8-Bit シフト レジスタのデータシート SHIFTREG8 V 1.0

Copyright © 2009-2011 Cypress Semiconductor Corporation. All Rights Reserved.

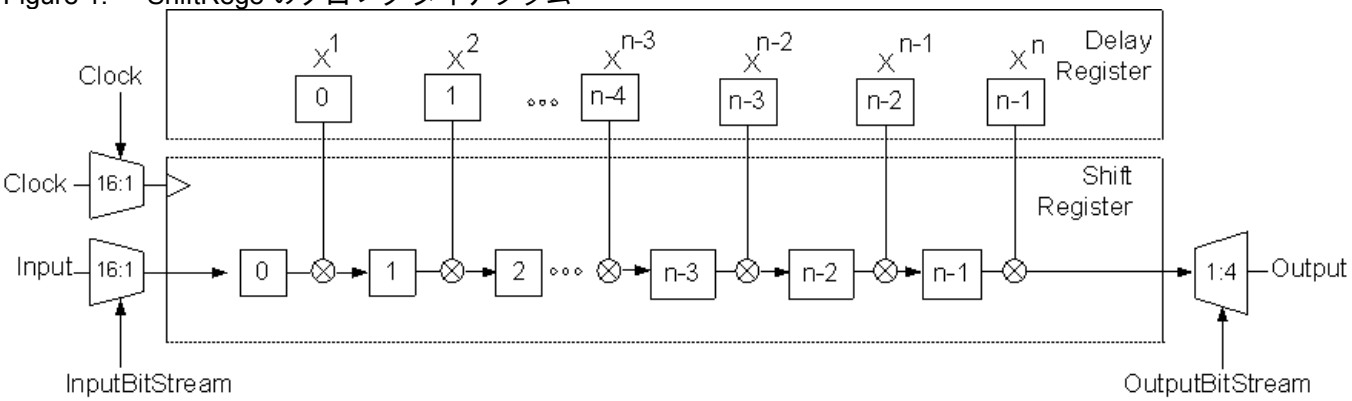
リソース	PSoC® ブロック			API メモリ (バイト)		ピン (外部入出力ごと)
	デジタル	アナログ CT	アナログ SC	Flash	RAM	
CY8C29/27/24/22/21xxx、CY8C23x33、CY8CLED02/04/08/16、CY8CLED03D/04D、CY8CTST110、CY8CTMG110、CY8CTST120、CY8CTMG120、CY8CTMA120、CY8C21x45、CY8C22x45、CY8CPLC20、CY8CLED16P01						
	0	0	0	25	0	1 ~ 4
CYWUSB6953	0	0	0	25	0	1 ~ 4
	0	0	0	25	0	1 ~ 4

特性および概要

- 8-bit シフト レジスタは、PSoC ブロック 1 つを使用
- 48 MHz 以内のクロック源を使用可能
- 最高 24 MHz までのクロックに対するデータ入力
- プログラマブルな遅延サイクル値
- シリアル出力ビットストリーム

ShiftReg8 ユーザ モジュールは、入力ビットストリームを遅延する、モジュール式リニア フィードバック シフト レジスタ (LFSR) の一種です。PSoC ブロック クロック最大 8 つで、遅延サイクル値を指定して、出力の遅延を定義できます。

Figure 1. ShiftReg8 のブロック ダイアグラム



機能説明

このユーザ モジュールは、周波数シフト キーイング (FSK) などのアプリケーションで、デジタル信号をシフトします。デジタル信号は、ユーザ モジュールを通して渡される際に、指定した DB クロック期間だけ遅延が可能です。

入力に対して遅延される出力ビット シーケンスの生成は、ShiftReg8 PSoC ブロックの遅延および制御レジスタを使用して定義および制御します。遅延レジスタは、書き込み専用です。遅延レジスタで 1 を持つ最上位ビット位置によって、遅延サイクルの数が決定されます。

デジタル PSoC ブロックの内部接続ハードウェアの構成が、単一の ShiftReg8 ユーザ モジュールとして PSoC ブロックの動作を調整します。遅延レジスタは、1 ～ 8 つの PSoC ブロック クロックから、ShiftReg の遅延サイクルをセットアップします。

