

Myway

SimCoder™

User Manual

Powersim Inc.

Mywayプラス株式会社

SimCoder User Manual

Version 2020a

Release 2

Copyright © 2008-2021 Powersim Inc., Myway Plus Corporation

All rights reserved. No part of this manual may be photocopied or reproduced in any form or by any means without the written permission of Powersim Inc and Myway Plus Corporation.

Disclaimer

Powersim Inc. (Powersim) and Myway Plus Corporation (Myway) make no representation or warranty with respect to the adequacy or accuracy of this documentation or the software which it describes. In no event will Powersim and Myway or their direct or indirect supplies be liable for any damages whatsoever including, but not limited to, direct, indirect, incidental, or consequential damages of any character including, without limitation, loss of business profits, data, business information, or any and all other commercial damages or losses, or for any damages in excess of the list price for the license to the software and documentation.

お問い合わせ先

Myway プラス株式会社

〒222-0022 神奈川県横浜市西区花咲町 6-145 横浜花咲ビル

Tel 045-548-8836, Fax 045-548-8832

Email: sales@myway.co.jp

URL: <https://www.myway.co.jp/>

目次

1	SimCoder について	8
1.1	まえがき	8
1.2	SimCoder のシミュレーション制御設定(ハードウェア等)	9
1.3	コード生成用の素子	12
2	コード生成方法	13
2.1	概要	13
2.2	連続系システム	13
2.3	離散系システム	14
2.4	ハードウェアターゲット用のシステムコード生成	16
2.5	C コード生成	19
2.6	イベントコントロール付きシステム	23
3	サブシステムのコード生成	26
3.1	サブシステム	26
3.2	コード生成	28
4	イベントコントロール	29
4.1	基本概念	29
4.2	イベントコントロール素子	30
4.3	イベント付きサブ回路の制限	31
5	SimCoder ライブラリ	34
5.1	標準 PSIM ライブラリの素子	34
5.1.1	パラメータファイルブロックを用いたグローバルパラメータ設定	37
5.1.2	のこぎり波生成	40
5.2	イベントコントロール素子	40
5.2.1	入力イベント	41
5.2.2	出力イベント	41
5.2.3	デフォルトイベント	42
5.2.4	イベント接続	43
5.2.5	イベントブロック初回実行フラグ	43
5.3	グローバル変数	43
5.4	ハードウェア割り込み	46
5.5	SimCoder C Block	47
6	IQmath Library	50
6.1	概要	50
6.2	IQmath データタイプ、レンジ及び分解能	50

7	TI F2833x Hardware Target.....	51
7.1	概要.....	51
7.2	ハードウェア構成.....	53
7.3	DSP クロック.....	54
7.4	PWM 生成器.....	55
7.4.1	3-phase PWM 生成器 :	56
7.4.2	1-phase PWM 生成器 (外部位相シフト機能付きを含む) :	57
7.4.3	2-phase PWM 生成器 :	62
7.4.4	Single PWM 生成器(APWM) :	65
7.4.5	PWM ブロック間の同期	66
7.5	可変周波数 PWM.....	67
7.6	Start PWM と Stop PWM.....	67
7.7	トリップゾーンとトリップゾーンステート	68
7.8	A/D 変換器	69
7.9	デジタル入力とデジタル出力	73
7.10	UP/DOWN カウンタ	74
7.11	エンコーダとエンコーダステート	75
7.12	キャプチャとキャプチャステート	77
7.13	シリアル通信インターフェース (SCI)	78
7.13.1	SCI 設定	78
7.13.2	SCI 入力	79
7.13.3	SCI 出力	79
7.14	シリアルペリフェラルインターフェース(SPI)	80
7.14.1	SPI 設定	80
7.14.2	SPI デバイス	81
7.14.3	SPI 入力	83
7.14.4	SPI 出力	84
7.15	Controller Area Network (CAN) Bus	86
7.15.1	CAN Configuration	86
7.15.2	CAN Input	87
7.15.3	CAN Output.....	88
7.16	割込み時間	89
7.17	プロジェクト設定とメモリ配置.....	89
8	TI F2803x Hardware Target.....	93
8.1	概要.....	93
8.2	ハードウェア構成.....	95

8.3	DSP クロック	96
8.4	PWM 生成器	96
8.4.1	3-phase PWM 生成器 :	97
8.4.2	1-phase PWM 生成器 (外部位相シフト機能付きを含む) :	101
8.4.3	2-phase PWM 生成器 :	107
8.4.4	Single PWM 生成器(APWM) :	113
8.4.5	PWM ブロック間の同期	113
8.5	可変周波数 PWM	114
8.6	Start PWM と Stop PWM	115
8.7	トリップゾーンとトリップゾーンステート	116
8.8	A/D 変換器	117
8.9	コンパレータ	121
8.9.1	コンパレータ入力	122
8.9.2	コンパレータ出力	123
8.9.3	コンパレータ DAC	123
8.10	デジタル入力とデジタル出力	124
8.11	UP/DOWN カウンタ	125
8.12	エンコーダとエンコーダステート	126
8.13	キャプチャとキャプチャステート	127
8.14	シリアル通信インターフェース (SCI)	128
8.14.1	SCI 設定	128
8.14.2	SCI 入力	129
8.14.3	SCI 出力	130
8.15	シリアルペリフェラルインターフェース(SPI)	131
8.15.1	SPI 設定	131
8.15.2	SPI デバイス	132
8.15.3	SPI 入力	134
8.15.4	SPI 出力	135
8.16	Controller Area Network (CAN) Bus	136
8.16.1	CAN Configuration	137
8.16.2	CAN Input	138
8.16.3	CAN Output	139
8.17	割込み時間	140
8.18	プロジェクト設定とメモリ配置	140
9	TI F2837x Hardware Target	143
9.1	概要	143

9.2	ハードウェア構成.....	144
9.3	DSP クロック	146
9.4	PWM 生成器.....	146
9.4.1	3-phase PWM、1-phasePWM,1-phasePWM(phase shift) :	147
9.4.2	2-phase PWM 生成器 :	153
9.4.3	Single PWM(キャプチャと共用) :	158
9.4.4	PWM ブロック間の同期 Synchronization Between PWMBlocks.....	159
9.5	可変周波数 PWM.....	160
9.6	Start PWM と Stop PWM.....	161
9.7	トリップゾーンとトリップゾーンステート	161
9.7.1	PWM トリップゾーン	161
9.7.2	トリップゾーンステート	166
9.8	ADC 電圧リファレンス	166
9.9	A/D 変換器	167
9.10	コンパレータ	168
9.10.1	コンパレータ入力	169
9.10.2	コンパレータ DAC.....	171
9.11	D/A コンバータ	173
9.12	デジタル入力とデジタル出力	174
9.13	デジタル入力サンプル時間.....	175
9.14	入力 X-BAR,出力 X-BAR,PWM X-BAR	176
9.14.1	入力 X-BAR	176
9.14.2	出力 X-BAR	177
9.14.3	PWM X-BAR	178
9.15	エンコーダとエンコーダステート、エンコーダインデックス/ストローブポジ ション	178
9.16	UP/DOWN カウンタ	180
9.17	キャプチャとキャプチャステート	181
9.18	シリアル通信インターフェース (SCI)	182
9.18.1	SCI 設定	182
9.18.2	SCI 入力	183
9.18.3	SCI 出力	183
9.19	シリアルペリフェラルインターフェース(SPI)	184
9.19.1	SPI 設定	184
9.19.2	SPI デバイス	185
9.19.3	SPI 入力	186

9.19.4	SPI 出力	188
9.20	Controller Area Network (CAN) Bus	189
9.20.1	CAN Configuration	189
9.20.2	CAN Input	190
9.20.3	CAN Output	191
9.20.4	CAN Remote Request	192
9.21	割り込み時間	192
9.22	プロジェクト設定とメモリ配置	193
10	PE-Expert4 Hardware Target	195
10.1	概要	195
10.2	DSP ボード	195
10.2.1	スタート・ストップ PWM	195
10.2.2	LED 出力	196
10.2.3	タイマー割り込み優先設定	196
10.3	PEV ボード	197
10.3.1	PEV ボード設定	197
10.3.2	PWM 発生器	198
10.3.3	A/D 変換器	201
10.3.4	デジタル入力・キャプチャ・カウンタ	202
10.3.5	デジタル出力	203
10.3.6	エンコーダ	204
10.4	PSPWM ボード	205
10.4.1	PSPWM ボード構成	205
10.4.2	PSPWM ボードの PWM 生成器	206
10.4.3	PSPWM 拡張ボード上の素子	211
10.4.4	A/D コンバータ	211
10.4.5	デジタル入出力	212
10.5	ADC ボード	212
10.6	PE-Expert4 ランタイムライブラリ	213

1 SimCoder について

1.1 まえがき

SimCoderは制御システムのコードを自動的に生成するためのPSIMシミュレータの追加モジュールです。SimCoderを利用して、PSIMシミュレータでシステムのシミュレーションができ、Mywayプラス製の実機コントローラPE-Expert4及びTI社製高速浮動小数点型DSPTMS320F2833x、F2803xなどのハードウェアプラットフォームに合わせたCコードを自動的に生成することができます。

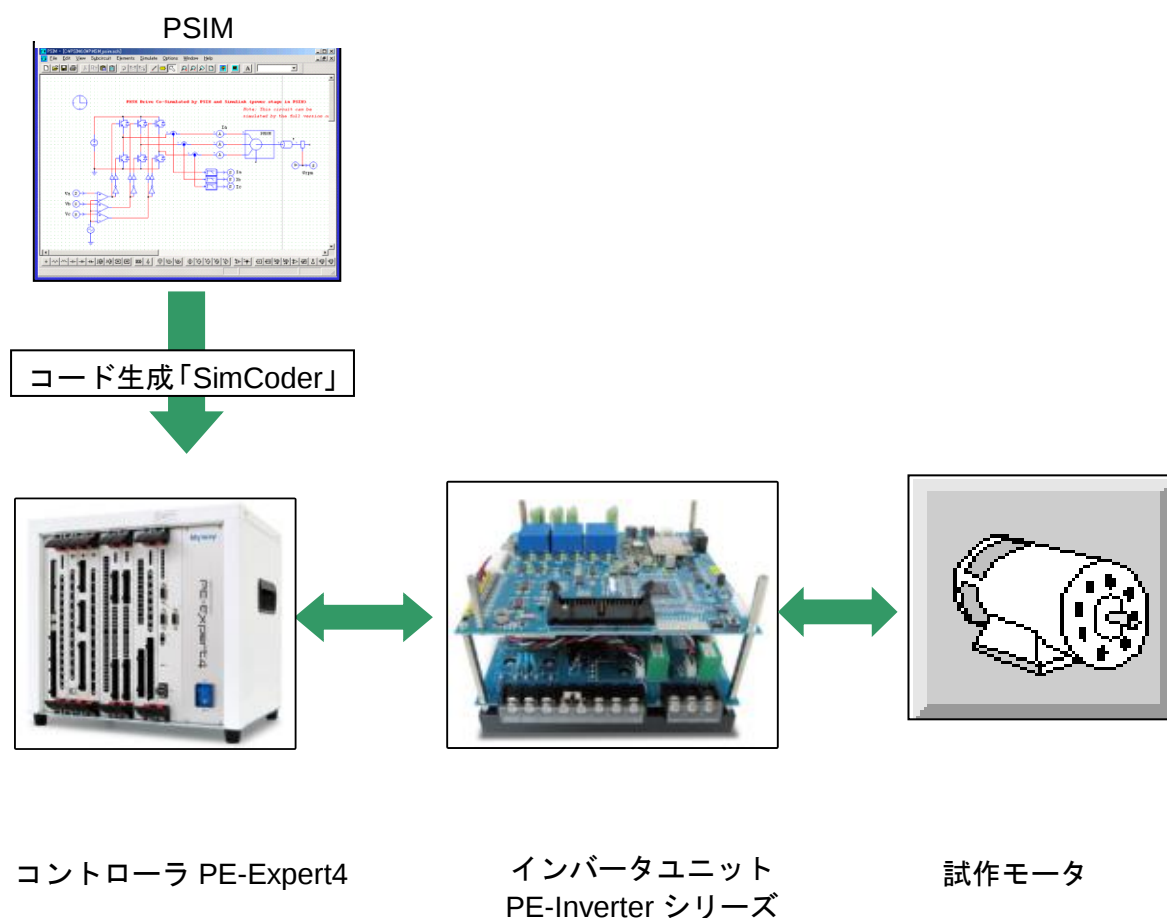


図 1.1 SimCoder と PE-Expert4 を利用した実験システム

SimCoderを利用して自動的にコードを作成すると、開発時間とコストを大幅に縮小することが可能になります。このマニュアルは、SimCoderを使用する方法について記述します。

1.2 SimCoder のシミュレーション制御設定(ハードウェア等)

SimCoder の設定は次の図のようにシミュレーション制御のウィンドウの SimCoder タブで、正しくコード生成できるよう各項目設定を行います。

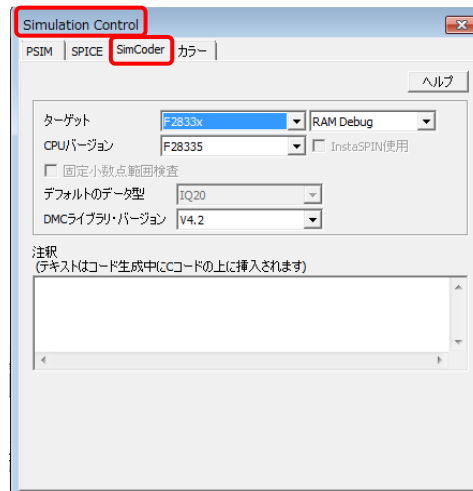


図 1.2 シミュレーション制御ウィンドウ SimCoder タブ画面

ハードウェアターゲット

- **PE-Expert4** : Myway プラス 製の PE-Expert4 DSP 開発プラットフォームで利用できる Cコード を生成します。**素子 → SimCoderコード自動生成 → PE-Expert4 Target** メニューを選択して PE-Expert4 に対応している素子を選ぶことができます。(利用するためには SimCoder モジュールに加えて PE-Expert4 **ターゲット** モジュールが追加オプションとして必要になります。)
- **F2833x** : TI 社製 浮動点 TMSF2833x シリーズの DSP を使用したハードウェアです。**素子 → SimCoderコード自動生成 → F2833x ターゲット** メニューを選択して TMS320F2833x に対応している素子を選ぶことができます。(利用するためには SimCoder モジュールに加えて TI F2833x **ターゲット** モジュールが追加オプションとして必要になります。)
- **F2803x** : TI 社製 浮動点 TMSF2803x シリーズの DSP を使用したハードウェアです。**素子 → SimCoderコード自動生成 → F2803x ターゲット** メニューを選択して TMS320F2803x に対応している素子を選ぶことができます。(利用するためには SimCoder モジュールに加えて TI F2803x **ターゲット** モジュールが追加オプションとして必要になります。)
- **F2837x** : TI 社製 浮動点 TMSF2837x シリーズの DSP を使用したハードウェアです。**素子 → SimCoderコード自動生成 → F2837x ターゲット** メニューを選択して TMS320F2837x に対応している素子を選ぶことができます。(利用するためには SimCoder モジュールに加えて TI F2837x **ターゲット** モジュールが追加オプションとして必要になります。)
- **ターゲット無し** : シミュレーションのみ。ハードウェアターゲット設定なし。

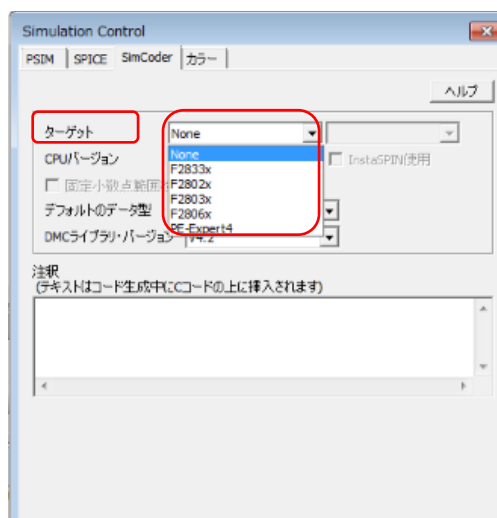


図 1.3 ターゲット選択画面

プロジェクトの構成

- PE-Expert4ターゲットに対してはPE-ViewX
- F2833x,F2803xターゲットに対してはRAM Debug,RAM Release,Flash ReleaseまたはFlash RAM Releaseを設定できます。

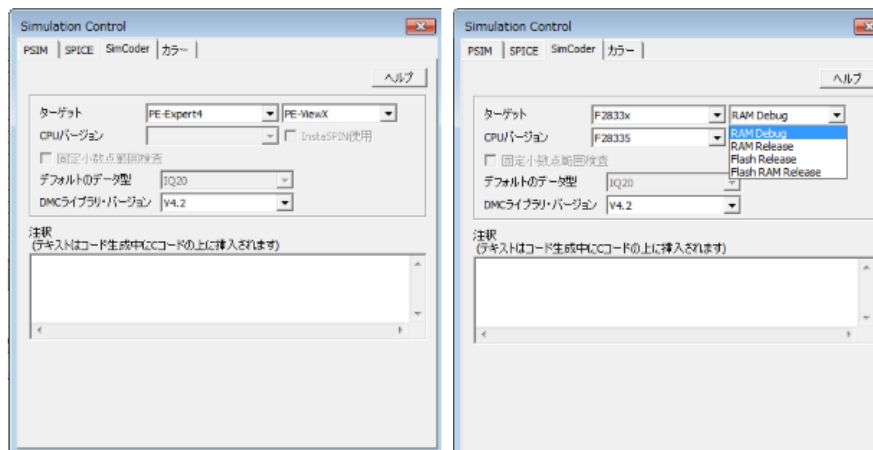


図 1.4 プロジェクト構成設定画面

CPUバージョン

- F2833xターゲットに対するCPUバージョンは：F28335,28334,28332です。
- F2803xターゲットに対するCPUバージョンは：F28035,28034,28033,28032,28031,28030です。
- F2837xターゲットに対するCPUバージョンは：28374S/D, 28375S/D, 28376S/D, 28377S/D, 28379S/D.です。

- PE-Expert4に対するCPUバージョンは：PE-Expert4は、Mywayプラス株式会社が製作したDSP開発プラットフォームです。
- TI社の浮動小数点DSP TMS320C6657と
- MywayプラスのPE-OSライブラリを使用しています。
-
-

固定小数点範囲検査

- これはF2803x ターゲットの場合のみ使用できます。チェックボックスにチェックを入れるとSimCoderはシミュレーションデータをチェックしデータ範囲のリストを作成します。このリストは範囲境界に近いものもしくは超えているデータはハイライトされています。

デフォルトのデータ型

- ハードウェアターゲットが浮動点型の場合に自動的に選択されます。ハードウェアターゲットが固定点型かNoneの場合、プルダウンメニューから選択できます。
- ターゲット無しの場合、次の1つとなります。： Float, Integer, IQ1, IQ2, ..., IQ30

