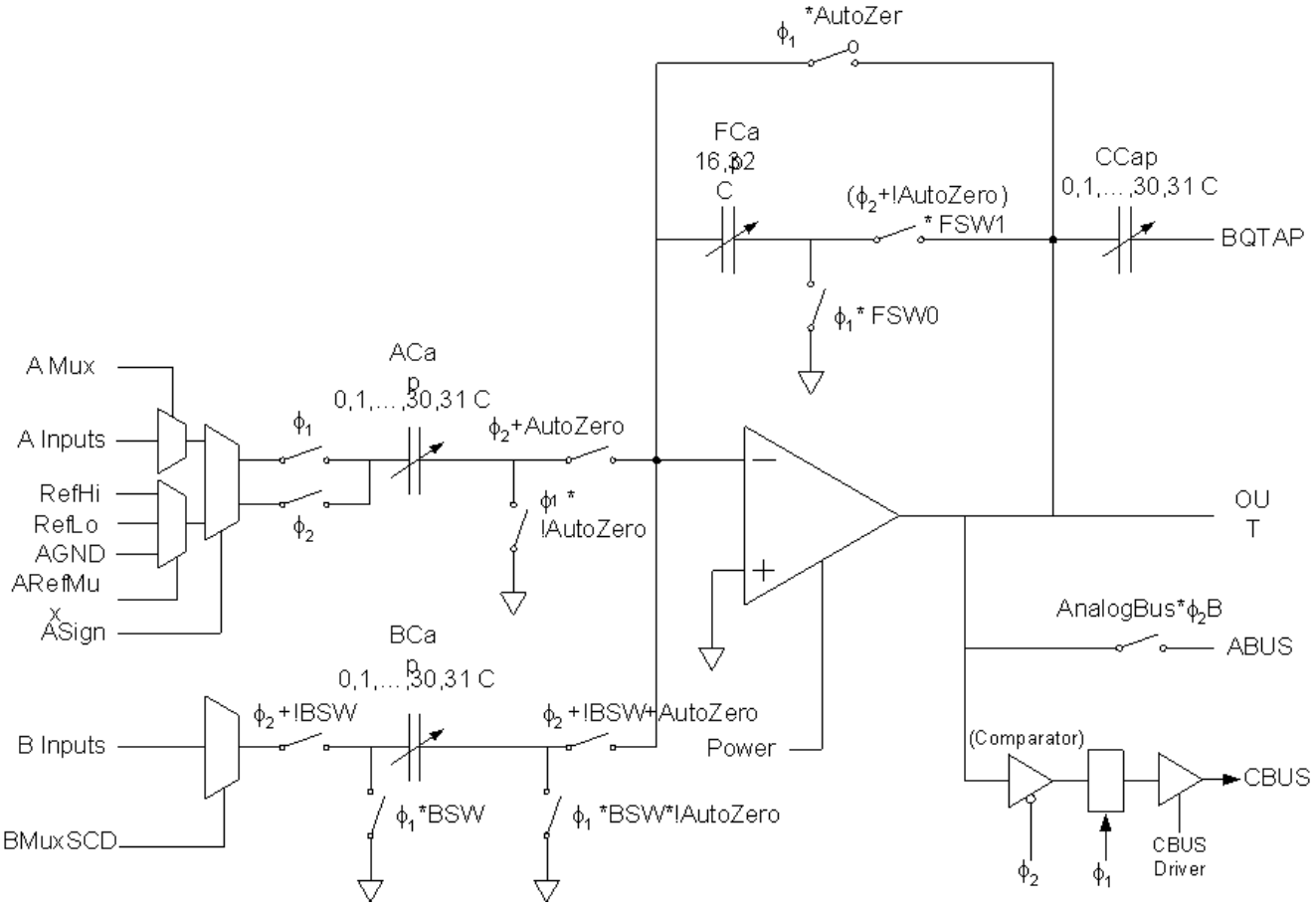
[+] Feedback

Figure 3. BSCBLOCK タイプ C (ASC) ブロック ダイアグラム



[+] Feedback

Figure 4. SCBLOCK タイプ D (ASD) ブロック ダイアグラム



機能説明

PSoC スイッチド キャパシタ ブロックの詳細な説明は、*アプリケーション ノート AN2041* を参照してください。このユーザ モジュールの目的は、スイッチド キャパシタ アナログ PSoC ブロックを使用したカスタムアナログ機能の設計を可能にすることです。