ShiftReg ユーザ モジュールでは、その起動前に遅延を初期化する必要があります。レジスタの更新中は、ShiftReg ユーザ モジュールが停止されます。

DC 電気的特性と AC 電気的特性

Table 1. CY8C22x45 デバイス ファミリにおける ShiftReg の AC 電気的特性

パラメータ	典型値	制限	単位	条件および注記
最大クロック周波数	--	48[1]	MHz	4.75V < Vdd < 5.25V
	--	24[1]	MHz	3.0V < Vdd < 4.75V
最大入力周波数	--	24[1]	MHz	4.75V < Vdd < 5.25V
最大出力周波数	--	24[1]	MHz	Vdd=5.0V および 24 MHz の入力クロック

電気的特性に関する注意事項

1. 入力または出力が、グローバル バスを通して迂回される場合は、周波数は、最大 12 MHz に制限されます。

配置

ShiftReg は、8 ビットの分解能ごとに、1 つのデジタル PSoC ブロックを消費します。各ブロックには、配置中または配置後、デバイス エディタで表示されるシンボリック名が与えられます。API は、ユーザが割り当てたインスタンス名とブロック名を使ってすべてのレジスタ名を確認し、API の include ファイルによって ShiftReg8 レジスタへの直接アクセスを提供します。ブロック名は、SHIFTREG8 です。

パラメータおよびリソース

Clock (クロック)

ShiftReg8 は 16 個の可能なソースのうちのいずれかによってクロック処理されます。外部のピンまたは異なる PSoC ブロックが生成するクロック関数にクロック入力を接続する場合は、グローバル I/O バスを使用することが可能です。48 MHz クロック、CPU_32 kHz クロック、分周クロック (24V1 または 24V2) のいずれか、別の PSoC ブロックの出力を、クロック入力として指定できます。

InputBitStream

利用可能なデータ経路から、データ入力ソースを選択します。

OutputBitStream

出力は、4 つのグローバル出力バスのいずれかに接続することと、無効にすることができます。

Length (長さ)

ShiftReg の遅延サイクルを設定します。このパラメータは、遅延サイクル レジスタの値を設定します。可能な値は、以下の表に記載されています。

パラメータ	説明
1	入力データが、PSoC ブロック クロック 1 つまで遅延されます。
2	入力データが、PSoC ブロック クロック 2 つまで遅延されます。
3	入力データが、PSoC ブロック クロック 3 つまで遅延されます。
4	入力データが、PSoC ブロック クロック 4 つまで遅延されます。
5	入力データが、PSoC ブロック クロック 5 つまで遅延されます。
6	入力データが、PSoC ブロック クロック 6 つまで遅延されます。
7	入力データが、PSoC ブロック クロック 7 つまで遅延されます。
8	入力データが、PSoC ブロック クロック 8 つまで遅延されます。

ClockSync (クロック同期)

PSoC デバイスでは、システム クロックに加え、デジタル ブロックがクロック源を提供することができます。デジタル クロック源は、複数をリップル形で連結することも可能です。こうすることで、システムクロックに対するスキューが発生します。このようなスキューは、さまざまなデータ経路の最適化、特にシステム バスに適用される最適化のため、PSoC デバイス ファミリ CY8C22x45 および CY8C21345 でより重要となります。このパラメータは、クロック スキューを制御するために使用され、PSoC ブロック レジスタ値を読み書きする場合の、正しい動作を保証します。このパラメータの適切な値は、以下の表で決定されます。