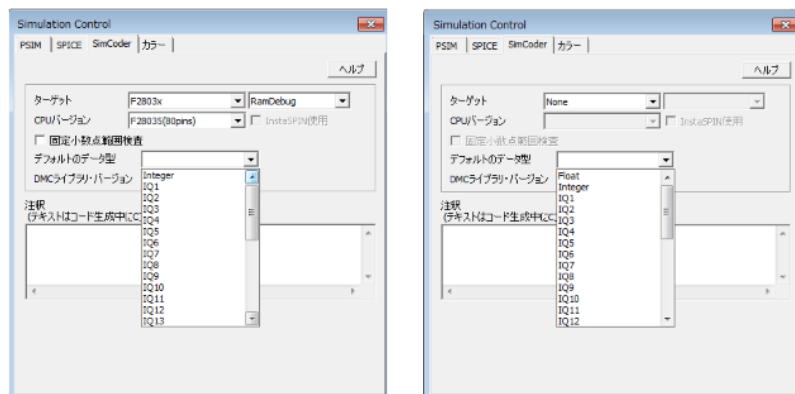


図 1.5 デフォルトデータ設定画面

DMCライブラリバージョン

- Texas InstrumentのDMC(Digital Motor Control)ライブラリがモーター制御ユーザーのためにC2000 DSPとして開発されたC言語の関数(またはマクロ)として構成されています。
- このリソースの活用するためには、SimCoderはコード生成のためのPSIM素子ライブラリへDMCライブラリ関数を組み込みます。
- これらのマクロはTIからリリースされた違ったバージョンがあります。SimCoderはバージョンV4.0, V4.1, V4.2を設定できます。一度バージョンを選択すると他のバージョンのマクロは無視されますのでご注意ください。

コメント

- コメント領域がSimCoderタブの下の方にあり、コメントとして追加できます。ここに入力したテキストはCコードの先頭にコメントとして追記されます。

1.3 コード生成用の素子

素子 → イベント制御 および **素子 → SimCoderコード自動生成**メニューのすべての素子がコード生成に使用できます。更に、ハードウェアターゲットの各々の素子は**素子>>SimCoder**メニューにあります。PSIM標準ライブラリにある多数の素子もコード生成時に使用することができます。

オプション → 設定メニューのタブAdvancedのHardware code generationの「*Show image next to elements that can be used for code generation*」オプションボックスをチェックすると、コード生成に使用できるPSIM標準ライブラリの素子のところにマークが表示されます。

また、各ハードウェアターゲットによって下記のようなマークも付きます。

- コード生成用素子 : **CG**
- TI F2833x,F2803x,F2806x,F2802x専用素子 : **TI**
- PE-Expert4専用素子 : **PE**

2 コード生成方法

2.1 概要

SimCoder を利用してコード生成する場合、次の 4 つのステップを行います。

- ・ 連続系の制御システムによるシステムのシミュレーション
- ・ 離散系の制御システムによるシステムのシミュレーション
- ・ 対象制御ハードウェアがない場合は、制御回路部をサブ回路にしてコード生成
- ・ 対象制御ハードウェアが決まっていれば、ハードウェア素子を追加してコード生成

しかし、最初の 2 つのステップは省略することもできます。例えば、PSIM で回路図を作成して、直接コードを生成することも可能であり、必ずしもシミュレーションを行う必要はありません。

制御システムを離散系で作成しなければ SimCoder でコード生成できないため、デジタルコントロールモジュールが必要です。

降圧チョップパを利用して SimCoder のコード生成方法を以下に具体的に説明します。

2.2 連続系システム

通常は、最初に連続系でシミュレーション回路を作成します。図 2.1 は電流制御付の降圧チョップパ回路です。この回路では、PI コントローラは連続系の s 領域で設計しています。このコントローラの PI ゲイン、 k と時定数 T はそれぞれ $k=0.4$ と $T=0.0004$ です。スイッチング周波数は 20kHz です。

この例の目的は、点線内の制御回路について C コードを自動的に生成することです。コードを生成するために最初に行わなければならないことは連続系 (s 領域) のアナログ PI コントローラを離散系 (z 領域) のデジタル PI コントローラに変換することです。

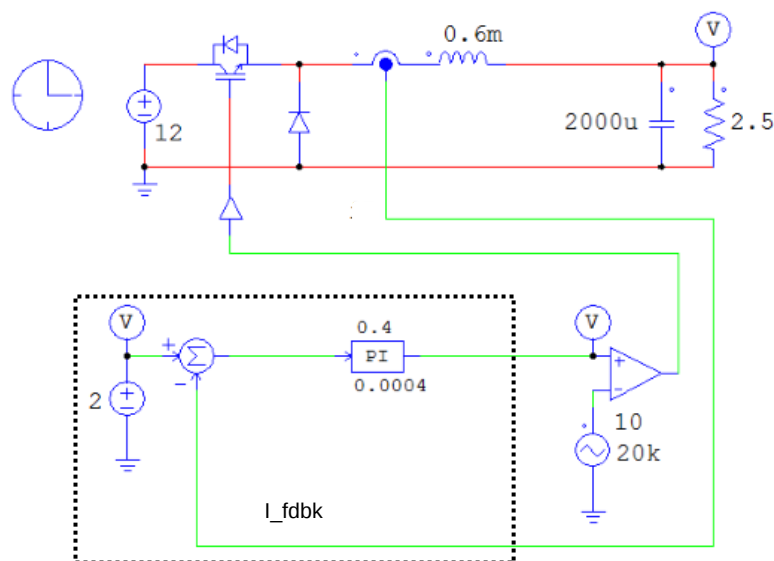


図2.1 降圧チョッパ制御回路（s領域）

2.3 離散系システム

アナログコントローラのパラメータをデジタルコントローラのパラメータに変換するために、デジタルコントロールモジュールのs2z Converterプログラムを使用します。s2z Converterを使用するには、PSIMを起動してユーティリティ -> s2z コンバータ(z)を選択してください。

アナログコントローラをデジタルコントローラに変換するとき、色々な変換法を使用することができます。最も一般的な方法はバイリニア変換（双一次変換 タスティン変換、台形差分法とも呼ばれる）およびバックワードオイラー法（後退オイラー法）です。

この例では、バックワードオイラー法を使用します。サンプリング周波数はスイッチング周波数と同じ20kHzとしてs2z Converter で変換すると、デジタルPIコントローラのパラメータは以下のようになります：

比例部分のゲイン、 $k_1 = 0.4$

積分部分のゲイン、 $k_2 = 1000$

デジタルコントローラを用いた回路を次に示します。

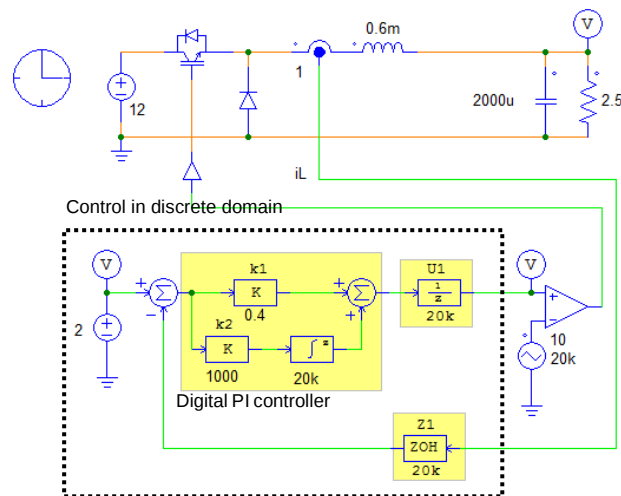


図2.2 降圧チョッパ制御回路（z領域）

図2.1と図2.2を比較すると、z領域の回路では黄色のハイライトで示した3箇所の変更があります。アナログPIコントローラはデジタルPIコントローラに置き換えられます。デジタル積分器の「Algorithm Flag」は1、サンプリング周波数は20kHzに設定します。ゲインk1とk2は上に述べた通りです。

更に、ゼロ次ホールドブロックZ1は、デジタル制御システムではフィードバック電流*i*_LをA/D変換してサンプリングするために使用されます。最後に、単位時間遅れブロックU1は、実際のデジタル制御の1サンプリング遅れをモデル化するため使用されています。

変換されたデジタル制御は安定した制御ループと所望の性能となっているはずですが、もしそうでない場合はアナログコントロールシステムへ戻りアナログコントローラの再設計を繰り返さなければなりません。

バックワードオイラー法を利用した積分器の入出力関係は以下のように表現できます。

$$y(n) = y(n-1) + T_s * u(n)$$

ここで、*y*(*n*)と*u*(*n*)は現在の時間の出力と入力であり、*y*(*n*-1) は1サンプリング前の出力で*T_s*はサンプリング周期です。上の式に基づいて、離散系の積分器を加算器と単位時間遅れブロックで以下の図2.3のように置き換えることができます。

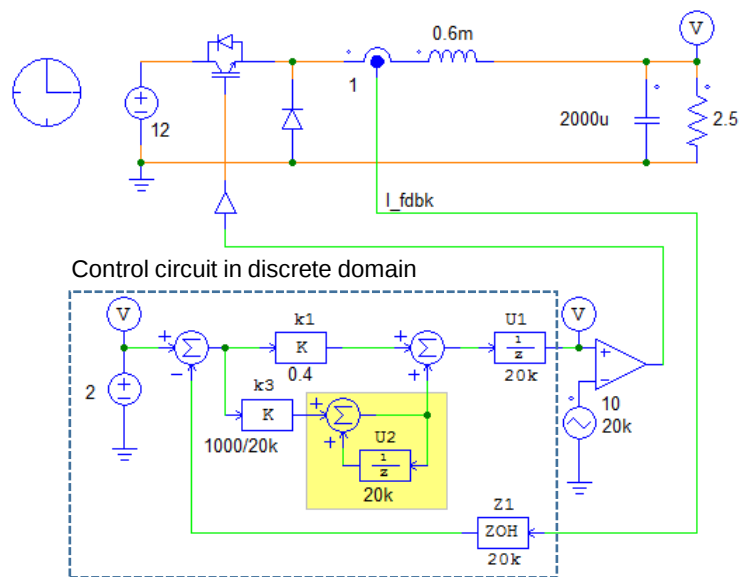


図2.3 バックワードオイラー法に基づいた降圧チョッパ制御回路（ z 領域）

積分ゲイン $k3$ は、 T_s をサンプリング周波数20kHzで割った値とします。この回路の利点は、積分器の積分動作の開始／停止を簡単に制御することができることです。（詳しくは図2.6で説明します。）

2.4 ハードウェアターゲット用のシステムコード生成

SimCoderは対象制御ハードウェア用のコードを生成します。コード生成の前に、適切なハードウェア素子を使用した回路を作成します。また、ハードウェア素子の各設定値は、実際のハードウェアの範囲を考慮して、必要に応じて値をスケーリングする必要があります。

ハードウェア素子付きシステムのシミュレーション回路の作成は、以下のようになります。

- A/D変換器、デジタル入出力等を追加する
- コンパレータを使用したPWM発生回路をハードウェアPWM発生器に置き換える
- 必要に応じてイベントシーケンス制御を追加する

図2.4はF2833x用のハードウェア素子を使用した回路であり、ハードウェア素子は黄色でハイライトされています。

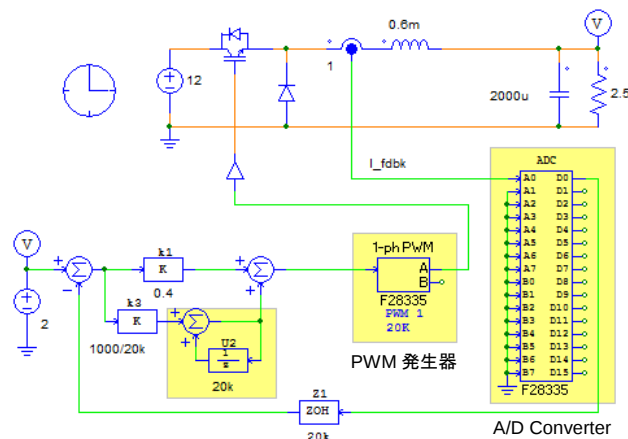


図2.4 ハードウェア素子を使用した降圧チョッパ制御回路

図2.3に対して、図2.4では以下のように変更されています。

A/D変換器を電流センサと制御サブ回路の間に置きます。このA/D変換器の入力範囲に注意する必要があります。電流センサ出力がA/D変換器の入力範囲外である場合、センサ信号をスケーリングしなければなりません。この例の場合は、A/D変換器は次のように設定します。

—ADCモード：start-stop(8-channel) これはADCが片側グループのみの指令に従ってトリガーされた時に一回だけ変換します。

—Ch A0 モード：DC 設定範囲は0から3Vです。

—ChA0 ゲイン：1.0

コンパレータとキャリア波はPWM生成器に置き換えられます。PWM生成器の設定は次のようになります。

—PWMソース：PWM1 これはF28335プロセッサのPWMモジュールを指定しています。

—出力モード：Use PWM A PWM出力ポートを指定しています。

—サンプリング周波数：20k

—キャリア波タイプ：のこぎり波 (start high) 詳細は後の章にあります。

—トリガADC：Trigger ADC Group A

—ADCTriggerPosition:0

—ピーク間電圧：10

ハードウェアPWM発生器には1サンプリング周期遅れを含んでいるので単位時間遅れブロックが削除されています。

「シミュレーション制御」メニューで、SimCoderのタブにはいり

—ハードウェアタイプ：F2833x withRAM Debug

—CPU Version：F28335

—Default Fixed-Point Position：フロートに設定されるので適用外です。

注釈部分にコメントを入力するとCコードの先頭へ挿入されます。編集、追加が可能です。

シミュレーションを実行し2.2の結果と近い結果が得られることを確認してみてください。

結果を確認できたら**シミュレート** -> **コード生成** を選択してCコードを生成することができます。

SimCoderで生成したCコードは次のようになります。

```
/*
*****
// This code is created by SimCoder Version 9.3.3 for TI F2833x Hardware Target
//
// SimCoder is copyright by Powersim Inc., 2009-2013
//
// Date: February 24, 2014 14:36:33
*****
#include<math.h>
#include"PS_bios.h"
typedef float DefaultType;
#define GetCurTime() PS_GetSysTimer()

interrupt void Task();

DefaultType fGbliref = 0;
DefaultType fGblU2 = 0;

interrupt void Task()
{
    DefaultType fU2, fSUMP1, fSUMP3, fk3, fk1, fSUM1, fZ1, fTI_ADC1, fVDC2;

    PS_EnableIntr();
    fU2 = fGblU2;

    fTI_ADC1 = PS_GetDcAdc();
    fVDC2 = 2;
    fZ1 = fTI_ADC1;
    fSUM1 = fVDC2 - fZ1;
    fk1 = fSUM1 * 0.4;
    fk3 = fSUM1 * (1000.0/20000);
    fSUMP3 = fk3 + fU2;
    fSUMP1 = fk1 + fSUMP3;
    PS_SetPwm1RateSH(fSUMP1);
#ifdef DEBUG
    fGbliref = fVDC2;
#endif
    fGblU2 = fSUMP3;
    PS_ExitPwm1General();
}

void Initialize(void)
{
    PS_SysInit(30, 10);
    PS_StartStopPwmClock(0);
    PS_InitTimer(0, 0xffffffff);
    PS_InitPwm(1, 0, 20000*1, (4e-6)*1e6, PWM_POSI_ONLY, 42822); // pwnNo, waveType, frequency, deadtime,
outtype
    PS_SetPwmPeakOffset(1, 10, 0, 1.0/10);
    PS_SetPwmIntrType(1, ePwmIntrAdc0, 1, 0);
    PS_SetPwmVector(1, ePwmIntrAdc0, Task);
    PS_SetPwmTzAct(1, eTZHighImpedance);
    PS_SetPwm1RateSH(0);
    PS_StartPwm(1);
}
```

```
PS_ResetAdcConvSeq();
PS_SetAdcConvSeq(eAdc0Intr, 0, 1.0);
PS_AdcInit(1, 11);

PS_StartStopPwmClock(1);
}

void main()
{
    Initialize();
    PS_EnableIntr(); // Enable Global interrupt INTM
    PS_EnableDbgm();
    for (;;) {
    }
}
```

図2.5 生成されたコードの例

生成されたコードは以下の構造になっています。

Interrupt void Task () : 割り込み実行関数であり、20kHzごとにこの関数が呼び出されます。

void Initialize () : 初期化関数であり、ハードウェアを初期化します。

void main () : メイン関数であり、初期化ルーチンを呼んで無限ループを走らせます。

この例では、すべての制御ブロックは20kHzのサンプリングレートで実行されます。制御システムに他のサンプリングレートがあるときは、もう1つの割り込み実行関数が作成されます。制御システムでサンプリングレートが付いてないブロックがある場合、対応するコードはメイン関数内に生成されます。

コードと必要なプロジェクトファイルはPSIM回路ファイルと同じフォルダのサブフォルダーに保存されます。このプロジェクトファイルをTIの環境にロードして、コードをコンパイルして、それをハードウェアにアップロードすることができます。

2.5 Cコード生成

ハードウェア素子を制御システムに追加後、**シミュレート -> コード生成**を選択してCコードを生成することができます。次に、生成されたプロジェクトファイルをMywayプラスのPE-View環境にロードして、コードをコンパイルし、それをハードウェアにアップロードすることができます。生成されたコードの例を以下に示します。

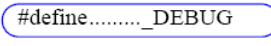
```

/*****
// This code is created by SimCoder Version 1.0 for Myway PE-Expert3
//
// SimCoder is copyright by Powersim Inc., 2008
//
// Date: March 14, 2008 15:10:04
*****/

#include.....<math.h>
#include.....<mwio3.h>
#define....._DEBUG * Comment it out if not watching data from PE_VIEW */
interrupt void Task();

float .....fGbliref = 0.0;
float      fGblUDELAY1 = 0.0;

```



If defined, values of voltage probes can be watched in PE-View.

```

interrupt void Task()    The main interrupt service routine for 20 kHz
{
    float fVDC2, fPEV_ADC1, fPEV_ADC1_1, fPEV_ADC1_2, fPEV_ADC1_3, fZOH3, fSUM1,
    fP2;
    float fSUMP3, fUDELAY1, fP1, fSUMP1, fPEV_PWM_11Div0, fPEV_PWM_11Div1, fPE
    V_PWM_11Div5;
    float fPEV_PWM_11Div2, fPEV_PWM_11Div6, fPEV_PWM_11Div3, fPEV_PWM_11Div7;
    pev_ad_start(0, 0);

    fUDELAY1 = fGblUDELAY1;
    fVDC2 = 2;

#ifdef    _DEBUG
    fGbliref = fVDC2;
#endif

    while (pev_ad_in_st(0, 0));
    pev_ad_in_grp(0, 0, &fPEV_ADC1, &fPEV_ADC1_1, &fPEV_ADC1_2, &fPEV_ADC1_3);

    fZOH3 = fPEV_ADC1;
    fSUM1 = fVDC2 - fZOH3;
    fP2 = fSUM1 * (1000./20000);
    fSUMP3 = fP2 + fUDELAY1;
    fGblUDELAY1 = fSUMP3;
    fP1 = fSUM1 * 0.4;
    fSUMP1 = fP1 + fSUMP3;
    fPEV_PWM_11Div0 = ((0+10)/2.0);
    fPEV_PWM_11Div1 = fSUMP1 - fPEV_PWM_11Div0;
    fPEV_PWM_11Div5 = fPEV_PWM_11Div1 * (2.0/(10));
    fPEV_PWM_11Div2 = 0 - fPEV_PWM_11Div0;
    fPEV_PWM_11Div6 = fPEV_PWM_11Div2 * (2.0/(10));
    fPEV_PWM_11Div3 = 0 - fPEV_PWM_11Div0;
    fPEV_PWM_11Div7 = fPEV_PWM_11Div3 * (2.0/(10));

    pev_inverter_set_uvw(0, 0, fPEV_PWM_11Div5, fPEV_PWM_11Div6, fPEV_PWM_11Div7);

#ifdef    _DEBUG
    watch_data();
#endif
}

```

```
void Initialize(void)    The initialization routine
{
    pev_init(0);
    pev_ad_set_range(0, 0, 5, 1, 1, 1);
    pev_ad_set_range(0, 1, 1, 1, 1, 1);
    pev_inverter_init(0, 0, 20000, (4e-6)*1E9);
    pev_inverter_set_syncint(0, 0.0);
    int5_init_vector(Task);
    int5_enable_int();
    pev_inverter_enable_up_int5(0);
    pev_inverter_set_uvw(0, 0, 0, 0, 0);
    wait(100);
    pev_inverter_stop_pwm(0, 0);
}

void main()             The main program
{
#ifdef    _DEBUG
    watch_init();
#endif

    int_disable();
    Initialize();
    int_enable();
    while (1) {
    }
}
```