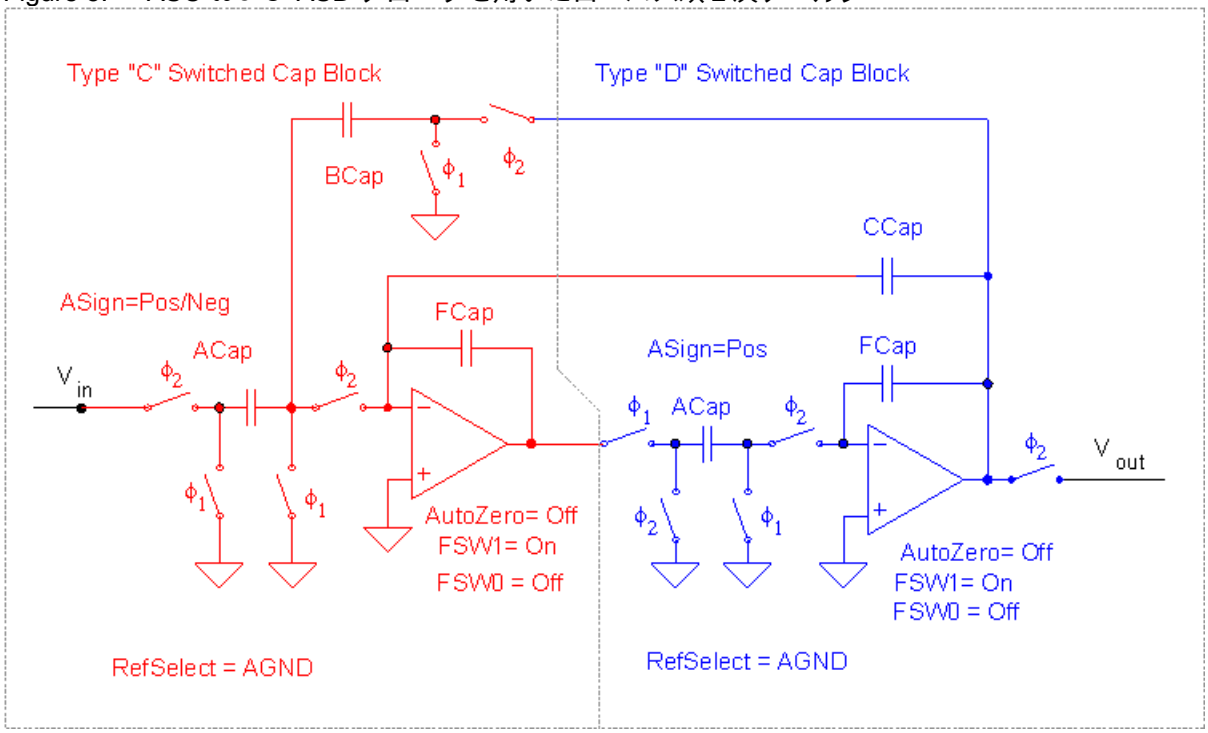
ASC と ASD タイプの SCBLOCK ユーザ モジュールの間には、いくつかの違いがあります。

- CCap 分岐は、ASC タイプの SCBLOCK で入力として、また ASD タイプの SCBLOCK で出力として使用されます。
- ASC タイプの SCBLOCK は、BCap の通常のオートゼロ操作を許可します。
- ASC タイプの SCBLOCK では 3 つの入力から選択可能で、ASD タイプの SCBLOCK では 2 つの入力から選択可能です。

ASC および ASD の SCBLOCK は、組み合わせて 4 次フィルタを構成できるように設計されています。ASC ブロックは 4 次フィルタの入力の段階で使用され、ASD ブロックは出力の段階で使用されます。下図は、2 つのユーザ モジュール（ASC および ASD SCBLOCK ユーザ モジュール）が結合して、1 つのローパス 双 2 次フィルタを構成する様子を示しています。ASC および ASD タイプの SCBLOCK は、ASA および ASB タイプの SCBLOCK の改良バージョンです。ASC および ASD タイプの SCBLOCK は、入力の選択オプション数が多く、オプアンプの電気的特性も改良されています。

[+] Feedback

Figure 5. ASC および ASD ブロックを用いたローパス双 2 次フィルタ



DC 電気的特性と AC 電気的特性

下の表に別途注記がない限り、すべての制限値は T_A = -40°C ~ +85°C、V_{dd} = 5.0V +/- 5% を想定します。

Table 1. SCBLOCK の DC 電気的特性と AC 電気的特性

パラメータ	条件および注記	標準値	制限	単位
コンデンサの単位		70		fF
キャパシタ・マッチング		0.1		%
F_{max}	SCBLOCK のクロック [1]	2[1]		MHz
電源電流				
バイアス = 低		125		uA
バイアス = 中		280		uA
バイアス = 高		760		uA