ClockSync 値	説明
Sync to SysClk (SysClk への同期)	2 以上で除算される 24 MHz (SysClk) 派生クロック ソースでこの設定を使用します。たとえば、VC1、VC2、VC3 (VC3 が SysClk によって駆動される場合)、32KHz、SysClk ベースのクロックで動作するデジタル PSoC ブロックがあります。正しい同期が行われるようにするため、外部で生成されたクロック源もこの値を使用します。
Sync to SysClk*2 (SysClk*2 への同期)	結果の周波数が 48 MHz でない限り (言い換えると、すべての除算結果が 1 になる場合)、48 MHz (SysClk*2) ベースのクロックでは、この設定を使用します。
Use SysClk Direct (SysClk を直接使用する)	24 MHz (SysClk/1) クロックが求められる場合、この設定を使用します。この設定では、実際にはクロックを同期させず、システム クロック自身へのスキューの少ないアクセスを提供します。この設定を選択すると、上述の Clock パラメータの設定を上書きします。組み合わせた全除算値の最終結果が 24 MHz 出力を生成する場合は、VC1、VC2、VC3、またはデジタル ブロックの代わりに常に使用してください。
Unsynchronized (同期せず)	48 MHz (SysClk*2) 入力が選択される場合に使用します。 同期されていない入力が求められる場合、この設定を使用します。一般に、カウンタが割り込みを発生させる目的だけに使用される場合にのみ、この設定を使用することを推奨します。

アプリケーション プログラミング インタフェース

SHIFTREG8_Start

説明

ShiftReg8 ユーザ モジュールの動作をイネーブルにします。モジュールを開始する前に、遅延サイクル値を初期化する必要があります。

C プロトタイプ

```
void  SHIFTREG8_Start(void)
```

アセンブリ

```
lcall  SHIFTREG8_Start
```

パラメータ

なし

戻り値

なし

副作用

ShiftReg8 が開始されると、シード レジスタに書き込まれたシード値は、シフト レジスタにラッチされません。

API セクションの冒頭にある注意事項 ** を参照してください。

SHIFTREG8_Stop

説明

ShiftReg8 ユーザ モジュールをディスエーブルにします。

C プロトタイプ

```
void  SHIFTREG8_Stop(void)
```

アセンブリ

```
lcall  SHIFTREG8_Stop
```

パラメータ

なし

戻り値

なし

副作用

シード レジスタにシード値を書き込むと、シフト レジスタにシード値がラッチされます。

API セクションの冒頭にある注意事項 ** を参照してください。

SHIFTREG8_SetLength

説明

遅延サイクル値とともに遅延サイクル レジスタ (DR1) をロードします。レジスタの更新中は、ShiftReg ユーザ モジュールが停止されます。

C プロトタイプ

```
void  SHIFTREG8_SetLength(BYTE bLength)
```

アセンブリ

```
mov   A, [bLength]
lcall  SHIFTREG8_SetLength
```

パラメータ

bLength: 8-bit 遅延サイクル値。遅延サイクルは、1 ～ 8 に設定できます。例えば、bit0 が設定されると、長さは 1 になります。bit1 が設定されると、長さは 2 になります。この長さ値は、パラメータ表にリストされている記号名を通して渡されます。この記号名は、次の形式になります。ShiftReg8_LENGTH_x ここで 「x」は、長さ (1、2、3 など) です。

記号名	値	説明
SHIFTREG8_LENGTH_1	1	入力データが、PSoC ブロック クロック 1 つまで遅延されます。
SHIFTREG8_LENGTH_2	2	入力データが、PSoC ブロック クロック 2 つまで遅延されます。
SHIFTREG8_LENGTH_3	3	入力データが、PSoC ブロック クロック 3 つまで遅延されます。
SHIFTREG8_LENGTH_4	4	入力データが、PSoC ブロック クロック 4 つまで遅延されます。
SHIFTREG8_LENGTH_5	5	入力データが、PSoC ブロック クロック 5 つまで遅延されます。
SHIFTREG8_LENGTH_6	6	入力データが、PSoC ブロック クロック 6 つまで遅延されます。
SHIFTREG8_LENGTH_7	7	入力データが、PSoC ブロック クロック 7 つまで遅延されます。
SHIFTREG8_LENGTH_8	8	入力データが、PSoC ブロック クロック 8 つまで遅延されます。