図2.5 生成されたコードの例

生成されたコードは以下の構造になっています。

Interrupt void Task () : 割り込み実行関数であり、20kHzごとにこの関数が呼び出されます。

void Initialize () : 初期化関数であり、ハードウェアを初期化します。

void main () : メイン関数であり、初期化ルーチンを呼んで無限ループを走らせます。

この例では、すべての制御ブロックは20kHzのサンプリングレートで実行されます。制御システムに他のサンプリングレートがあるときは、もう1つの割り込み実行関数が作成されます。制御システムでサンプリングレートが付いてないブロックがある場合、対応するコードはメイン関数内に生成されます。

コードと必要なプロジェクトファイルはPSIM回路ファイルと同じフォルダのサブフォルダーに保存されます。このプロジェクトファイルをPE-View環境にロードして、コードをコンパイルして、それハードウェアにアップロードします。

2.6 イベントコントロール付きシステム

制御システムでは、イベントコントロールが必要な場合があります。すなわち、ある条件が満たされた時、システムがある状態から別の状態へ移行する場合があります。SimCoderはPSIMのサブ回路を利用してイベントコントロールを行います。イベントコントロールの詳細については第4章を参照してください。

StopモードとRunモードの2つのモードがあるイベントコントロール付きシステムを次のように作成します。

ーシステムのstart/stopを制御するために手動スイッチを加えます。これにより、システムにはStopモードとRunモードの2つモードができます。この手動スイッチの位置を変化させてあるモードから別のモードに移行します。

ーStopモードでは、積分器は働かないようにし、その出力を0にリセットします。イベントコントロール付きシステムの例を図2.6に示します。図において、S1およびS2はサブ回路であり、このサブ回路S1およびS2の内容を図2.7に示します。

ーこのシステムには、サブ回路S1によって表わされるStopモードと、サブ回路S2によって表わされるRunモードの2つのモードがあります。デフォルトの運転モードはStopモードであり、これは、サブ回路S1のポートEIN1にデフォルトイベント素子を接続することによって定義されます。

ーサブ回路S1には2つの入力イベントポートEIN1とEIN2、1つの出力イベントポートEORun、1つの入力信号ポートRunSwitch、1つの出力信号ポートRunModeがあります。サブ回路S2には1つの入力イベントポートEIN、1つの出力イベントポートEOSTOP、1つの出力信号ポートRunModeがあります。

ーStopモードとRunモードを相互に移行するために条件を定義します。条件はグローバル変数RunSWを使用します(グローバル変数についてセクション5.2を参照してください)。この変数はグローバル変数素子をデジタル入力素子の出力に接続することによって定義されます。

ー押しボタンスイッチSW1のオン／オフ状態はハードウェアデジタル入力素子によって検出され、スイッチがオフの場合、デジタル入力とグローバル変数RunSWは1 (HIGH)になります。システムがRunモードへ移動する。RunSWが0 (LOW)の時は、Stopモードになります。

ーマルチプレクサMUX1は、Stopモードの時に積分回路が積分動作しないように追加されています。つまり、Stopモードの時RunSWは0であり積分回路は動作しません。RunモードでRunSWが1の時、積分回路は積分動作を行います。

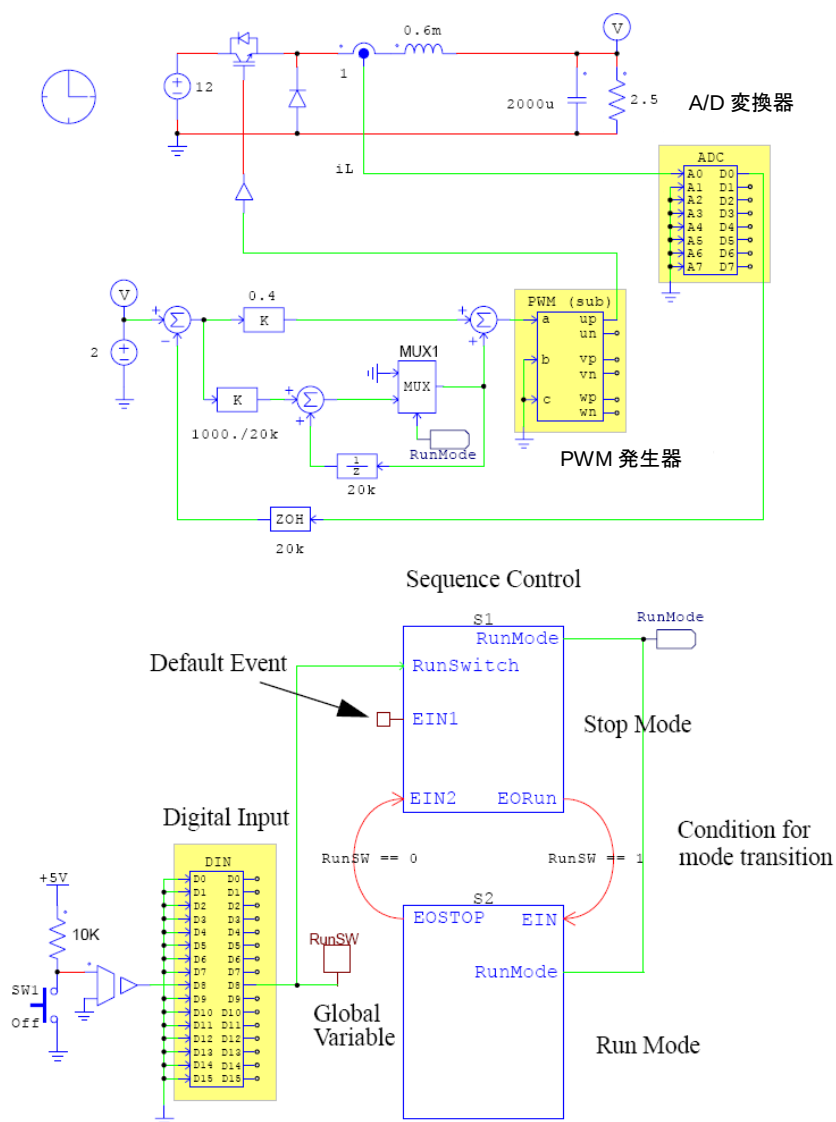


図2.6 シーケンス制御付き降圧チョッパ制御回路

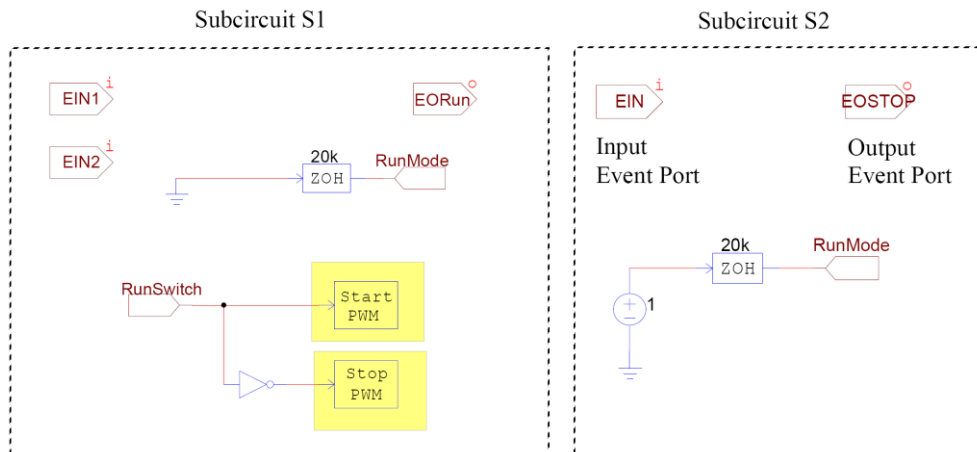


図 2.7 Run モードと Stop モードのサブ回路

システムは以下のように動作します：

- 手動スイッチのオン／オフ状態はハードウェアデジタル入力素子によって検出され、その後、この信号はポート「RunSwitch」を通じてサブ回路S1に送られます。同じ信号がグローバル変数「RunSw」として指定されます。グローバル変数は条件文の中で使用することができます。

- システムは「Stopモード」から開始します。これは、サブ回路S1のポートEIN1にデフォルトイベント素子を接続することで定義しています。条件「RunSw == 1」が成立すると、システムは「Stopモード」から「Runモード」へ移行します。これは、S1の出力イベントポートEORunとS2の入力イベントポートEINの接続によって定義されています。

- 「Runモード」にいるとき、条件「RunSw == 0」が成立すると、システムは「Runモード」から「Stopモード」へ移行します。これはS2の出力イベントポートEOSTOPとS1の入力イベントポートEIN2の接続によって定義されています。

- 「Stopモード」では、RunModeを「0」に設定しています。また、RunSwitchが「0」ならばハードウェアPWM発生器は停止していますが、RunSwitchが「1」に変更されると、PWMを開始し同時に「Stopモード」から「Runモード」へ移行します。

- 「Runモード」では、積分器の動作を有効にするためにRunModeは「1」に設定されています。

3 サブシステムのコード生成

3.1 サブシステム

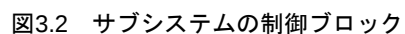
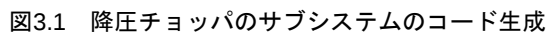
SimCoderでは、サブシステムを含むシステム、またはサブシステム単独でコード生成することができます。しかし、いくつかの制限があるため以下に説明します。

- サブ回路には双方向の入出力信号ポートは使用できません。入力には入力信号ポート、出力には出力信号ポートを使用します。
- ハードウェア入出力素子（A/D変換器やPWM発生器等）およびハードウェア割り込み素子はサブシステムでは使用できません。これらは、トップレベルのメイン回路でのみ使用可能です。
- サブシステムの入力にサンプリングレートがあり、サブシステム内の回路からこのレートを得ることができないとき、入力のサンプリングレートを定義するためにゼロ次ホールドブロックを接続しなければなりません。ゼロ次ホールドブロックが使用されないと、この入力およびこの入力に接続するその後のブロックはサンプリングレートなしで扱われます。

例えば、サブシステムの外でサンプリングレートが定義され、サブシステム内にはサンプリングレートを示すような離散的な素子がない場合、同じサンプリングレートのゼロ次ホールドブロックを入力に接続しなければなりません。

サブシステムの入力にゼロ次ホールドブロックが接続されてない時、SimCoderはサブシステムが接続されているブロックからサンプリングレートを引き出します。しかし、あいまいさを避けるために、入力にゼロ次ホールドブロックを接続してサンプリングレートを定義することをお勧めします。

サブシステムのコードを生成する方法を説明するために、セクション2.5のシステムの電流フィードバックとPIコントローラ部のコードを生成します。図3.1のように、コード生成する部分をサブ回路S3として回路図を作成します。サブ回路S3には2つの入力 i_L 、 $RunMode$ 、そして1つの出力 V_m があります。入力 i_L および $RunMode$ のサンプリングレートは20kHzであり、図3.2のようにサブ回路S3の入力はゼロ次ホールドブロックと接続されています。



3.2 コード生成

サブシステムのコードを生成するためには、サブ回路を右クリックし、Attributesを選びSubcircuit VariablesタブでGenerate Codeをクリックします。

このサブシステムには、1つのサンプリングレートしかありませんので、生成コードには、1つの関数しかありません。変数 $fIn0$ と $fIn1$ はサブ回路の2つの入力 iL とRunModeに対応しており、また、変数 $fOut0$ はサブ回路の出力 Vm に対応しています。全体システムの生成コードと異なって、サブシステムのコードにはmain関数および初期化ルーチンがありません。

```
float      fGblS3_iref = 0.0;
float      fGblS3_UDELAY1 = 0.0;

void TaskS3(float fIn0, float fIn1, float *fOut0)
{
    float fS3_VDC2, fS3_Z3, fS3_SUM1, fS3_P1, fS3_P2, fS3_SUMP3, fS3_Z4, fS3_MUX1;
    float fS3_UDELAY1;

    fS3_UDELAY1 = fGblS3_UDELAY1;
    fS3_VDC2 = 2;
    fS3_Z3 = fIn0;
    fS3_SUM1 = fS3_VDC2 - fS3_Z3;
    fS3_P1 = fS3_SUM1 * 0.4;
    fS3_P2 = fS3_SUM1 * (1000./20000);
    fS3_SUMP3 = fS3_P2 + fS3_UDELAY1;
    fS3_Z4 = fIn1;
    fS3_MUX1 = (fS3_Z4 == 0) ? 0 : fS3_SUMP3;
    *fOut0 = fS3_P1 + fS3_MUX1;

#ifdef   _DEBUG
    fGblS3_iref = fS3_VDC2;
#endif

    fGblS3_UDELAY1 = fS3_MUX1;
}
```

4 イベントコントロール

4.1 基本概念

イベントは、システムがある状態から別の状態へ移行することを記述するために使用します。以下の図4.1はシステムの状態とそれらの間の状態遷移を示しています。

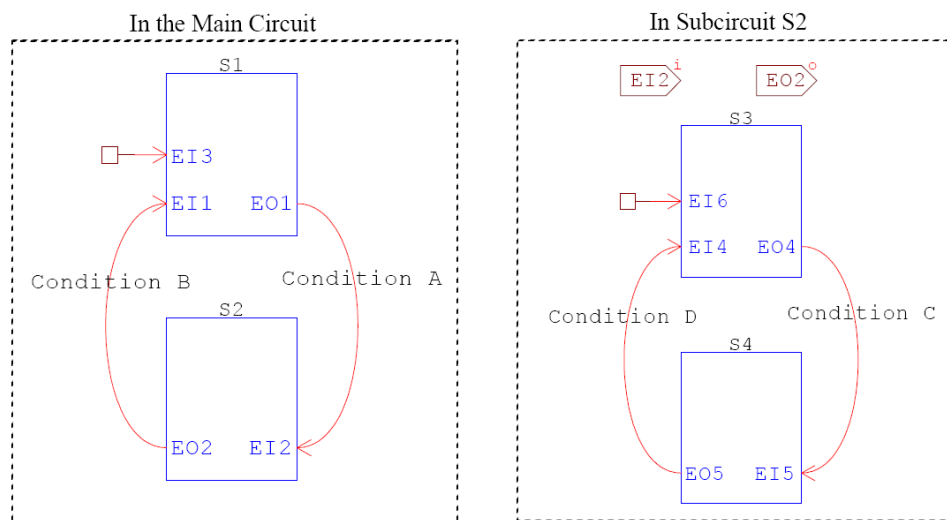


図4.1 状態遷移図

メイン回路では、2つの状態(S1およびS2)があり、状態はサブ回路で実現されています。S1には2つの入力イベントポートEI1とEI3、1つの出力イベントポートEO1があります。S2には1つの入力ポートEI2および1つの出力イベントポートEO2があります。最初は、S1がデフォルト状態になっています。これは入力イベントポートEI3にデフォルトイベント素子を接続することで定義されます。

S1の出力イベントポートEO1はS2の入力イベントポートEI2に接続されています。条件Aが成立すると、システムはS1からS2に移行します。同様に、S2の出力イベントポートEO2はS1の入力イベントポートEI1に接続されています。この場合、条件Bが成立すると、システムはS2からS1に移行します。

2つ以上の状態は共存できず、1つの状態であればいつでも存在できます。このS1とS2のようなケースを排他的と称しています。

右側のシステムはサブ回路S2の内容を示しています。サブ回路S2には2つの状態、S3とS4があります。システムはサブ回路S2へ移動すると、デフォルトでS3になります。条件Cが成立すると、S3からS4に移行します。条件Dが成立すると、S4に移行します。このようにシステムが持つことができる状態の数に制限はありません。

4.2 イベントコントロール素子

イベントと状態遷移を制御するためには次の素子を使用します。

- 入力イベントポート
- 出力イベントポート
- デフォルトイベント素子
- 初回エントリーフラグ素子
- ハードウェア割り込み素子（セクション5.4を参照）
- グローバル変数

サブ回路に入力イベントポートを追加すると、このサブ回路への移行できるポートが作成されます。同様に、サブ回路に出力イベントポートを追加すると、このサブ回路から他のサブ回路へ移行できるポートが作成されます。入力イベントポート(EI1)と出力イベントポート(EO1)を追加した場合のサブ回路は図4.2のようになります。

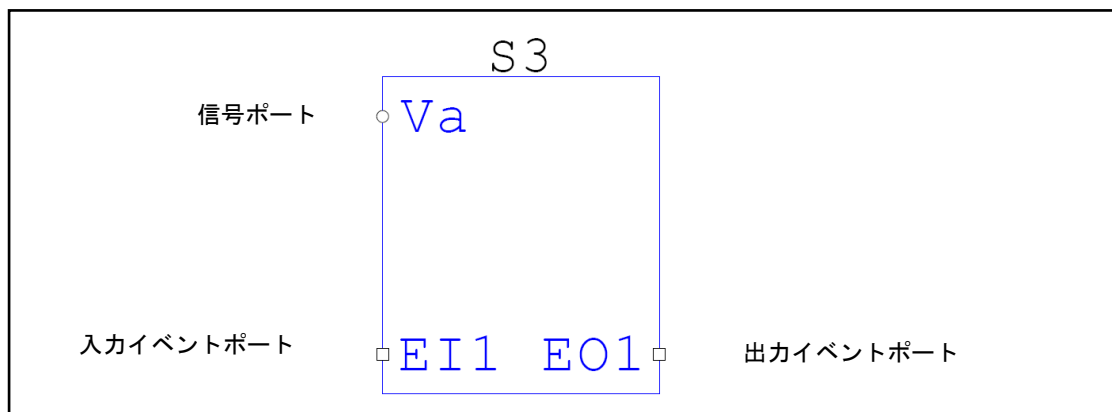


図4.2 入出力イベントポート付きサブ回路

図のようにイベントポートは正方形、信号ポートは円形のノードシンボルで表示されます。

Editメニューの「Event Connection」機能を選択して、入力イベントポートには出力イベントポートかハードウェア割り込み素子を接続することができます。出力イベントポートには入力イベントポートを接続することができます。入出力イベントポートやハードウェア割り込み素子は他のタイプのノードに接続することはできません。

出力イベントポートに関しては、移行条件を定義しなければなりません。例としてサブ回路S3の出力イベントポートEO1のプロパティウインドウを図4.3に示します。

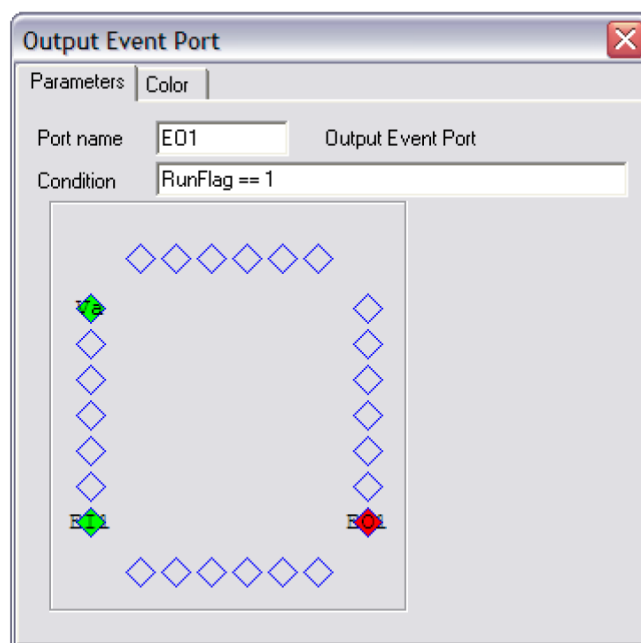


図4.3 出力イベントポート

移行条件「RunFlag==1」が成立すれば、他のサブ回路へ移行します。条件式は以下の例のようにCコードで表記します。

```
(RunFlag == 1) && (FlagA >= 250.) || (FlagB < Vconst)
```