電気的特性に関する注意事項

1. クロック信号は、アナログクロック マルチプレクサと SCBLOCK の間の 1:4 分周器を通ります。SCBLOCK に 2 MHz のクロックを供給するには、アナログクロック マルチプレクサに 8 MHz のクロックが接続されていなければなりません。

[+] Feedback

配置

SCBLOCK ブロックは、どのスイッチド キャパシタ PSoC ブロックにも配置できます。ASC および ASD タイプのブロックの性能は、配置によって異なります。

パラメータおよびリソース

このセクションでは、SCBLOCK 用のパラメータについて説明します。注：いくつかのパラメータは、ASC、ASD タイプのどちらかのみに適用されます。

FCap

- **16** - コンデンサを 16 単位に設定
- **32** - コンデンサを 32 単位に設定

ClockPhase

- **Norm** - 標準的フェーズ
- **Swap** - φ1 と φ2 クロックを交換

ASign

- **Pos** - コンデンサは φ1 のとき A 入力に接続し、φ2 のときはリファレンス入力に接続
- **Neg** - コンデンサは φ1 のときリファレンス入力に接続し、φ2 のときは A 入力に接続

ACap

ACap 値に 0 ~ 31 単位を設定します。

ACMux (ASC)

ACap に接続する A 入力と、CCap に接続する C 入力を選択します。SCBLOCK が特定のブロックに配置されたとき、PSoC デザイナは利用可能な接続オプションを示します。

Note このパラメータは、SCBLOCK コンポーネントが ASC ブロックに配置されているときのみ適用可能です。

AMux (ASD)

ACap に接続される A 入力を選択します。SCBLOCK が特定のブロックに配置されたとき、PSoC デザイナは利用可能な接続オプションを示します。

Note このパラメータは、SCBLOCK コンポーネントが ASD ブロックに配置されているときのみ適用可能です。

BCap

BCap 値に 0 ~ 31 単位を設定します。

AnalogBus (アナログ バス)

- **ディスエーブル** - アナログ出力は、アナログ出力バスに接続されないため、他のユーザ モジュールが出力バスを使用できます。
- **イネーブル** - アナログ出力はそのコラムのアナログ出力バスに接続され、アナログバッファにルーティングされることもあります。

Note アナログブロック出力は、コラム毎に 1 つのみアナログバスに接続できます。

CompBus (コンパレータ バス)

- **ディスエーブル** - コンパレータ出力はコンパレータ出力バスに接続されていないため、同じコラムに配置されている他のユーザ モジュールがこれを使用できます。
- **イネーブル** - コンパレータ出力はコラムのコンパレータ出力バスに接続されており、デジタルリソースまでルーティングされることもあります。

Note コラム毎に 1 つのアナログブロック コンパレータ出力のみ、コンパレータバスに接続できます。

AutoZero (オートゼロ)

- **オフ** - オートゼロ機能は無効です。
- **オン** - オートゼロ機能は有効です。出力は φ1 のとき負の入力に接続され、オプアンプのオフセットインピーダンスを測定します。φ2 のとき、実際のシステムはこのオフセットに適合するよう補正されます。

CCap

CCap 値に 0 ~ 31 単位を設定します。

ARefMux

- **AGND** - ACap は φ2 のとき AGND に接続されます。
- **REFHI** - ACap は φ2 のとき REFHI に接続されます。
- **REFLO** - ACap は φ2 のとき REFLO に接続されます。
- **ComparatorBus_x** - φ2 の間、ACap は、コンパレータが HIGH の場合 REFHI に、コンパレータが LOW の場合 REFLO に接続されます。

FSW1

- **オフ** - フィードバックループから FCap への接続は外されています。
- **オン** - FCap はフィードバックループに接続されています。

FSW0

- **オフ** - φ1 のとき、FCap は放電されません。
- **オン** - φ1 のとき、FCap は放電されます。

BSW (ASD)

- **オフ** - BCap はコンデンサとして接続されません。
- **オン** - BCap はコンデンサとして接続されます。

BMux

BCap に接続される B 入力を選択します。

Power (出力)

- **オフ** - SCBLOCK はパワーダウン中です。
- **低** - SCBLOCK は最低パワーに設定されています。
- **中** - SCBLOCK は中程度のパワーに設定されています。
- **高** - SCBLOCK はフルパワーに設定されています。

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ハイ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、各関数のインターフェイスと、それに関する 「include」ファイルによって提供される定数を示します。