戻り値

なし

副作用

レジスタの更新中は、ShiftReg ユーザ モジュールが停止されます。

API セクションの冒頭にある注意事項 ** を参照してください。

SHIFTREG8_bReadRegister

説明

この API は、LFSR 値を DR2 に転送する、DR0 レジスタを読み取ります。次にこの API は値を読み取り、DR2 にそれを返します。この API を呼び出す前に、シフト レジスタを停止させる必要があります。

C プロトタイプ

```
BYTE  ShiftReg8_bReadRegister (void)
```

アセンブリ

```
lcall  ShiftReg8_bReadRegister
```

パラメータ

なし

戻り値

この関数は、シフト レジスタの内容であるバイトを返します。

副作用

API セクションの冒頭にある注意事項 ** を参照してください。

SHIFTREG8_WriteRegister

説明

この API は、渡された値を DR2 レジスタに書き込みます (同時に値は LFSR にも転送されます)。この API を呼び出す前に、シフト レジスタを停止させる必要があります。

C プロトタイプ

```
void  ShiftReg8_WriteRegister(bValue)
```

アセンブリ

```
mov A, [bValue]
lcall  ShiftReg8_WriteRegister
```

パラメータ

bValue: シフト レジスタに書き込まれる値です。

戻り値

なし

副作用

API セクションの冒頭にある注意事項 ** を参照してください。

ファームウェア ソースコードの例

以下の例では、C およびアセンブリの両コード間の対応は単純で直接的です。期間および比較値として示される値は、レジスタがゼロをベースにしており、ダウン カウント サイクルでゼロが最終カウントになるため、基本値から 1 だけずれます。ユーザ モジュール API に対して、スタックではなく単純に A レジスタで 1 バイトのパラメータを渡しているのは、性能を最適化するためで、アセンブラおよび C コンパイラの両方で使われています。C コンパイラは、SHIFTREG8 .h ファイルに #pragma fastcall 宣言を見つけると、スタックで引数をプッシュする代わりに、「INT」タイプでこのメカニズムを導入します。

以下は、API の使用を示すアセンブリ言語ソースです。

```
;*****************************************************************
;  Setup the SHIFTREG8 to generate an 8-bit maximal sequence.
;
;*****************************************************************
include "SHIFTREG8.inc"
export  SetupSHIFTREG8

SetupSHIFTREG8:
    ; load the ShiftReg8 Delay cycle value
    mov   A, SHIFTREG8_LENGTH_8 ; delay 8 PSoC block clocks
    call  SHIFTREG8_SetLength
    ;start the ShiftReg8
    call  SHIFTREG8_Start

    ret
```

C での同じコードは以下のようになります。

```
#include "SHIFTREG8.h"

void SetupSHIFTREG8(void)
{
// load the ShiftReg8 Delay cycle value
SHIFTREG8_SetLength(SHIFTREG8_LENGTH_8);
// start the ShiftReg8
SHIFTREG8_Start();
}
```

コンフィグレーション レジスタ

レジスタを通して ShiftReg8 ユーザ モジュールがパーソナライズされ、パラメータ化されます。以下の表に、定数としてのレジスタ値と、名前付きビットフィールドとしてのパラメータを短い説明とともに示します。これらのレジスタのシンボル名は、ユーザ モジュール インスタンスの C およびアセンブリ言語インタフェース ファイル (「.h」および 「.inc」ファイル) で定義されます。

Table 2. SHIFTREG8_FUNC_REG

ビット	7	6	5	4	3	2	1	0
値	0	0	1	0	0	0	1	0

Table 3. SHIFTREG8_INPUT_REG

ビット	7	6	5	4	3	2	1	0
値	InputBitStream[3:0]				Clock[3:0]			

InputBitStream は、16 のソースのいずれかからデータ入力を選択します。クロックは、16 のクロック ソースからクロック入力を選択します。どちらのパラメータも、デバイス エディタで設定します。

Table 4. SHIFTREG8_OUTPUT_REG

ビット	7	6	5	4	3	2	1	0
値	AuxClk (補助クロック)	0	0	0	0	OutputBitStream		

デバイス エディタでユーザ モジュールの 「ClockSync」パラメータは、AuxClk ビットの値を決定します。OutputBitStream は、4 つのグローバル バスのいずれかから出力を選択します。このパラメータはデバイス エディタで設定します。