条件式にはグローバル変数、数値、パラメータファイルで定義する定数、またはメイン回路からサブ回路に送られた変数を使用することができます。

グローバル変数を作成するには、グローバル変数素子をノードに接続してください。

4.3 イベント付きサブ回路の制限

入力または出力イベントポートを含むサブ回路はイベント付きサブ回路であり、イベント付きサブ回路の中のすべてがイベントの特性を引き継ぎます。すなわち、サブ回路Bの中にサブ回路Aがあって、サブ回路Bがイベント付きである場合、サブ回路Aが入力/出力イベントポートもたない場合でも、サブ回路Aはイベント付きサブ回路として考えなければなりません。

PSIMIには、3種類のサブ回路があります。

- **標準サブ回路**：標準サブ回路はイベントポートを含んでおらず、従来のサブ回路と同じです。
- **イベント付きサブ回路**：イベント付きサブ回路には入出力イベントポートがあり、ハードウェア割り込み素子は接続されていません。

- **ハードウェア割り込み付きサブ回路**：ハードウェア割り込み付きサブ回路は、出力イベントポートを含まず入力イベントポートだけを含んでおり、入力イベントポートにはハードウェア割り込み素子が接続されます。このタイプのサブ回路はハードウェア割り込みを扱うためだけに使用されます。

イベント付きサブ回路およびハードウェア割り込み付きサブ回路にはコード生成が可能となるように次のような制限があります。

- イベント付きサブ回路およびハードウェア割り込み付きサブ回路は、コード生成に使用可能な素子のみを含めることができます。つまり、このタイプのサブ回路はコード生成に使用できない抵抗やrmsブロックを含むことができません。
- ハードウェア割り込み付きサブ回路は、複数の入力イベントポートを持つことができますが出力イベントポートを持つことができません。更に、入力イベントポートにはハードウェア割り込み素子のみ接続可能です。入出力信号ポートにはハードウェア素子のみ接続することができ、他の機能ブロックに接続することができません。図4.4にハードウェア割り込みのあるサブ回路の接続方法を示します。

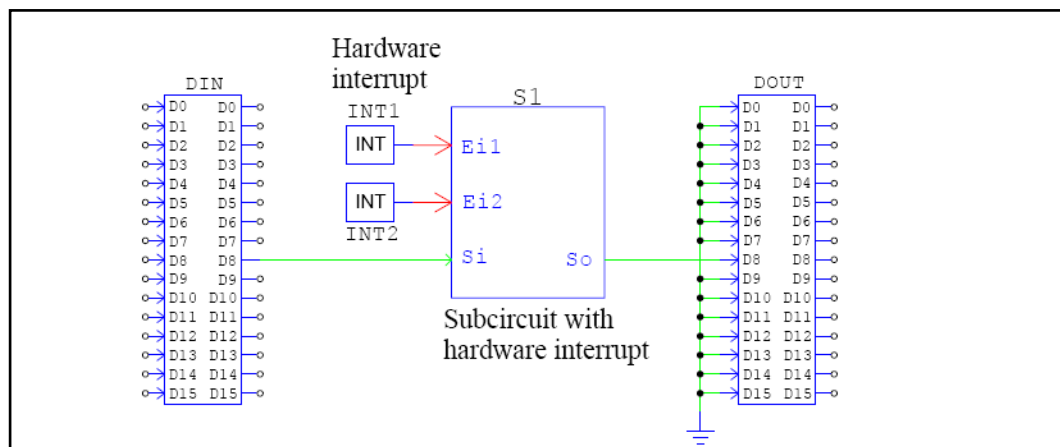


図4.4 ハードウェア割り込み付きサブ回路

ここで、S1はハードウェア割り込み付きサブ回路です。このサブ回路には2つハードウェア割り込み素子INT1とINT2が接続されています。この回路の入力信号ポートSiはハードウェアデジタル入力として出力信号ポートSoはハードウェアデジタル出力に接続されています。

- ハードウェア割り込み付きサブ回路がサンプリングレートの定義された離散系のブロックを含んでいると、そのサンプリングレートは無視されハードウェア割り込みが発生したときだけサブ回路が呼ばれます。例えば、サブ回路が離散積分器を含んでいると、離散積分器のサンプリングレートは無視されます。離散積分器の計算における前の時間は前回の割り込みが発生した時間になります。

- 2つのサブ回路の出力信号を接続する場合、直接接続しなければなりません。他のブロックの経由で接続することができません。これについては以下の図で説明します。

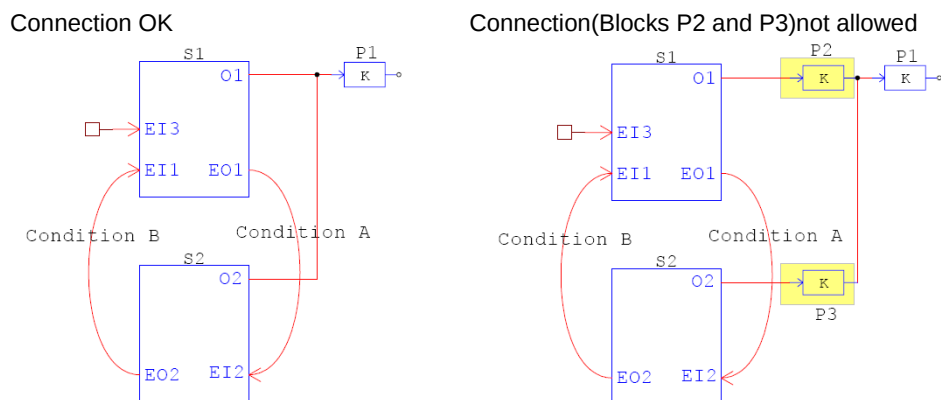


図4.5 2つのサブ回路の接続方法

左側の回路では、サブ回路S1とS2の両方がそれぞれ出力信号ポートO1およびO2を持っています。サブ回路S1およびS2の出力信号ポートはともに比例制御器ブロックP1の入力に接続されています。この接続で、P1ブロックの入力はポートO1またはO2のどちらかから信号が来ることになります。この接続は許可されます。

しかし、右側の回路では、ポートO1はブロックP2に接続され、ポートO2はブロックP3に接続され、そして、次にP2とP3の出力がP1の入力に接続されています。このような接続は許可されません。この場合、ブロックP2はサブ回路S1の中に、そして、ブロックP3はサブ回路S2の中に移動する必要があります。

5 SimCoder ライブラリ

SimCoderは、MywayプラスのPE-Expert4ハードウェア開発プラットフォームをサポートしています。更に、SimCoderを利用して、汎用ハードウェアプラットフォームのためにCコードを生成することができます。ユーザは、生成された汎用的なCコードを修正してそれぞれのハードウェア（DSP・ μ C）で使用することができます。

5.1 標準 PSIM ライブラリの素子

PSIM標準ライブラリの素子の中では、以下の素子がコード生成に使用できます。

これらはオプション>>設定>>Advancedのタブにある “*Show image next to elements that can be used for code generation*” をチェックするとコード生成できる素子に **Cg** がつきます。SimCoderとSPICE用素子には **C/P** がつきます。

数式関数ブロック、シンプルCブロック、変数t（時間）、デルタ（タイムステップ）はコード生成のためのSimCoderでは使用できません。

パラメータファイル素子とのこぎり波生成はSimCoderの特殊な使用方法はこの後の章で説明します。

素子→制御ライブラリ:

Proportional block（比例制御器）

Comparator（比較器）

Limiter（リミッタ）

Upper Limiter（上限リミッタ）

Lower Limiter（下限リミッタ）

Range Limiter（レンジリミッタ）

Summer (+/-)（加算器）

Summer (+/+)（加算器）

Summer (3-input)（加算器-3入力）

素子→制御ライブラリ→計算ブロック:

Multiplier（乗算器）

Divider（除算器）

Square-root（平方根ブロック）

Sine

Sine (in rad.)

Cosine

Cosine (in rad.)

Tangent

Arctangent

Arctangent (単位値で入力)

Arctangent 2(in rad.)

Arctangent 2(単位値で入力)Exponential (a^x) (指数ブロック)

Power (x^a) (累乗)

LOG (base e) (自然対数ブロック)

LOG10 (base 10) (常用対数ブロック)

Absolute Value (絶対値ブロック)

Sign Block (符号関数ブロック)

素子→制御ライブラリ→その他機能ブロック:

Multiplexer (2-input) (マルチプレクサ2入力)

Multiplexer (3-input 2-control) (マルチプレクサ3入力2制御)

Multiplexer (4-input) (マルチプレクサ4入力)

Multiplexer (4-input 1-control) (マルチプレクサ4入力1制御)

Multiplexer (4-input 3-control) (マルチプレクサ4入力3制御)

Multiplexer (8-input) (マルチプレクサ8入力)

Multiplexer (8-input 1-control) (マルチプレクサ8入力1制御)

空間ベクトル PWM

空間ベクトルPWM(α/β)

素子→制御ライブラリ→論理素子:

AND Gate (ANDゲート)

AND Gate (3-input) (ANDゲート-3入力)

OR Gate (ORゲート)

OR Gate (3-input) (ORゲート-3入力)

XOR Gate (XORゲート)

NOT Gate (NOTゲート)

NAND Gate (NANDゲート)

NOR Gate (NORゲート)

素子→制御ライブラリ→デジタル制御モジュール:

Zero-Order Hold (ゼロ次ホールド)
Unit Delay (単位時間遅れ)
Integrator (積分器)
Integrator(with limiter) (積分器 (制限器付き))
Differentiator (離散型微分器)
PI(with limiter) PI(制限器付き)
External Resettable Integrator (外部リセット付き積分器)
Internal Resettable Integrator (内部リセット付き積分器)
PID(with reset) PID(リセット付き)
1st-order Low-pass Filter (一次ローパスフィルタ)
2nd-order Low-pass Filter (二次ローパスフィルタ)
FIR Filter (FIRフィルタ)
FIR Filter (file) (FIRフィルタ (1))
Digital Filter (デジタルフィルタ)
z-domain Transfer Function (z 領域伝達関数)
Circular Buffer(single-output) 循環バッファ(単出力)
FIR Filter(file) FIRフィルタ (ファイル)
Digital Filter(file) デジタルフィルタ (ファイル)

素子→その他:

Parameter File (パラメータファイル)

素子→その他→その他関数ブロックメニュー:

abc-dqo Transformation (abc-dqo変換)
dqo-abc Transformation (dqo-abc変換)
Clarke Transformation (abc-alpha/beta) (クラーク変換 (abc-alpha/beta))
Clarke Transformation(ab-alpha/beta) (クラーク変換 (ab-alpha/beta))
Clarke Transformation(ac-alpha/beta) (クラーク変換 (ac-alpha/beta))
Inverse Clarke Transformation (alpha/beta-abc) (逆クラーク変換 (alpha/beta-abc))
Park Transformation(alpha/beta-dq) (パーク変換 (alpha/beta-dq))
Park Transformation(alpha/beta/sin/cos-dq) (パーク変換 (alpha/beta/sin/cos-dq))
Inverse Park Transformation(dq-alpha/beta) (逆パーク変換 (dq-alpha/beta))
Inverse Park Transformation(dq/sin/cos-alpha/beta) (逆パーク変換 (dq/sin/cos-alpha/beta))
x/y-r/angle Transformation(x/y-r/angle 変換)
r/angle-x/y Transformation(r/angle-x/y 変換)Lookup Table (ルックアップテーブル)
2-D Lookup Table (integer) (2次元ルックアップテーブル(整数))

2-D Lookup Table (interpolation) (2次元ルックアップテーブル (浮動小数点))

Math Function (数式関数)

Math Function (2-input) (数式関数-2入力)

Math Function (3-input) (数式関数-3入力)

Math Function (5-input) (数式関数-5入力)

Math Function (10-input) (数式関数-10入力)

Simplified C Block (シンプルCブロック)

素子→電源メニュー:

Constant (定数素子)

Ground (グラウンド 接地 G)

Ground (1) (グラウンド 接地(1))

Ground (2) (グラウンド 接地(2))

素子→電源→電圧源メニュー:

DC (直流電圧源)

DC (battery) (直流電圧源(電池))

Sine (正弦波電圧源)

3-ph Sine (三相正弦波電圧源)

Triangular (三角波電圧源)

Sawtooth (のこぎり波)

Square (方形波電圧源)

Step (ステップ電圧源)

Step(2-level) (ステップ電圧源(2レベル))

Grounded DC (circire) (接地電圧源)

Grounded DC (1) (接地電圧源)

Math Function (数式関数)

5.1.1 パラメータファイルブロックを用いたグローバルパラメータ設定

パラメータファイルブロックはSimCoder内でも同様に使用することができます。また、SimCoder内ではグローバルパラメータとして使用することもできます。

コードの可読性や修正の簡易性を高めるためにコード内に直接の値を書き込むのではなく、定数名を用いることがあります。例えば、制御器のゲインを1.23とした場合、コード内に1.23と直接書き込むのではなく、Kp=1.23と一度定義したのち、コード内ではKpを用います。

この場合、パラメータはコード内の複数の場所で用いられ、グローバルな定数になります。パラメータをグローバルパラメータとして使いたい場合は、パラメータファイルブロックの中で、パラメータ名の前に「(global)」という文字を入れてください。

下記にその例を示しています。比例制御器のゲインの値をパラメータKpとして、パラメータファイル内ではKpを「(global)Kp=0.4」と記述しています。

対象DSPが浮動小数点型F28335の場合、パラメータファイルにKpを下記のように定義します。

(global) Kp = 0.4

対象DSPが固定小数点型F28335の場合、またデータタイプがIQ-24でパラメータファイルにKpを下記のように定義します。

(global_lq24) = 0.4

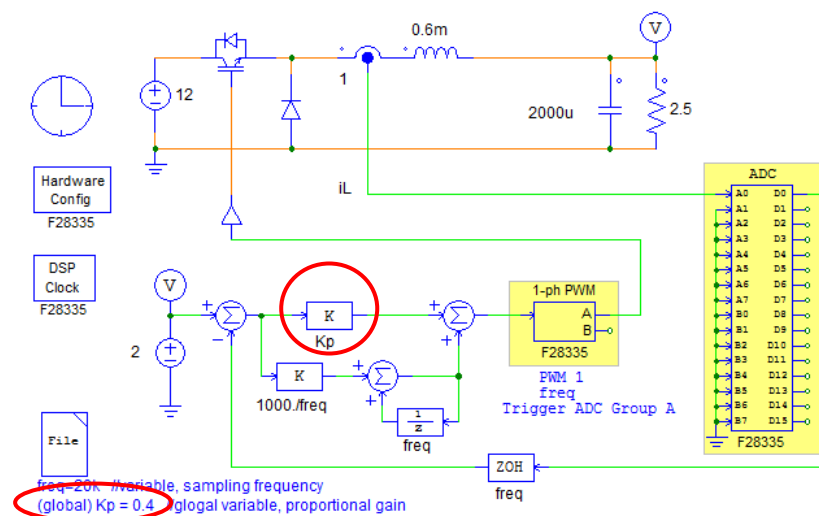


図5.1 降圧回路例1

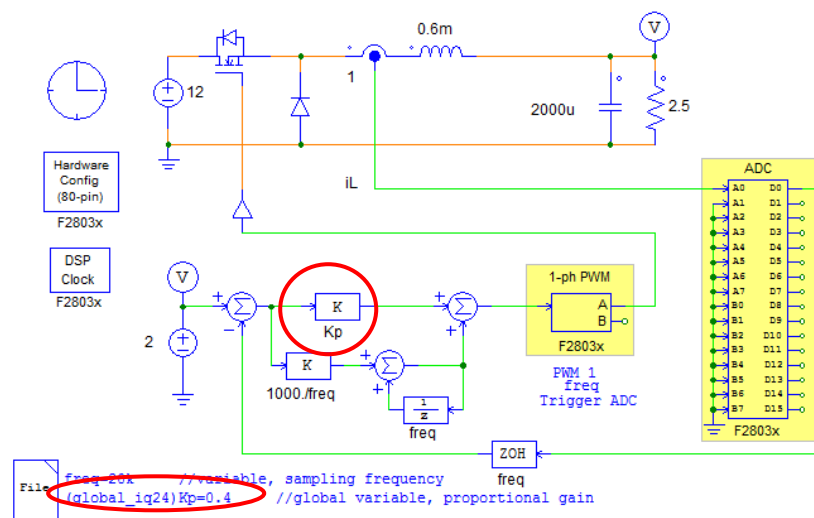


図5.2 降圧回路例2

下記に生成されるコードを示しています。パラメータKpはコードの最初に0.4と定義され、以降のコード内ではKpが使われています。

```

/*****
// This code is created by SimCoder Version 9.1 for TI F28335 Hardware Target
//
// SimCoder is copyright by Powersim Inc., 2009-2011
//
// Date: November 08, 2011 11:26:37
*****/
#include <math.h>
#include "PS_bios.h"
typedef float DefaultType;
#define GetCurTime() PS_GetSysTimer()

interrupt void Task();

DefaultType fGbliref = 0.0;
DefaultType fGblUDELAY1 = 0;

DefaultType Kp = 0.4; // The global parameter Kp is defined here.
interrupt void Task()
{
    DefaultType fVDC2, fTI_ADC1, fZOH3, fSUM1, fP2, fSUMP3, fUDELAY1, fP1, fSUMP1;
    PS_EnableIntr();
    fUDELAY1 = fGblUDELAY1;

    fTI_ADC1 = PS_GetDcAdc(0);
    fVDC2 = 2;
#ifdef _DEBUG
    fGbliref = fVDC2;
#endif
    fZOH3 = fTI_ADC1;
    fSUM1 = fVDC2 - fZOH3;
    fP2 = fSUM1 * (1000./20000);
    fSUMP3 = fP2 + fUDELAY1;
    fGblUDELAY1 = fSUMP3;
    fP1 = fSUM1 * Kp; // The parameter Kp is used here.
    fSUMP1 = fP1 + fSUMP3;
    PS_SetPwm1RateSH(fSUMP1);
    PS_ExitPwm1General();
}
...