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 デザイナが、そのプロジェクトのユーザ モジュールの第 1 インスタンスに SCBLOCK_1 を割り当てます。これは識別子の構文ルールに従った任意の名前に変更できます。指定されたサンプルの名前は、すべてのグローバル機能名、変数、定数記号の接頭辞になります。次の説明では、簡略にするために、インスタンス名は省略されて単に 「SCBLOCK」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。関数を呼び出す場合、呼出し後に A と X の値が必要になるならば、必ず呼び出し前に A と X の値を保存してください。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、このポリシーを認識しなければいけません。ユーザ モジュール API 関数の中には、A と X を変更しないものもありますが、今後も変更されないという保証はありません。

SCBLOCK_Start

説明

出力レベルを設定し、このユーザ モジュールで必要な初期化をすべて実行します。

C プロトタイプ :

```
void  SCBLOCK_Start (BYTE bPowerSetting)
```

アセンブラ :

```
mov   A, bPowerSetting
lcall  SCBLOCK_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、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 及びアセンブリ言語で提供されたシンボル名と関連する値は、次の表に記載されています。

シンボル名	値
SCBLOCK_OFF	0
SCBLOCK_LOWPOWER	1
SCBLOCK_MEDPOWER	2
SCBLOCK_HIGHPOWER	3

出力レベルは、アナログ性能に影響します。適正な出力設定は、アプリケーション毎に指定します。

戻り値 :

なし

特殊作用：

この関数によって、A および X レジスタが変更される場合があります。

SCBLOCK_SetPower

説明

SCBLOCK ユーザ モジュールの出力レベルを設定します。

C プロトタイプ：

```
void  SCBLOCK_SetPower (BYTE bPowerSetting)
```

アセンブラ：

```
mov   A, bPowerSetting
lcall  SCBLOCK_SetPower
```

パラメータ：

bPowerSetting: Start エントリ ポイントで使用した bPowerSetting パラメータと同じ。これにより、ブロックを動作させながら、出力レベルを変更できます。

戻り値：

なし

特殊作用：

この関数によって、A および X レジスタが変更される場合があります。

SCBLOCK_Stop

説明

出力レベルを 0FF に設定します。

C プロトタイプ：

```
void  SCBLOCK_Stop()
```

アセンブラ：

```
lcall  SCBLOCK_Stop
```

パラメータ：

なし

戻り値：

なし

副作用：

この関数によって、A および X レジスタが変更される場合があります。

ファームウェア ソースコードの例

C 言語で書かれた SCBLOCK を用いたサンプルプログラムの例を示します。

```
#include "SCBLOCK.h"
void main(void)
{
    SCBLOCK_Start(SCBLOCK_HIGHPOWER);    // Turn on SCBlock power
    // User code
    SCBLOCK_Stop();                          // Turn off SCBlock power
}
```

このサンプルコードは、SCBLOCK ユーザ モジュールを起動し、255 カウンド待ち、SCBLOCK を停止します。

```
;;; Sample Code for the SCBLOCK

include "SCBLOCK.inc"
include "m8c.inc"
export _main

_main:
   mov  A,SCBLOCK_HIGHPOWER      ; Set Power
   call SCBLOCK_Start            ; Turn on SCBLOCK

   mov  A,ffh                    ; Set loop counter
loop1:
   dec  A
   jnz  loop1
   call SCBLOCK_Stop             ; Turn off SCBlock
   ret
```

コンフィグレーション レジスタ

Table 2. ブロック SCBLOCK: レジスタ CR0

ビット	7	6	5	4	3	2	1	0
値	FCap	ClockPhase	ASign	ACap				

Table 3. ブロック SCBLOCK: レジスタ CR1

ビット	7	6	5	4	3	2	1	0
ASC	ACMux			BCap				
ASD	AMux			BCap				

ACMux は、ブロックを ASC ブロックに配置する場合に使用します。AMux は、ブロックを ASD ブロックに配置する場合に使用します。どちらのフィールド値も、ユーザによる入力接続によって異なります。

Table 4. ブロック SCBLOCK: レジスタ CR2

ビット	7	6	5	4	3	2	1	0
値	AnalogBus (アナログバス)	CompBus (コンパレータ バス)	AutoZero (オートゼロ)	CCap				

Table 5. ブロック SCBLOCK: レジスタ CR3

ビット	7	6	5	4	3	2	1	0
ASC	ARefMux		FSW1	FSW0	BMuxASC		Power (出力)	
ASD	ARefMux		FSW1	FSW0	BSW	BMuxASD	Power (出力)	

BMuxASC はブロックが ASC ブロックに配置されているときに使用され、ユーザが入力をどのように接続するかによって異なります。BMuxASD はブロックが ASD ブロックに配置されているときに使用され、ユーザが入力をどのように接続するかによって異なります。BSW は、ブロックを ASD ブロックに配置する場合に使用します。この値は、BCap スイッチのアクティブ・非アクティブを指定します。

Copyright © 2002-2011 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.