Table 5. SHIFTREG8_DELAY_REG

ビット	7	6	5	4	3	2	1	0
値	Delay Cycle Number (遅延サイクル数)							

遅延サイクル数は、ShiftReg8 遅延レジスタです。ShiftReg8 API を使って変更できます。遅延レジスタは、書き込み専用です。遅延レジスタで 1 を持つ最高位ビットが、実際の遅延サイクルを決定します。

Table 6. SHIFTREG8_CONTROL0_REG

ビット	7	6	5	4	3	2	1	0
値	0	0	0	0	0	0	0	Enable (イネーブル)

イネーブル ビットが設定されると、ShiftReg8 が有効であることを示します。ShiftReg8 API を使って変更できます。

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 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ています。このセクションでは、ユーザ 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 © 2009-2011 © Cypress Semiconductor Corporation. 本文書に記載される情報は、予告なく変更される場合があります。Cypress Semiconductor Corporation (サイプレス セミコンダクタ社) は、サイプレス 製品に組み込まれた回路以外のいかなる回路を使用することに対しても一切の責任を負いません。特許またはその他の権限下で、ライセンスを譲渡または暗示することもありません。サイプレス 製品は、サイプレスとの書面による合意に基づくものでない限り、医療、生命維持、救命、重要な管理、または安全の用途のために使用することを保証するものではなく、また使用することを意図したものでもありません。さらにサイプレスは、誤動作や故障によって使用者に重大な傷害をもたらすことが合理的に予想される、生命維持システムの重要なコンポーネントとしてサイプレス製品を使用することを許可していません。生命維持システムの用途にサイプレス 製品を供することは、製造者がそのような使用におけるあらゆるリスクを負うことを意味し、その結果サイプレスはあらゆる責任を免除されることを意味します。

PSoC® は Cypress Semiconductor Corp の登録商標であり、PSoC Creator ™ および Programmable System-on-Chip ™ は Cypress Semiconductor Corp. の商標です。本文書で言及するその他すべての商標または登録商標は、各社の所有物です。

全てのソース コード (ソフトウェアおよび / またはファームウェア) はサイプレス セミコンダクタ社 (以下「サイプレス」) が所有し、全世界の特許権保護 (米国およびその他の国)、米国の著作権法ならびに国際協定の条項により保護され、かつそれらに従います。サイプレスが本書面によりライセンシーに付与するライセンスは、個人的、非独占的かつ譲渡不能のライセンスであって、適用される契約で指定されたサイプレスの集積回路と併用されるライセンシーの製品のみをサポートするカスタム ソフトウェアおよび / またはカスタムファームウェアを作成する目的に限って、サイプレスのソース コードの派生著作物をコピー、使用、変更そして作成するためのライセンス、ならびにサイプレスのソース コードおよび派生著作物をコンパイルするためのライセンスです。上記で指定された場合を除き、サイプレスの書面による明示的な許可なくして本ソース コードを複製、変更、変換、コンパイル、または表示することは全て禁止されます。

免責条項 : サイプレス は、明示的または黙示的を問わず、本資料に関するいかなる種類の保証も行いません。これには、商品性または特定目的への適合性の黙示的な保証か゛含まれますが、これに限定されません。サイプレスは、本文書に記載される資料に対して今後予告なく変更を加える権利を留保します。サイプレスは、本文書に記載されるいかなる製品または回路を適用または使用したことによって生ずるいかなる責任も負いません。サイプレスは、誤動作や故障によって使用者に重大な傷害をもたらすことが合理的に予想される生命維持システムの重要なコンポーネントとしてサイプレス製品を使用することを許可していません。生命維持システムの用途にサイプレス製品を供することは、製造者がそのような使用におけるあらゆるリスクを負うことを意味し、その結果サイプレスはあらゆる責任を免除されることを意味します。

ソフトウェアの使用は、適用されるサイプレス ソフトウェア ライセンス契約によって制限され、かつ制約される場合があります。