```

図5.3 生成されたコード

5.1.2 のこぎり波生成

のこぎり波はDSP内で、システムのタイマーとしてや、また他の周期波形（正弦波など）を生成するために使われます。のこぎり波電圧源（素子 → 電源 → 電圧源の中より）はDSPのカウンタを使って実現されます。

例えば、PE-Expert4では、PEVボードの32ビットフリーランカウンタ（20ns毎カウン）を使ってのこぎり波を生成します。

ハードウェアでは、対象とするDSP内に32ビットフリーランカウンタ（20ns毎カウン）が存在するとしてのこぎり波を生成します。

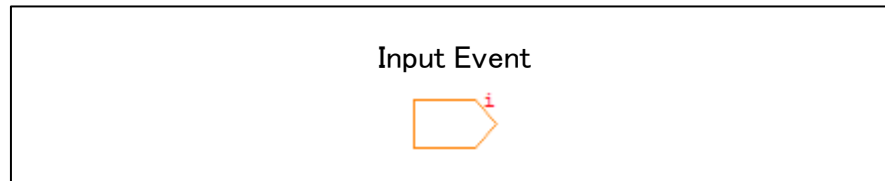
5.2 イベントコントロール素子

イベントコントロールを実現するために使用される素子を以下に説明します。

5.2.1 入力イベント

入力イベント素子のシンボルを以下に示します。

シンボル



シンボルの右上隅の文字「i」は「入力」を表しています。入力イベント素子はサブ回路インターフェースポートの1つであり、サブ回路の中でのみ使用可能です。この素子をダブルクリックすると以下に示すようにポート名と位置を定義できます。

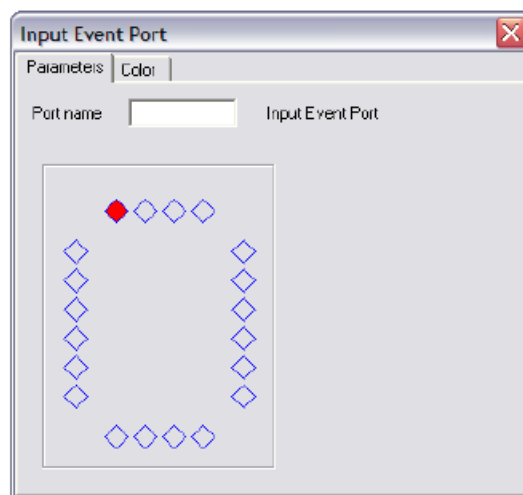


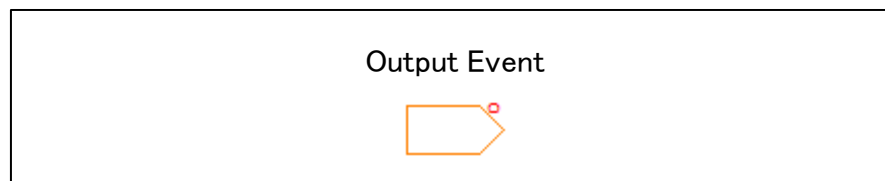
図5.4入力イベントポート

入力イベントポートがあるサブ回路を呼び出すメイン回路において、この入力イベントポートにイベント接続ワイヤを接続すると、イベント接続の条件が成立した場合にこのポートを通してこのサブ回路へシステムの状態が移行します。

5.2.2 出力イベント

出力イベント素子のシンボルを以下に示します。

シンボル



シンボルの右上隅の文字「o」は「出力」を表しています。出力イベント素子はサブ回路インターフェースポートの1つであり、サブ回路の中でのみ使用可能です。この素子をダブルクリックすると以下に示すようにポート名と位置と移行条件を定義できます。

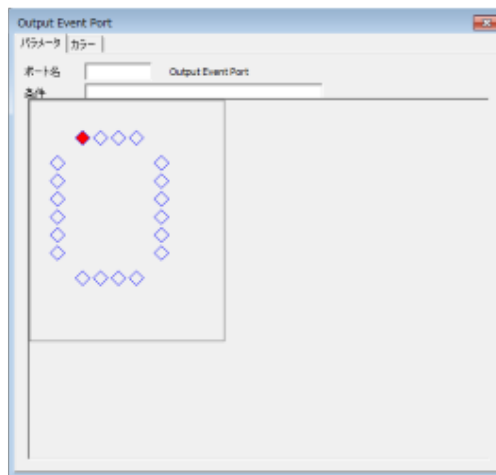


図5.5 出力イベントポート

出力イベントポートで定義された移行条件が成立すると、システムはこのサブ回路から他のサブ回路へ移行します。条件式は以下の例のようにCコードで表記します。

`A == 1`

`A >= B`

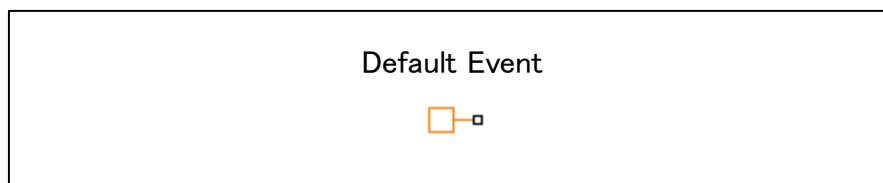
`(A > B) && (C > D)`

ここで、A、B、C、Dはグローバル変数か定数です。

5.2.3 デフォルトイベント

デフォルトイベント素子のシンボルを以下に示します。

シンボル



デフォルトイベント素子は、どのサブ回路がデフォルト状態であるかを定義するのに使用され、サブ回路の入力イベントポートに接続されます。

5.2.4 イベント接続

イベント接続ワイヤは入力イベントポートと出力イベントポートとを接続するツールです。

イベント接続ワイヤはイベントを接続するときのみ使用することができます。

メニューの 素子>>イベント制御>>イベント接続にあります。

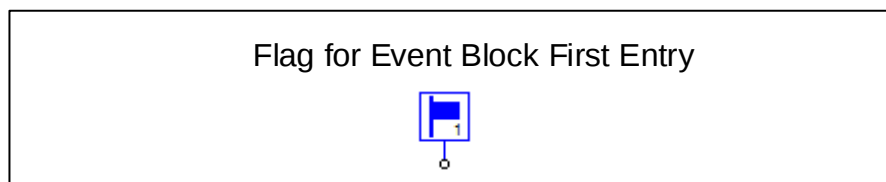
イベント接続ワイヤをダブルクリックして、出力イベントポートの条件式を修正することができます。

イベント接続ワイヤでは、開始点と終点以外にも2つのポイントがあり、これらの2つのポイントを移動することによって接続ワイヤの形を変えることができます。これらの2つのポイントを変更するには、イベント接続ワイヤをハイライトし、右クリックして、「Modify handle 1」あるいは「Modify handle 2」を選択してください。

5.2.5 イベントブロック初回実行フラグ

プログラムの組み方によっては、イベント用のサブ回路に入った初回だけ区別して命令を実行することがあります。イベントブロック初回実行フラグ（Flag for Event Block First Entry）を使うと、イベント用サブ回路に入ったのが初めてなのか、そうでないのかを区別することができます。

シンボル：



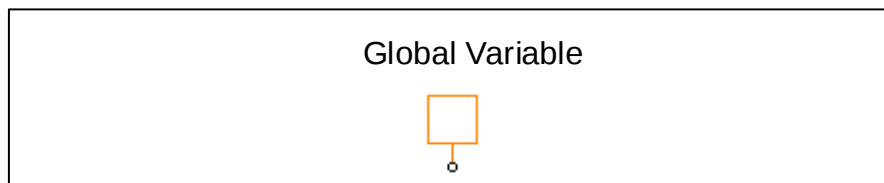
仕様:

パラメータ	機能
イベント・サブ回路ブロック名	グローバル変数名

パラメータの「Event Subcircuit Block Name」にサブ回路名を設定します。（イベントサブ回路ブロックS1を初回実行したかどうかの区別をしたい場合は、パラメータ「Event Subcircuit Block Name」をS1にしてください。）サブ回路内を初回実行している間、本ブロックの出力ノードの値が1になります。その他の場合は0になります。

5.3 グローバル変数

シンボル



仕様

パラメータ	機能
名前	グローバル変数名
初期値	グローバル変数の初期値

信号をグローバル変数として定義するには、グローバル変数素子をノードに接続してください。コード生成のための制御回路の信号のみグローバル変数として定義することができます。

グローバル変数はグローバルにアクセスすることができ、サブ回路を含む回路内のすべてのグローバル変数の初期値を変化させることができます。

グローバル変数は、信号シンクか信号ソースとして使用することができます。信号シンクとして使用するときはノードから信号値を読みこみ、信号ソースとして使用するときはノードの値を設定します。

条件式の変数はグローバル変数として定義しなければなりません。イベント条件式でグローバル変数を使用する例を以下に示します。

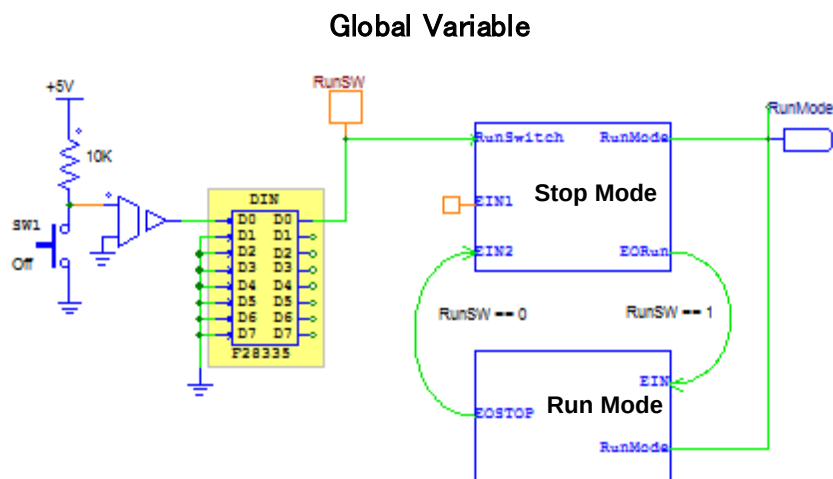


図5.6 グローバル変数の定義方法

この例では、グローバル変数RunSWはデジタル入力素子の出力ピンD0に接続されています。この

グローバル変数はStopモードとRunモードとの移行条件に使用されます。

グローバル変数は信号ソースとしても使用することができ、別のブロックに値を渡すことができます。

グローバル変数はあるノードから別のノードに値を渡すためのラベルとしては使用できません。

グローバル変数の使用には以下の制限があります：

- 同じ信号伝達経路でのみ同じ名前のグローバル変数を複数使用できます。
- 同じ信号伝達経路ではない場合、同じ名前のグローバル変数は同時に動作しないサブ回路内でのみ複数使用できます。

グローバル変数の接続方法を図5.7に説明します。

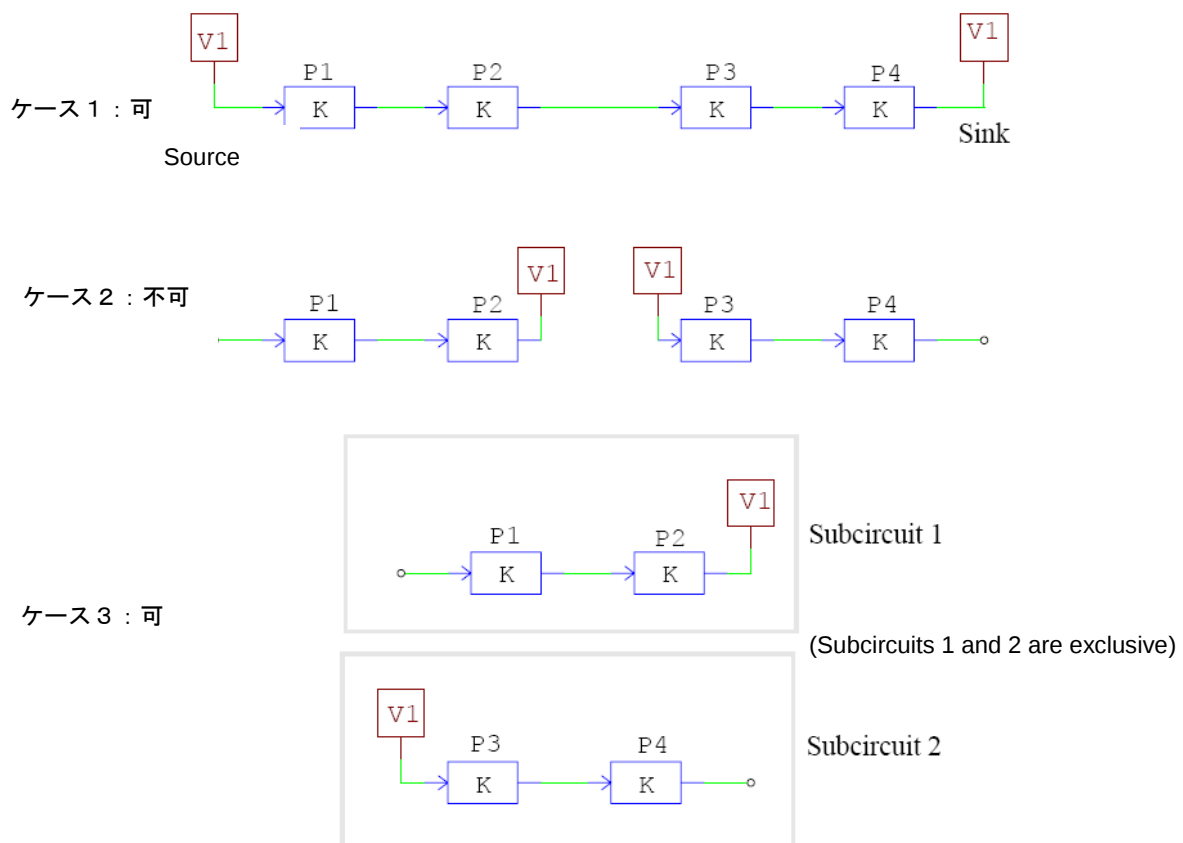


図5.7 グローバル変数の接続方法

ケース1では、グローバル変数V1は、ソースとして最初に使用されブロックP1の入力に値を割り当てます。一連の計算の後、ブロックP4の出力は同じグローバル変数V1に割り当てられます。両方の

グローバル変数が同じ信号伝達経路にあるので、この使用方法是正しいです。

ケース2では、ブロックP2の出力からの値をブロックP3の入力へ渡すために、グローバル変数V1はラベルとして使用していますが、これは許可されずエラーとなります。値をあるノードから別のノードへ渡すためには、ラベル()を使用するかワイヤでノードを接続してください。

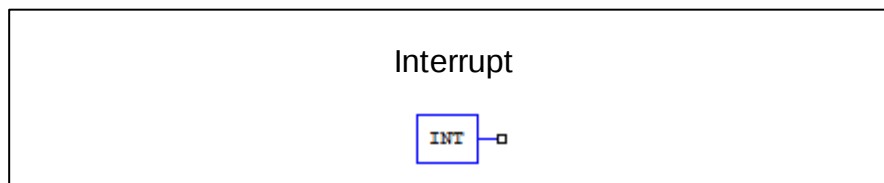
ケース3では、グローバル変数V1はサブ回路1およびサブ回路2で使用されます。システムでは、サブ回路1とサブ回路2は同時に実行しないので、このようなグローバル変数の使用は可能です。

5.4 ハードウェア割り込み

DSPIは、デジタル入力、エンコーダ、キャプチャ、PWM生成器(F2833xとF2803x)などからハードウェア割り込みを発生させることができます。ハードウェア割り込みブロックは素子から発生した割り込みと対応させたい割り込み関数であるサブ回路を繋げる役割を持っています。

ハードウェア割り込み要素はサブ回路に置くことはできず、トップレベルのメイン回路にしか置くことはできません。

シンボル:



仕様:

パラメータ	機能
名前	割り込みを発生させる素子名の設定。
チャンネル番号 Channel No	割り込みを発生させる素子のチャンネル番号の設定。 例えばデジタル入力のチャンネルD0が割り込みを発生するとチャンネル番号は0に設定されなければなりません。 このパラメータは - デジタル入力、 - キャプチャ(PE-Expert4ターゲットのみ) だけで使用されます。エンコーダやPWM生成器では使えません。
エッジ検出タイプ	割り込み信号の発生エッジタイミングの設定。 デジタル入力およびキャプチャ割り込みのときに有効。 - No edge detection : 割り込みは発生しません。 - Rising edge : 入力信号の立ち上がりで割り込みが発生します。 - Falling edge : 入力信号の立ち下がりで割り込みが発生します。 - Rising/falling edges : 入力信号の立ち上がりおよび立ち下がりの両方で割り込みが発生します。。

割り込み素子の使用方法を次の図に示します。

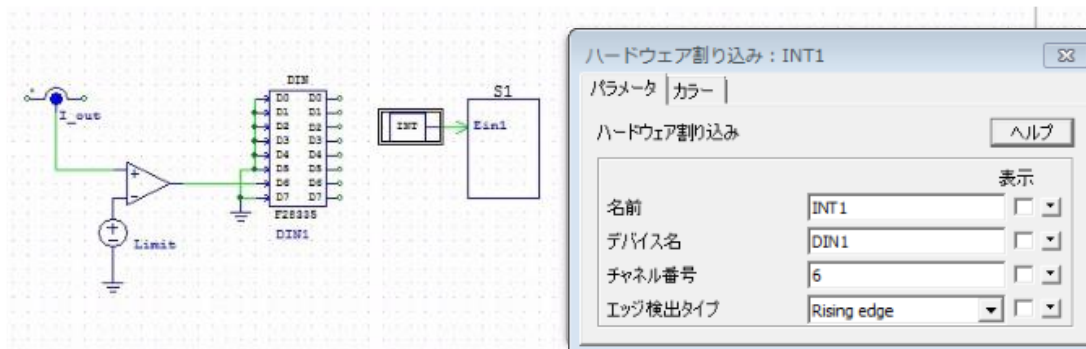


図5.8 割り込み素子の使用方法

この回路では、検出電流 i_L とリミット値 i_{L_limit} を比較して、検出電流 i_L がこのリミット値を超えると、パルス信号が発生してそれをデジタル入力素子DIN1の入力D6に送ります。この回路では、割り込み素子の「Device Name」は「DIN1」として、「Edge Detection Type」は「Rising edge」として定義します。これにより、入力信号の立ち上がりがハードウェア割り込みを発生させ、サブ回路S1のイベント入力ポートEint1から割り込み関数S1へ移行します。

5.5 SimCoder C Block

外部ヘッダファイルとCソースファイルを受け入れることができ、シミュレーションコードとハードウェアターゲットコードを簡単に組み合わせることができるCブロックです。

SimCoder Cブロックは、PSIM回路でハードウェアコード生成用に使用できるという点で、シンプルCブロックに似ています。但し、シンプルCブロックとは異なり、外部ヘッダファイルとCソースファイルを簡単にインクルードし、初期化を実行し、事前定義されたマクロを定義して呼び出すために、SimCoder Cブロックに関数が用意されています。さらに、“Flag_Simulation”と呼ばれるフラグが用意されているため、ユーザーはCブロック内にシミュレーションコードとハードウェアコードの両方を持つことができ、フラグを使用して2つを区別することができます。下記はSimCoder Cブロックのイメージとなります。

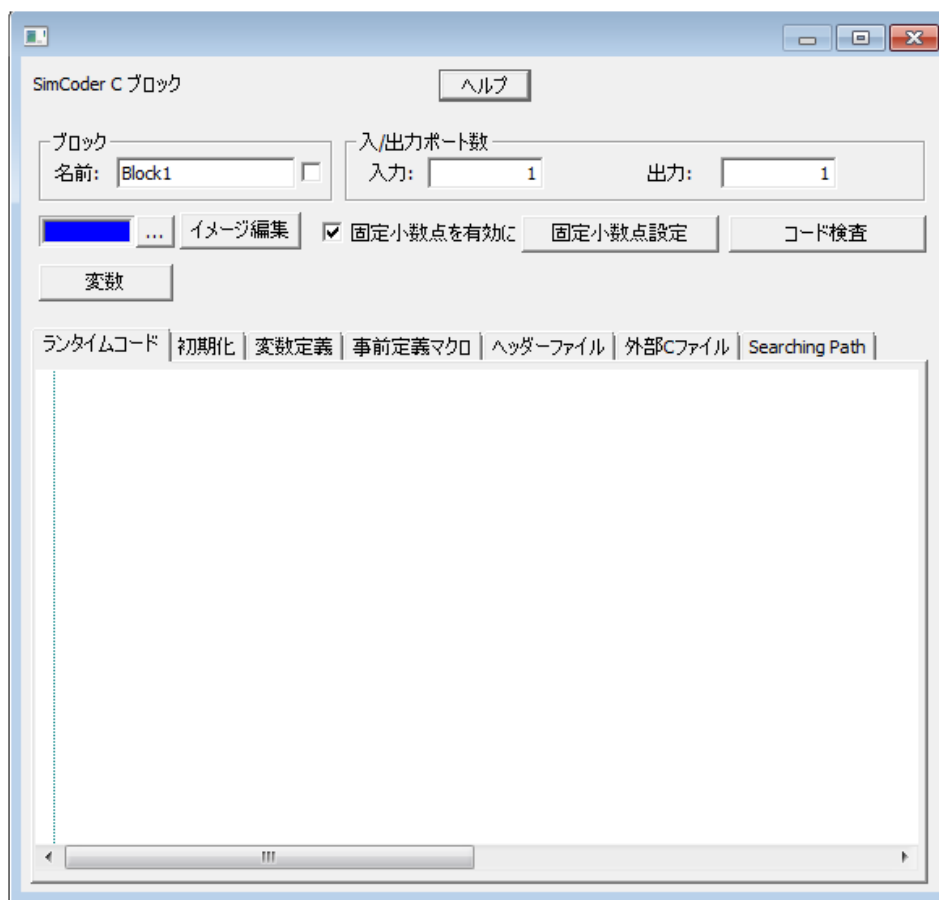


図5.9 SimCoder Cブロック

タブの内容は次のようになります。

タブメニュー	機能
ランタイムコード	毎回実行されるメインコード
初期化	初期化のために最初に1回だけ実行されるコード
変数定義	グローバル変数の定義
事前定義マクロ	外部ヘッダーファイル又はCファイルの前に定義され、コードを前処理するマクロです。例えば、IQmathライブラリでは、ライブラリが浮動小数点（MATH_TYPE = 1）か固定小数点（MATH_TYPE = 0）かを判断するために "MATH_TYPE" というマクロを使用します。マクロ "#define MATH_TYPE 1" をこのタブで定義することができます。
ヘッダーファイル	
外部Cファイル	外部ヘッダーファイル.h 外部Cソースファイル

ダイアログWindowに含まれる他の機能は次のようになります。

機能	説明
イメージ編集	Cブロックのイメージを編集します。
固定小数点を有効に	このチェックボックスにチェックを入れると固定小数点のブロックになります。そうでない場合は浮動小数点ブロックとなります。
固定小数点設定	固定小数点ブロックの時にCブロックの入出力のデータフォーマットを定義します。
変数マクロ	メイン回路からCコードへ渡す変数を定義します。変数値としては数値または数式を使用できます。これらの数式にはパラメータファイルで定義された変数やコマンドラインから回路図へ渡された変数を含むことができます。数式はシミュレーション開始前に評価され、t(時間)またはこのブロックの入出力値を含むことはできません。 例： a=15.2 b=Freq/sqrt(2) 'Freq'はパラメータファイルで定義されています。これらの変数はSimCoderでも使えます。SimCoderで生成されたコードには変数の計算式ではなくその評価値を使います。
コード検査	コードの構文検査をします。

SimCoderプリプロセッサはシンプルコードを生成するためにSimCoderCブロックで使われます。C言語のプロセッサと区別するために '##' を使います。次のプリプロセッサをサポートしています。

```
##if condition  
##elif condition  
##else  
##endif
```

“範例/SimCoder”フォルダのそれぞれのターゲットフォルダにSimCoderCブロックの使い方の例があります。

6 IQmath Library

6.1 概要

TIのIQmath Libraryが浮動小数点コードを固定小数点コードとして接続できるように関数リストとして準備されています。これにより処理速度が上がり精度も高くなります。

IQmath Libraryの詳細につきましては関連するTI資料を参照してください。

6.2 IQmath データタイプ、レンジ及び分解能

IQmath 機能の入出力は 32 ビットの固定小数点型です。

データのタイプは_iq で _iq1 から _iq30、GLOABLE_Q フォーマットは IQ1 から IQ30 のフォーマットとなります。データのレンジと分解能一覧は次の表のようになります。

表 6- 1 IQmath データタイプ、レンジ及び分解能一覧

データタイプ	レンジ		分解能/精度
	Minimum	Maximum	
_iq30	-2	1.999 999 999	0.000 000 001
_iq29	-4	3.999 999 998	0.000 000 002
_iq28	-8	7.999 999 996	0.000 000 004
_iq27	-16	15.999 999 993	0.000 000 007
_iq26	-32	31.999 999 985	0.000 000 015
_iq25	-64	63.999 999 970	0.000 000 030
_iq24	-128	127.999 999 940	0.000 000 060
_iq23	-256	255.999 999 981	0.000 000 119
_iq22	-512	511.999 999 762	0.000 000 238
_iq21	-1024	1023.999 999 523	0.000 000 477
_iq20	-2048	2047.999 999 046	0.000 000 954
_iq19	-4096	4095.999 998 093	0.000 001 907
_iq18	-8192	8191.999 996 185	0.000 003 815
_iq17	-16384	16383.999 992 371	0.000 007 629
_iq16	-32768	32767.999 984 741	0.000 015 259
_iq15	-65536	65535.999 969 482	0.000 030 518
_iq14	-131072	131071.999 938 965	0.000 061 035
_iq13	-262144	262143.999 877 930	0.000 122 070
_iq12	-524288	524287.999 755 859	0.000 244 141
_iq11	-1048576	1048575.999 511 719	0.000 488 281
_iq10	-2097152	2097151.999 023 437	0.000 976 563
_iq9	-4194304	4194303.998 046 875	0.001 953 125
_iq8	-8388608	8388607.996 093 750	0.003 906 250
_iq7	-16777216	16777215.992 187 500	0.007 812 500
_iq6	-33554432	33554431.984 375 000	0.015 625 000
_iq5	-67108864	67108863.968 750 000	0.031 250 000
_iq4	-134217728	134217727.937 500 000	0.062 500 000
_iq3	-268435456	268435455.875 000 000	0.125 000 000
_iq2	-536870912	536870911.750 000 000	0.250 000 000
_iq1	-1073741824	1 073741823.500 000 000	0.500 000 000

7 TI F2833x Hardware Target

7.1 概要

オプションモジュールTI F2833x Hardware TargetとSimCoderの組み合わせによって、PSIMの制御回路図からTexas Instruments社製浮動小数点型DSP F2833xシリーズを搭載した基板で汎用的に使用できるコードを生成することができます。

TI F2833xハードウェアターゲットはF2833x全てのパッケージに対応しています。次の2ページにわたる図はLQFPのF2833xDSPのピンアサインです。TI F2833xHardware Targetが対応している主な関数は図内で色を付けています。

TI F2833x Hardware Targetには下記の機能があります。

- 3-phase、2-phase、1-phase、とAPWM(Single PWM)生成機能
- 可変周波数PWM
- PWM生成のStart、Stop機能
- トリップゾーン、トリップゾーンステート機能
- A/D変換機能
- デジタル入力出力機能
- SCI基板設定機能、入力出力
- SPI基板設定機能、デバイス、入力出力
- CAN設定機能、入力出力
- キャプチャ、キャプチャステート機能
- エンコーダ、エンコーダステート機能
- Up/Down カウンタ 機能
- 割り込み時間機能
- DSP クロック機能
- ハードウェア基板設定機能

複数のサンプリング周波数でコードを生成する回路の場合、SimCoder は優先的に PWM 割り込みの周波数を使います。他のサンプリング周波数では、タイマ 1 割り込み、タイマ 2 割り込みの順に使われます。3 つ以上のサンプリング周期がある場合は対応する割り込み関数をメイン関数内で実行する形にします。

PWM 生成部からハードウェア割り込みを発生させることができますので、SimCoder は PWM 生成ブロックに繋がっている PWM 生成ブロックと同じ周波数を持つ全ての素子を検索してグループ化します。各素子ブロックは自動的に配置され、生成されるコード内の割り込みサービスルーチンタスクとして実行されます。

また、ハードウェア割り込みはデジタル入力、エンコーダ、キャプチャ、そしてトリップゾーンにもあります。各ハードウェア割り込みは割り込みブロックから生成され、割り込みブロックは割り込み関数と繋げる必要があります。（割り込み関数はサブ回路で作ります。）二つ以上の割り込みを使うときはそれぞれの機能に割り込みブロックと割り込み関数を設定します。

本章では TI F2833x Hardware Target ライブラリ内の各素子の使い方について解説します。

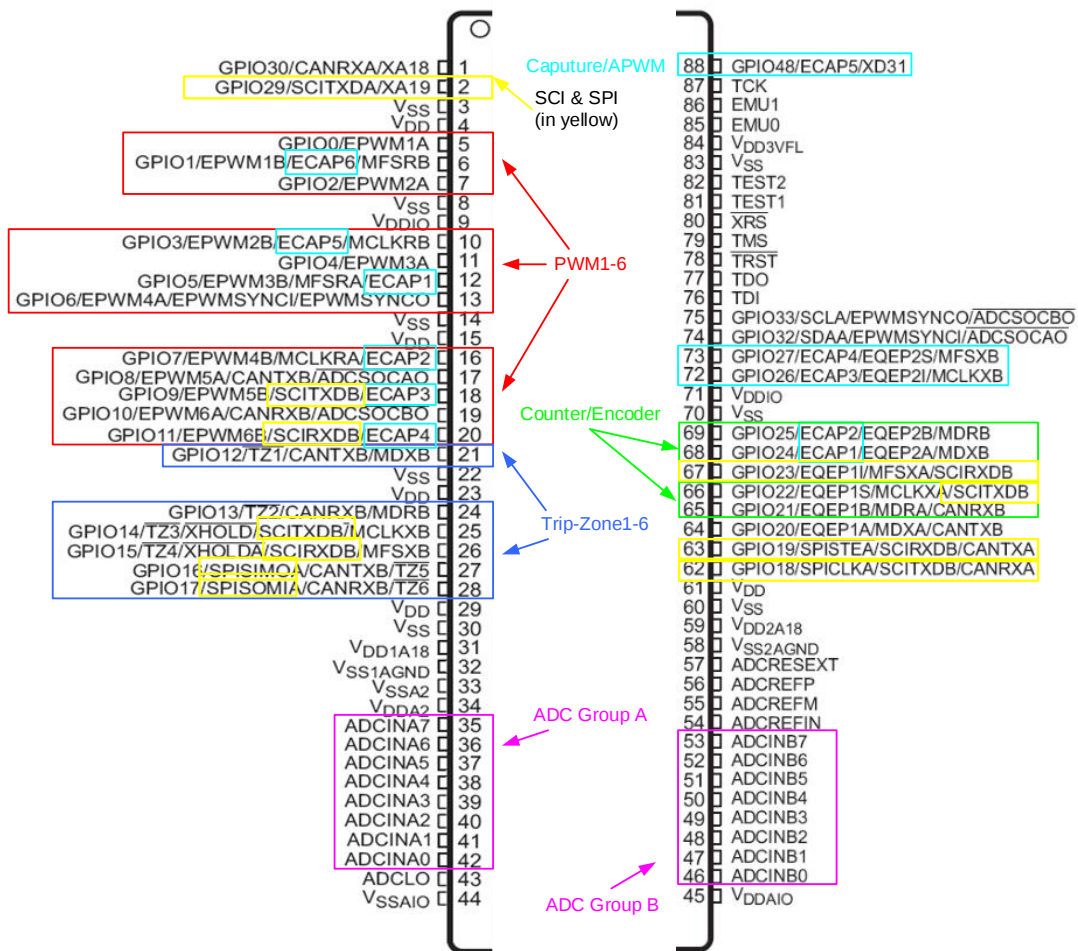


図 7-1 F28335 DSP ポートアサイン (Pin 1-88)

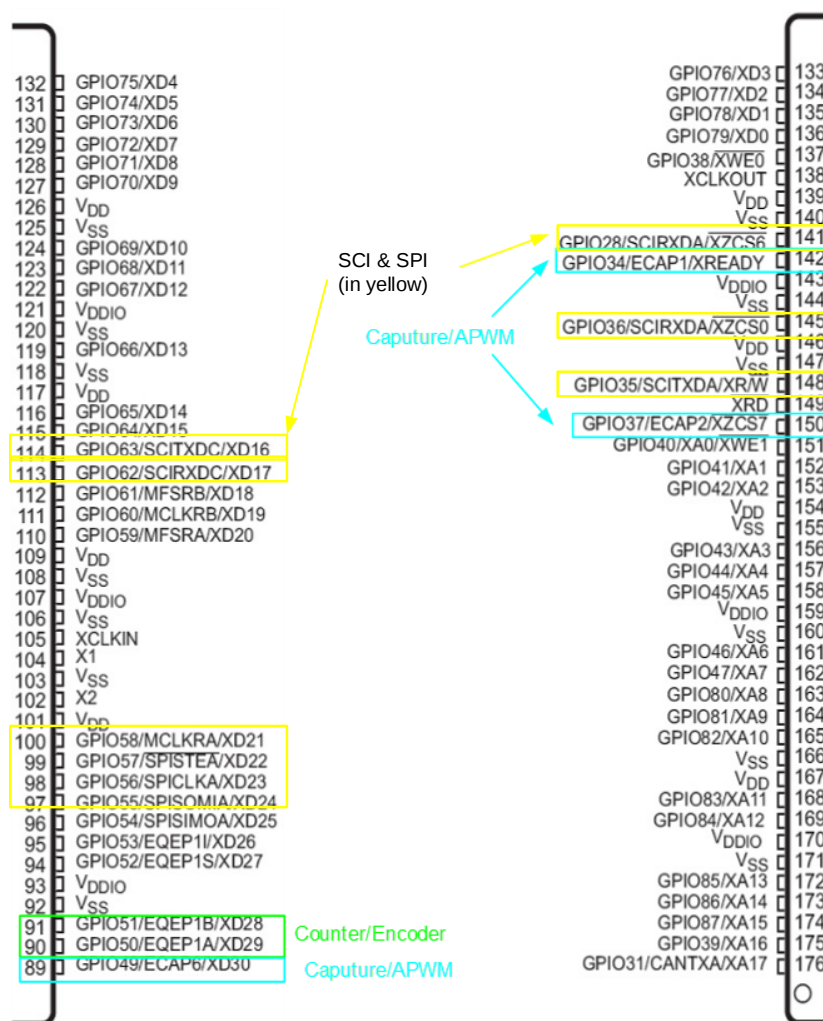


図 7-2 F28335 DSP ポートアサイン (Pin 89-176)

7.2 ハードウェア構成

F2833x には 88 の GPIO ポート (GPIO0~GPIO87) があります。各ポートは違う機能をもっています。しかし、特定の DSP ボード用ではすべてのポートが外部からのアクセスが可能ではなく、またいくつかは機能が固定されています。ハードウェアコンフィギュレーションブロックを使うと DSP ボードに SimCoder を設定できます。

シンボル



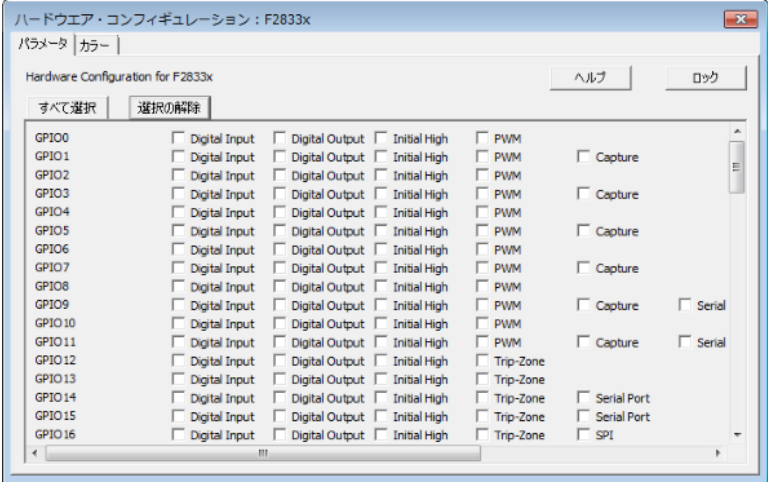


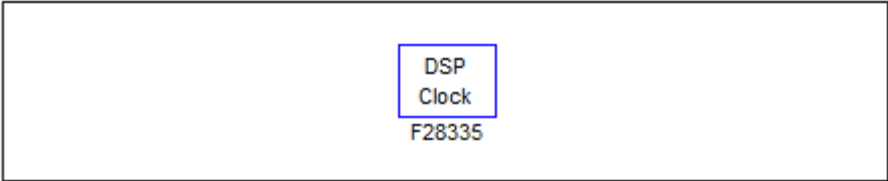
図 7.3 ハードウェアコンフィギュレーションブロックの設定

各々の GPIO ポートのチェックボックスにある機能が使用可能です。SimCoder ではこのボックスがチェックされている機能だけが使用できます。例えばポート GPIO1 では “Digital Input”, “Digital Output”, “PWM”, “Capture” が使用できます。ポート GPIO1 の “PWM” のみをチェックしているにもかかわらず回路のポート GPIO1 で “Digital Input” として使用すると SimCoder はエラーとなります。

7.3 DSP クロック

DSP クロックブロックは外部クロック周波数と F2833x DSP の周波数の設定ができます。

シンボル:



仕様:

パラメータ	機能
外部 Clock (MHz)	DSPボードの外部クロック周波数の設定（単位はMHz）。 入力する周波数は整数値である必要があり、30MHzが最大値です。
DSP 速度 (MHz)	DSPの周波数の設定（単位はMHz）。 入力する周波数は整数値である必要があり、外部クロック周波数の整数倍（1~12倍）である必要があります。また、150MHzが最大値です。

DSP クロックブロックが回路中で使用されていない場合は DSP ブロックのデフォルト値が使用され

ます。

7.4 PWM 生成器

F2833x には、

PWM1(GPIO0、GPIO1)

PWM2(GPIO2、GPIO3)

PWM3(GPIO4、GPIO5)

PWM4(GPIO6、GPIO7)

PWM5(GPIO8、GPIO9)

PWM6(GPIO10、GPIO11)

のように合計で 6 組の PWM 出力があります。各組は互いに相補的になっており、特別な設定の場合を除いて、PWM1A が正の出力ならば PWM1B は逆の負の出力になります。

SimCoder では 6 つの PWM を下記のように使うことができます。

1. 3-phase PWM 生成器 : 2 種類あります。PWM1、2、3 と PWM4、5、6 という組み合わせです。
2. 2-phase PWM 生成器 : 6 種類あります。PWM1~6 を相補的な動作ではなく、設定によって幾つかの特別なモードで動作させることができます。
3. 1-phase PWM 生成器 : PWM1~6 互いに相補的な二つの出力をもっています。
4. 位相シフト付 1-phase PWM 生成器 : PWM2~6 は互いに相補的な 2 つの出力をもっています。

これらの PWM 生成器は A/D 変換のトリガを掛けることや、トリップゾーン信号を用いることもできます。

上記で説明した PWM 生成器の他にもう一点、6 つの APWM 生成器があります。これは機能的には他のブロックに比べて制限されているものとなっており、A/D 変換トリガやトリップゾーンが使えず、更にピンがキャプチャと共用になっているため、キャプチャ機能と同時に使うことができません。

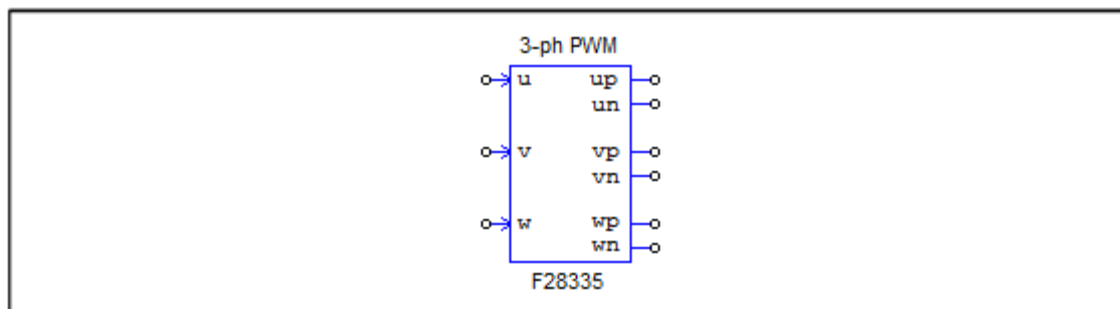
SimCoder の PWM 生成器は内部にスイッチング一回分の遅れを持っており、PWM の出力は入力が入ってから 1 サイクル分遅れて出力されます。これは実際の DSP の持つ遅れをシミュレーションに反映するためです。

PWM 周波数倍率は F2833x の制限では 1~3 です。(設定は 1 から 100 までできます。3 より大きくしたい場合は、PSIM が未使用の PWM を使って、制御周波数で割り込みを引き起こします。この未使用の PWM は周期的な割り込みを生成するためにしか使用されません、PWM の出力端子は、他の機能に設定することができます。もしシステム内に未使用 PWM がない場合、CPU タイマーが使用されます。

7.4.1 3-phase PWM 生成器：

三相 PWM 生成器の図中では、u、v、w は三相を示しています。（a、b、c 相とも呼ばれます。）p は正の出力を示し、n は負の出力を示しています。例えば、三相 PWM123 では、“up” は PWM1A、“un” は PWM1B となります。

シンボル:



仕様:

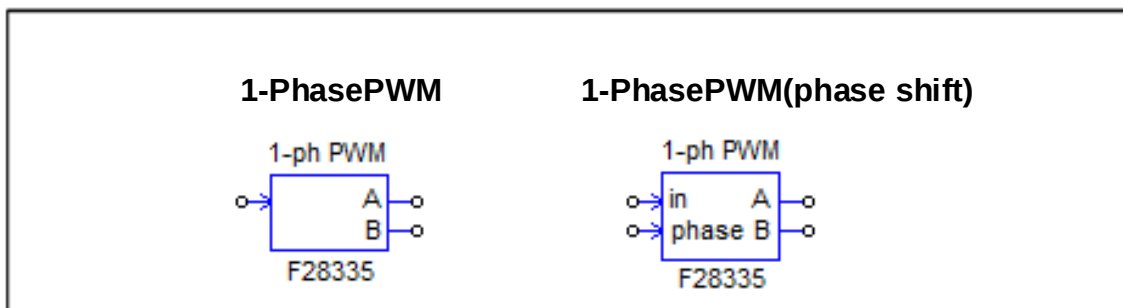
パラメータ	機能
PWM ソース	PWM生成器の出力ピン設定。「3-phase PWM 123」（PWM1~3）か「3-phase PWM 456」（PWM4~6）のどちらかを選択します。
デッドタイム	PWM生成器のデッドタイム（Td）の設定。単位は[sec]
サンプリング周波数	PWM生成器のサンプリング周波数の設定。単位は[Hz] PWM信号のデューティサイクルはここで設定した周波数で更新されます。
PWM周波数倍率	PWM周波数とサンプリング周波数の倍率の設定。 1、2、3の設定ができます。 PWM周波数（制御スイッチで生成されるPWM出力信号の周波数）がサンプリング周波数の何倍かを設定します。 例）サンプリング周波数が50kHzで本設定値が3の場合はPWM周波数が150kHzになります。150kHzのスイッチングでは3回につき1回値の更新を行います。
キャリア波形タイプ	キャリア波形タイプの設定。 「Triangular(start low)」：キャリアを三角波、PWM出力の初期値はLow。 「Triangular(start high)」：キャリアを三角波、PWM出力の初期値はHIGH。 「Sawtooth(start low)」：キャリアをのこぎり波、PWM出力の初期値はLow。 「Sawtooth(start high)」：キャリアをのこぎり波、PWM出力の初期値はHIGH。
A/D変換トリガ	A/D変換へのトリガの設定。 「Do not trigger ADC」：A/D変換トリガ機能を無効にします。 「Trigger ADC Group A」：A/D変換器のGroup Aへのトリガを有効にします。 「Trigger ADC Group B」：A/D変換器のGroup Bへのトリガを有効にします。 「Trigger ADC Group A&B」：A/D変換器のGroup AとGroup Bの両方へのトリガを有効にします。
A/D変換トリガ位置	A/D変換トリガを出力する位置の設定。値は0~1まで設定可能です。 （例）0の場合、PWM周期の始まりでA/D変換トリガを掛けます。 0.5の場合、PWM周期の180° の位置でA/D変換トリガを掛けます。

トリップゾーンi 使用	PWM生成器と繋がる6つの各トリップゾーンの設定。 - 「Disable Trip-Zone i」：対応するトリップゾーン信号を無効にします。 - 「One shot」：One shotモードでトリップゾーン信号を使用します。トリップゾーン信号が検出されると、PWM出力は手動で開始させる必要があります。 - 「Cycle by cycle」：cycle-by-cycleモードでトリップゾーン信号を使用します。そのときの周期内でトリップゾーン信号が有効になり、次の周期でPWMは自動的に再出力されるようになります。
トリップ アクション	トリップゾーン信号に対するPWM生成器の動作の設定。 - 「High Impedance」：PWM出力部はハイインピーダンスになります。 - 「PWM A high & PWM B low」：PWMの正の出力はハイレベル、負の出力はローレベルになります。 - 「PWM A low & PWM B high」：PWMの正の出力はローレベル、負の出力はハイレベルになります。 - 「No action」：何も動作を設定しません。
ピーク間電圧	キャリア波のピーク間電圧Vppの設定。
オフセット電圧	キャリア波のオフセット電圧Voffsetの設定。
初期入力値u, v, w	u、v、w三相入力ノードの初期値の設定。
最初からPWM 信号出力	開始からPWM信号出力をするかどうかの設定。 - 「Start」：プログラム開始時にPWM出力を許可します。 - 「Do not start」：「Start PWM」関数を実行するまでPWM出力をしません。
シミュレーション 出力モード	シミュレーションの出力モードはSwitchingモードかAverageモードに設定されます。“Switchingモード”へ設定された時にはPWMブロックの出力はPWM信号となります。“Averageモード”へ設定された時にはPWMブロックの出力は平均モード信号となります 例えばPWMブロックの入力をVuとすると出力のupとunは $V_{up} = V_u / (V_{pk_pk} + V_{offset})$ $V_{un} = -V_u / (V_{pk_pk} + V_{offset})$ となります。Average mode に設定されるとPWMブロックの出力はaverage mode modelのインバータへ接続されます。

7.4.2 1-phase PWM 生成器（外部位相シフト機能付きを含む）：

外部入力位相シフト機能付き 1-phase PWM 生成器は 2 入力の素子となります。1 つは PWM 入力（“in”ノード）、もう 1 つは位相シフト[deg]の入力（“phase”ノード）です。

シンボル:



仕様:

パラメータ	機能
PWM ソース	PWM生成器の出力ピン設定。位相シフトを使用しない場合は「PWM1」から「PWM6」まで、位相シフトを使用する場合は「PWM2」から「PWM6」を選択します。
出力モード	PWM生成器の出力モードの設定。 「Use PWM A&B」：PWM出力のAとBを両方使います。また互いに相補的になります。 「Use PWM A」：PWM出力Aのみ使用します。 「Use PWM B」：PWM出力Bのみ使用します。
デッドタイム	PWM生成器のデッドタイム (Td) の設定。単位は[sec]
サンプリング周波数	PWM生成器のサンプリング周波数の設定。単位は[Hz] PWM信号のデューティはここで設定した周波数で更新されます。
PWM 周波数倍率	PWM周波数とサンプリング周波数の倍率の設定。 1、2、3の設定ができます。 PWM周波数（制御スイッチで生成されるPWM出力信号の周波数）がサンプリング周波数の何倍かを設定します。 例）サンプリング周波数が50kHzで本設定値が3の場合はPWM周波数が150kHzになります。150kHzのスイッチングでは3回につき1回値の更新を行います。
キャリア波形タイプ	キャリア波形タイプとPWM出力信号の初期状態の設定。 「Triangular(start low)」：キャリアを三角波、PWM出力の初期値はLow。 「Triangular(start high)」：キャリアを三角波、PWM出力の初期値はHIGH。 「Sawtooth(start low)」：キャリアをのこぎり波、PWM出力の初期値はLow。 「Sawtooth(start high)」：キャリアをのこぎり波、PWM出力の初期値はHIGH。
A/D変換トリガ	A/D変換へのトリガの設定。 「Do not trigger ADC」：A/D変換トリガ機能を無効にします。 「Trigger ADC Group A」：A/D変換器のGroup Aへのトリガを有効にします。 「Trigger ADC Group B」：A/D変換器のGroup Bへのトリガを有効にします。 「Trigger ADC Group A&B」：A/D変換器のGroup AとGroup Bの両方へのトリガを有効にします。
A/D変換トリガ位置	A/D変換トリガを出力する位置の設定。値は0~1まで設定可能です。 (例) 0の場合、PWM周期の始まりでA/D変換トリガを掛けます。 0.5の場合、PWM周期の180° の位置でA/D変換トリガを掛けます。
トリップゾーン i 使用	PWM生成器と繋がる6つの各トリップゾーンの設定。 「Disable Trip-Zone i」：対応するトリップゾーン信号を無効にします。 「One shot」：One shotモードでトリップゾーン信号を使用します。トリップゾーン信号が検出されると、PWM出力は手動で開始させる必要があります。 「Cycle by cycle」：cycle-by-cycleモードでトリップゾーン信号を使用します。そのときの周期内でトリップゾーン信号が有効になり、次の周期でPWMは自動的に再出力されるようになります。
トリップアクション	トリップゾーン信号に対するPWM生成器の動作の設定。 「High Impedance」：PWM出力部はハイインピーダンスになります。 「PWM A high & PWM B low」：PWMの正の出力はハイレベル、負の出力はローレベルになります。 「PWM A low & PWM B high」：PWMの正の出力はローレベル、負の出力はハイレベルになります。

	力はハイレベルになります。 - 「No action」：何も動作を設定しません。
ピーク間電圧	キャリア波のピーク間電圧Vppの設定。
オフセット電圧	キャリア波のオフセット電圧Voffsetの設定。
位相シフト	PWM生成器の出力に対してシフトする位相の値[deg] (外部位相シフト入力なしの1-phase PWM生成器)
初期入力値	入力ノードの初期値の設定。
Use HRPWM	高分解能PWMの定義を行います。 <i>Do not use HRPWM</i> : 高分解能PWMを使用しません。 <i>UseHRPWM</i> (校正なし): 校正なしで高分解能PWMを使用します。 <i>UseHRPWM</i> (校正あり): 校正ありで高分解能PWMを使用します。
最初からPWM信号	開始からPWM信号出力をするかどうかの設定。 - 「Start」：プログラム開始時にPWM出力を許可します。 - 「Do not start」：「Start PWM」関数を実行するまでPWM出力をしません。
シミュレーション出力モード	シミュレーションの出力モードはSwitchingモードかAverageモードに設定されます。 “Switchingモード”へ設定された時にはPWMブロックの出力はPWM信号となります。 “Averageモード”へ設定された時にはPWMブロックの出力は平均モード信号となります。 例えばPWMブロックの入力をVuとすると出力のupとunは $V_{up} = V_u / (V_{pk_pk} + V_{offset})$ $V_{un} = -V_u / (V_{pk_pk} + V_{offset})$ となります。Average mode に設定されるとPWMブロックの出力はaverage mode modelのインバータへ接続されます。

1-phase PWM 生成器は、別の PWM 信号に対して位相をシフトさせた PWM 信号を生成することができます。「PWM1、2、3」と「PWM1、4、5、6」の2つの組み合わせがあります。

- 基準のPWMと位相をシフトさせるPWMは、同じ組み合わせのものでなければなりません。
つまり、PWM1を基準として、PWM2と3、またはPWM4、5、6がPWM1に対して位相シフトさせることができます。またはPWM2を基準として、PWM3はPWM2に対して位相シフトさせることができます。同様に、PWM4（または5）を基準として、PWM5（または6）をPWM4（または5）に対して位相シフトさせることができます。
しかし、PWM4、5、6の基準としてPWM2または3を使用することは許可されていません。
- また、基準となるPWMとシフトされるPWMは、組み合わせ内で連続している必要があります。
つまり、PWM1を基準としてPWM3、またはPWM5またはPWM6を位相シフトすることはできません。

位相シフトの値は、度[deg]になります。値が-30°の場合、出力は基準のPWM生成器の出力に対してスイッチングサイクルの30°右に（遅れ）にシフトされます。これは、30°右にPWMの搬送波をシフトすることと同じです。位相の値が30°の場合、出力は30°左（進み）にシフトされます。

搬送波

搬送波には2種類あります。三角波（等しい傾きの間隔の立ち上がりと立ち下がりを持つ）とノコギリ波です。加えて、後述のような“Start-Low”と“Start-High”の2つの動作モードがあります。

三角搬送波の PWM 生成器の入力波形と出力波形は以下のとおりです。

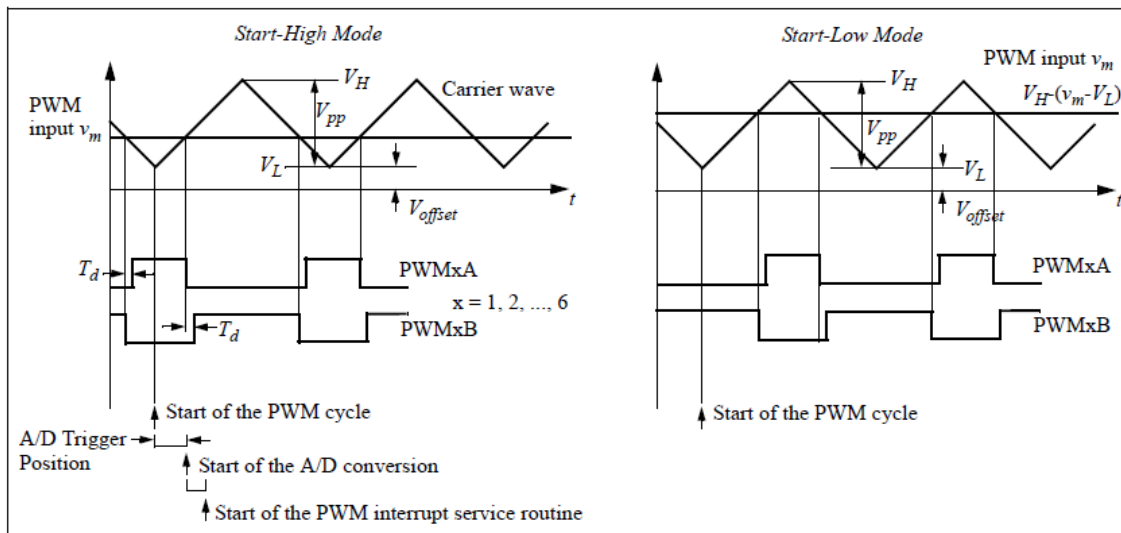


図 7.1 三角搬送波の PWM 生成器の入力波形と出力波形

ノコギリ搬送波の PWM 生成器の入力波形と出力波形は以下のとおりです。

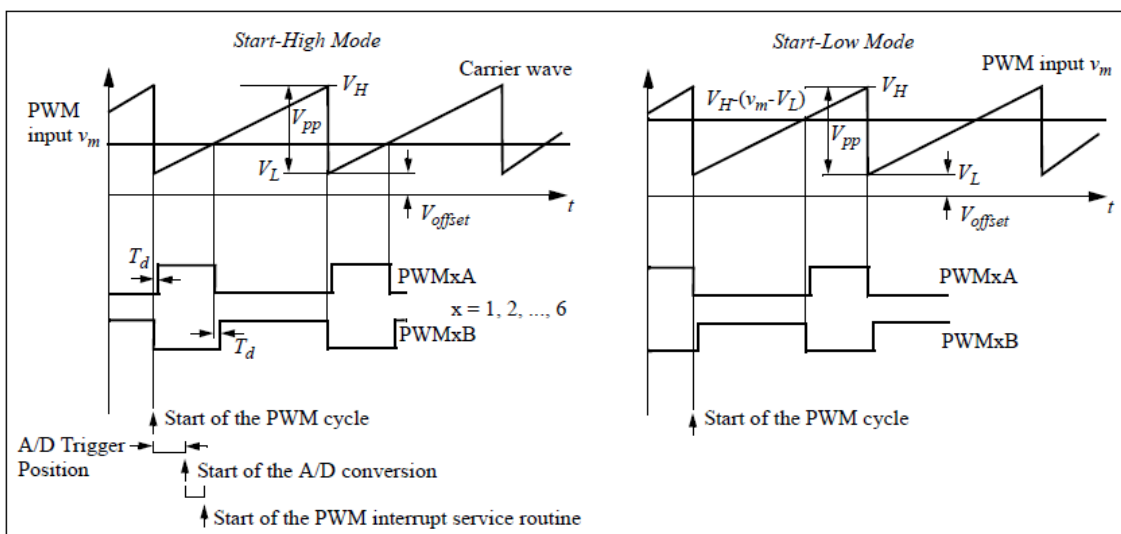


図 7.2 ノコギリ搬送波の PWM 生成器の入力波形と出力波形

上記の図は、デッドタイムの定義と PWM 生成器が A/D コンバータを動作させる際の時系列を示しています。PWM サイクルの開始時に A/D コンバータのトリガを設定した場合、A/D トリガポジションによって定義されるある一定の遅延の後、A/D 変換を開始します。A/D 変換が完了した後、PWM 割り込みルーチンが開始されます。

PWM 生成器が A/D コンバータを動作させない場合は、PWM サイクルの開始時に PWM 割り込みルーチンが開始されます。

上記の図は、“Start-High”と“Start-Low”モードの動作方法を示しています。PWM 入力を“Vm”、搬送波の最小値を“VL”、最大値を“VH”とします。“Start-High”モードでは、PWM の正出力 PWMA はスイッチングサイクルの開始時に HIGH となり、入力“Vm”が搬送波よりも大きい間は HIGH を持続します。例えば、搬送波が 0~1、つまり VL=0、VH=1 の時、Vm が 0.2 の場合、PWM 出力 PWMA は搬送波が 0.2 より小さい間は HIGH を持続します。

一方、“Start-Low”モードでは、PWM 正出力 PWMA はスイッチングサイクルの開始時に LOW となり、搬送波が「 $VH - (Vm - VL)$ 」の値より大きい際に HIGH となります。例えば、搬送波が 0~1、つまり VL=0、VH=1 の時、Vm が 0.2 の場合、PWM 出力 PWMA は搬送波が 0.8 より大きい間は HIGH となります。

キャリア開始モードはスイッチ電流の測定方法によりかわります。3 相インバータの場合はスイッチ電流が測定され、スタートハイモードが選択されます。これは上のスイッチが周期の最初にトップスイッチゲーティング信号が高く、電流が流れていることとなります。

一方低いスイッチ電流が測定されているとスタートローモードが選択されます。これは周期の最初に

トップスイッチゲーティング信号が低く、下のスイッチゲーティング信号が高く、電流が流れていることとなります。

次のような回路の場合、位相 A と B の下のスイッチ電流が測定されます。この場合はキャリアスタートローモードを選択すべきです。

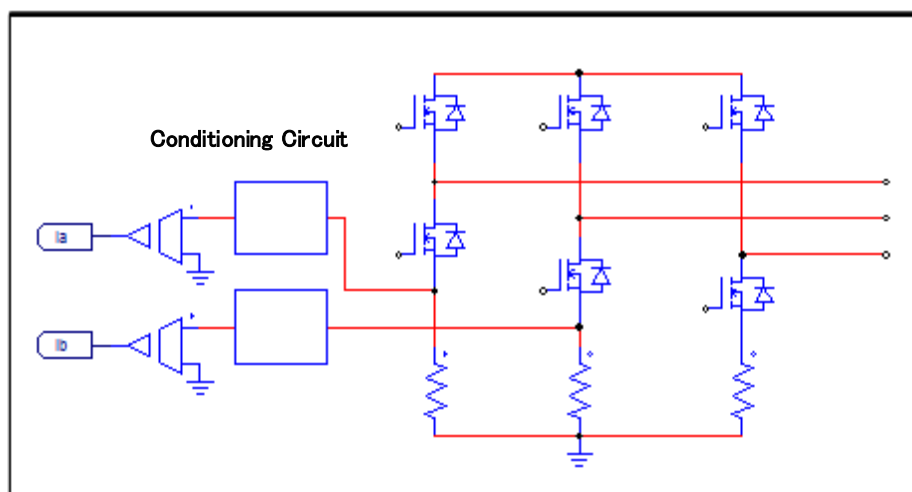


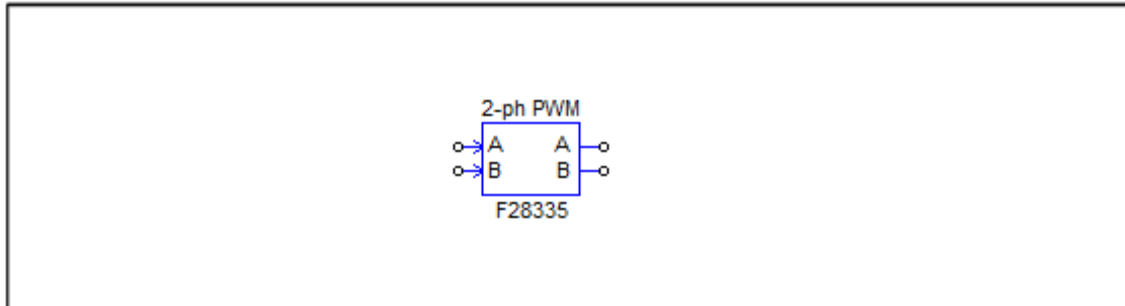
図 7.3 下のスイッチ電流が測定される場合の回路

注：“Start-Low”モードでは、PWM 信号を生成するために搬送波と比較する前に PWM 入力“Vm”は“ $VH - (Vm - VL)$ ”に変換されます。変換によって、“Start-Low”と“Start-High”の両方のモードで同じデューティ比の表現となります。例えば、VL=0、VH=1 のノコギリ波、または VL=-VH の三角波の場合、PWMA 出力のデューティ比 D は“Start-Low”と“Start-High”の両方で「 $D = Vm / VH$ 」となります。

7.4.3 2-phase PWM 生成器 :

2 層 PWM ブロックは 6 つの動作モードがあります。

シンボル:



仕様:

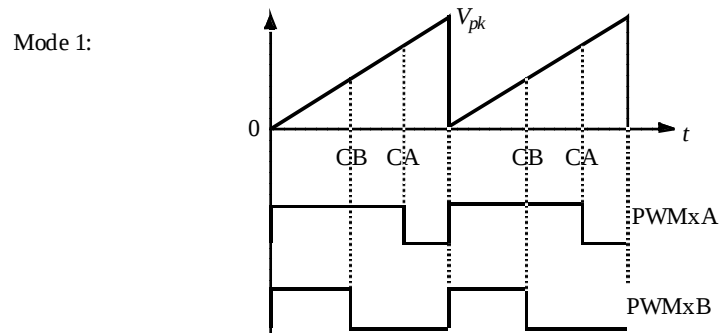
パラメータ	機能
PWM ソース	PWM生成器の出力ピン設定。 PWM1~6までを選択します。
出力モード	PWM生成器の動作モードの設定。 6つのモードから一つを選択します。各モードの動作の詳細は欄外の図を参照してください。
サンプリング 周波数	PWM生成器のサンプリング周波数の設定。単位は[Hz] PWM信号のデューティはここで設定した周波数でアップデートされます。
PWM 周波数倍率	PWM周波数とサンプリング周波数のスケーリングファクターの設定。 1、2、3の設定ができます。 PWM周波数（制御スイッチで生成されるPWM出力信号の周波数）がサンプリング周波数の何倍かを設定します。 例）サンプリング周波数が50kHzで本設定値が2の場合はPWM周波数が100kHzになります。100kHzのスイッチング2回につき1回値のアップデートを行います。
A/D変換トリガ	A/D変換へのトリガの設定。 - 「Do not trigger ADC」 : A/D変換トリガ機能を無効にします。 - 「Trigger ADC Group A」 : A/D変換器のGroup Aへのトリガを有効にします。 - 「Trigger ADC Group B」 : A/D変換器のGroup Bへのトリガを有効にします。 - 「Trigger ADC Group A&B」 : A/D変換器のGroup AとGroup Bの両方へのトリガを有効にします。
A/D変換トリガ 位置	ADCチャンネルのPWMモジュールによるトリガされた変換位置。トリガされた変換位置は、搬送波の開始時に、または搬送波の途中のいずれかに設定できます。 途中トリガ位置は、動作モード4,5、及び6の場合に有効です。 もしPWMがADCをトリガーしない場合は、この位置はPWM割り込みの位置になります。
トリップゾーンi 使用	PWM生成器と繋がる6つの各トリップゾーンの設定。 - 「Disable Trip-Zone I」 : 対応するトリップゾーン信号を無効にします。Z - 「One shot」 : One shotモードでトリップゾーン信号を使用します。トリガが掛かった後はPWMは手動で出力を開始するようになります。 - 「Cycle by cycle」 : cycle-by-cycleモードでトリップゾーン信号を使用します。そのときの周期内でトリップゾーン信号が有効になり、次の周期でPWMは自動的

	に再出力されるようになります。
トリップアクション	トリップゾーン信号に対するPWM生成器の動作の設定。 - 「High Impedance」 : PWM出力部はハイインピーダンスになります。 - 「PWM A high & PWM B low」 : PWMの正の出力はハイレベル、負の出力はローレベルになります。 - 「PWM A low & PWM B high」 : PWMの正の出力はローレベル、負の出力はハイレベルになります。 - 「No action」 : 何も動作を設定しません。
ピーク間電圧	キャリア波のピーク電圧 V_{pk} の設定。
初期入力値A, B	A、B入力ノードの初期値の設定。
最初からPWM信号	開始からPWM信号出力をするかどうかの設定。 - 「Start」 : 開始からPWM出力を許可します。 - 「Do not start」 : 「Start PWM」関数を実行するまでPWM出力をしません。

2-phase PWM 生成器の出力形式は動作モード設定によって変わります。(以降の図参照) モードによってキャリアは三角波とのこぎり波のどちらかになります。また、キャリア値は 0 から V_{pk} までの間の値を取り、DC オフセットの設定はできません。

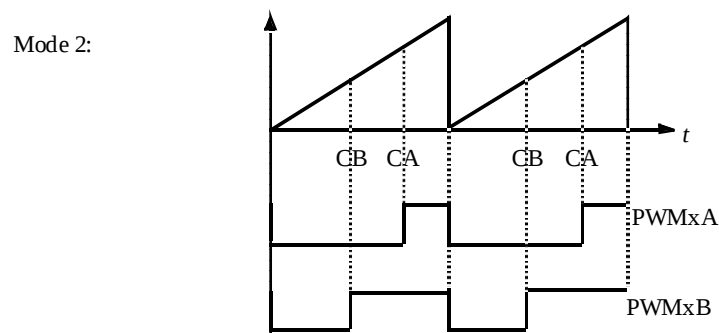
動作モード 1 :

図の CA と CB は 2-phase PWM 生成器の二つの入力 A、B になり、各出力のターンオフタイミングを決めます。



動作モード 2 :

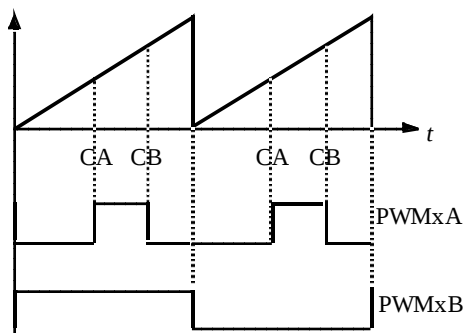
動作モード 1 と違い、各出力のターンオンタイミングを決めるモードです。



動作モード 3 :

入力 A は PWM A 出力のターンオンのタイミングを決め、入力 B は PWM B 出力のターンオフのタイミングを決めます。PWM B 出力は PWM 周期の間オンになり、次の周期の間はオフになる、を繰り返します。

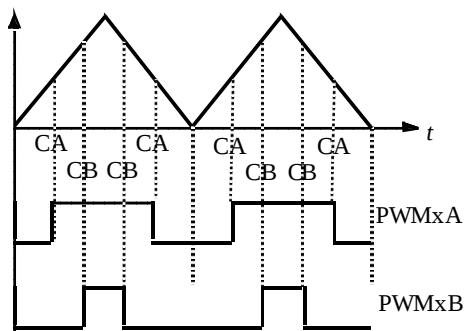
Mode 3:



動作モード 4 :

キャリアは三角波になります。各入力是对应する出力のターンオンとターンオフのタイミングを決めます。

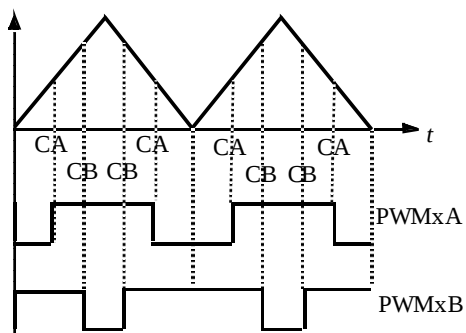
Mode 4:



動作モード 5 :

動作モード 4 と違い、PWM A 出力と PWM B 出力のターンオフとターンオンのタイミングが反対に動作します。

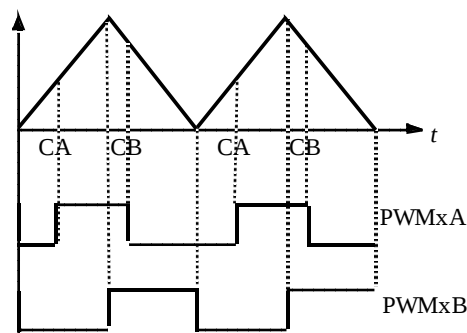
Mode 5:



動作モード 6 :

入力 A は PWM A のターンオンのタイミングを決め、入力 B は PWM A のターンオフのタイミングを決めます。PWM B は PWM 周期の最初の半周期でオンになり、次の半周期でオフになる動作をします。

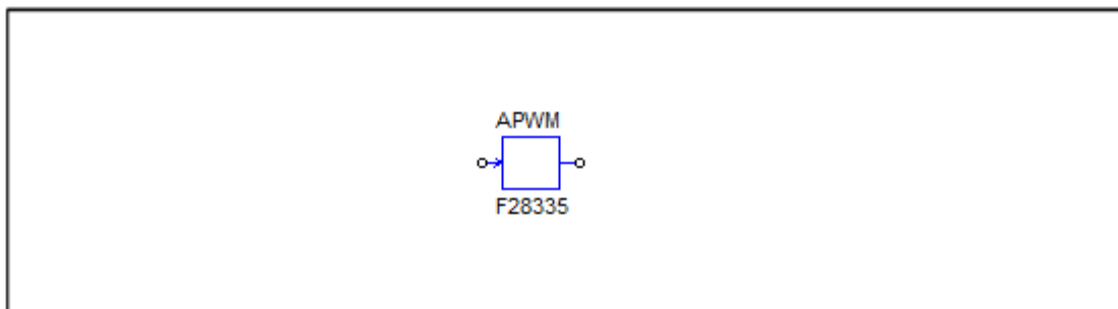
Mode 6:



7.4.4 Single PWM 生成器(APWM) :

SinglePWM 生成器は APWM ともわれキャプチャと同じリソースを共有しています。

シンボル:



仕様:

パラメータ	機能
PWM ソース	PWM生成器の出力ピン設定。 Single PWM生成器はキャプチャと同じピンを共有しており、下記のように各 APWMは決められたピンから出力させることが可能です。 - APWM1(GPIO5, GPIO24, GPIO34) - APWM2(GPIO7, GPIO25, GPIO37) - APWM3(GPIO9, GPIO26) - APWM4(GPIO11, GPIO27) - APWM5(GPIO3, GPIO48) - APWM6(GPIO1, GPIO49)
PWM 周波数	PWM生成器の周波数の設定。単位は[Hz]
キャリア波形タイプ	キャリア波形タイプの設定。 次のいずれかになります。: - のこぎり波 (低でスタート): キャリアをのこぎり波、PWM出力の初期値はLOW。 - のこぎり波 (高でスタート): キャリアをのこぎり波、PWM出力の初期値はHIGH。
ストップ出力	PWMジェネレータが停止している時の出力設定。

	- 出力 低: PWM出力は低に設定されます。 - 出力 高: PWM出力は高に設定されます。
ピーク間電圧	キャリア波のピーク間電圧の設定。
オフセット値	キャリア信号のオフセット電圧の設定。
位相シフト	PWM生成器の出力に対しての位相シフト値（単位deg）
初期入力値	入力の初期値の設定。
最初からPWM信号	開始からPWM信号出力をするかどうかの設定。 - /Start: 開始からPWM出力を許可します。 - /Do not start: 「Start PWM」関数を実行するまでPWM出力をしません。

1 相 PWM 生成器と同様に APWM 生成器は他の PWM 生成器に関して位相シフトをもつ PWM 信号を生成します。位相シフトのルールは 1 相 PWM 生成器と同じです。

Single PWM 生成器は機能的には他のブロックに比べて制限されており、A/D 変換トリガやトリップゾーン信号が利用できません。

7.4.5 PWM ブロック間の同期

PWM の 3 つのタイプで同期が可能です。位相シフトはお互いの中で定義できます。単相 PWM、単相 PWM(位相シフト付)と APWM(もしくは単層 PWM(キャプチャと共用)です。

単相 PWM ブロックは別の PWM 信号で位相シフトした PWM 信号を生成できます。通常の PWM ブロックには PWM1、2、3 と PWM1、4、5、6 の 2 つのシリーズがあります。

同様に APWM ブロックには位相シフトのための 2 つのシリーズがあります。それは PWM1、APWM1、2、3 と PWM1、APWM4、5、6 です。

位相シフトのための PWM ブロックの定義は次のようになります。

ー基準 PWM と位相シフトされる PWM は同じシリーズでなければなりません。すなわち PWM1 が基準、PWM2、3 か PWM4、5、6 が PWM1 に関して位相シフトできます。または、PWM2 が基準となる場合は PWM3 が PWM2 について位相シフトできます。

同様に PWM4 (または 5)は基準となり、PWM5 (または 6)は PWM4(または 5)に関して位相シフトできます。しかし、PWM4、5、6 に対して PWM2 または 3 を基準として使うことはできません。

これはまた APWM についても同様です。例えば PWM1 が基準となる場合、APWM1、2、3 は位相シフトされます。同様に PWM1 が基準の場合、APWM4、5、6 が位相シフトできます。または、APWM4 が基準の場合、APWM5 と 6 は位相シフトできます。

ー基準 PWM と位相シフトされる PWM は省略された PWM が使用されないようにシリーズの中で連続でなければなりません。例えばもし PWM1、2、3 すべてが使用されているが PWM2 が PWM1 と 3 と同期していない場合は使用できません。しかしもし PWM2 が回路中で使用されていない場合は PWM1 を基準、位相シフト PWM3 として使用することができます。

例えば正しい設定は次のようになります。最初が基準でその後のブロックが位相シフトとなります。

PWM 1 (基準), PWM 2, PWM 3

PWM 1 (基準), PWM 4, PWM 5, PWM 6

PWM 2 (基準), PWM 3

PWM 4 (基準), PWM 5

PWM 5 (基準), PWM 6

PWM 1 (基準), APWM 1, APWM 2, APWM 3

PWM 1 (基準), APWM 4, APWM 5, APWM 6

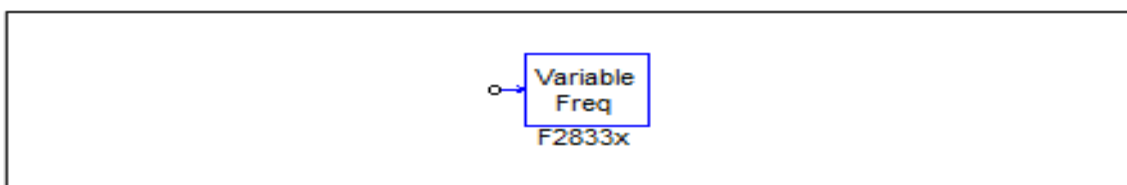
APWM 1 (基準), APWM2, APWM 3

APWM 4 (基準), APWM5, APWM 6

7.5 可変周波数 PWM

可変周波数 PWM ブロックでは PWM 生成器のサンプリング周波数の変更ができます。

シンボル：



仕様

パラメータ	機能
PWM ソース	PWM 生成器のソース PWW1~6、3 相 PWM123 と PWM456 を選択できます。
割り込みポジション調整	PWM 周波数が調整する時、割り込み位置を調整するかどうかを指定します。 次のいずれかになります： - 調整しない: 割り込み位置はベース周波数で計算された初期位置に維持します。 - 調整する: 割り込み位置は、新しい周波数に基づき再計算されず、次の PWM 周期で適用されます。

PWM ブロックと一致するサンプリング周波数は次のような PWM 周期の最初に変更されます。

$$PWM_Frequency = PWM_Base \text{ Frequency} / Input_Value$$

ここで $PWM_Base_Frequency$ は PWM_block に一致したサンプリング周波数です。

$Input_Value$ はこのブロックの入力値です。

割り込み位置が調整されると各サイクルで再計算されます。割り込み位置調整は時間がかかるので周波数変更が小さい場合はやらないのがお勧めです。

7.6 Start PWM と Stop PWM

Start PWM と Stop PWM ブロックは、PWM 生成器の動作を開始する、または停止する機能があります。ブロック図とパラメータは下記に示します。

シンボル：



仕様:

パラメータ	機能
PWM ソース	PWM生成器の出力の選択。 PWM1~6、3-phase PWM123と3-phase PWM456、そしてCapture 1~6が 選択できます。

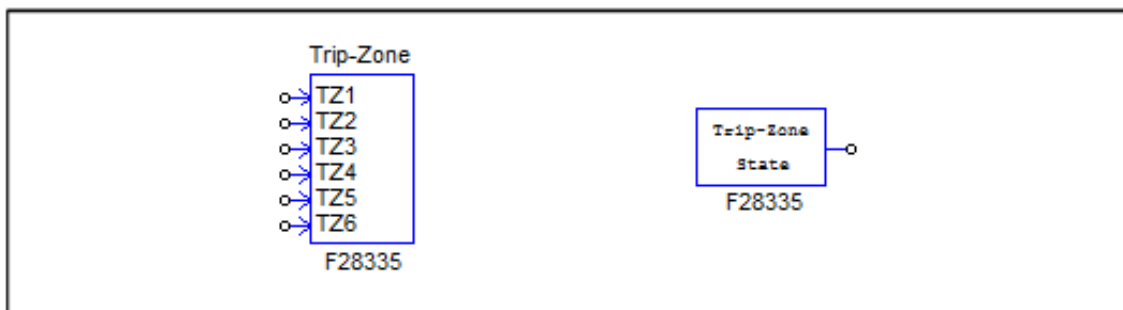
7.7 トリップゾーンとトリップゾーンステート

F2833xは6つのトリップゾーンを持っており、トリップゾーン1~6がそれぞれ GPIO12~GPIO17に対応しています。トリップゾーンは外部からの不良や問題発生信号によって動作します。各 PWM出力は対応した動作をするようにプログラムされます。

一つの PWM 生成器は 1 つから 6 つ全てまで一度に複数のトリップゾーンを使うことができます。割り込みブロックと共に使えばトリップゾーン信号は割り込み生成用に使えます。

入力信号がロー（0）になったとき、トリップゾーン信号は Trip 時動作を行います。

シンボル:



仕様（トリップゾーン）：

パラメータ	機能
Port GPIO12（トリップゾーン1）	GPIO12ポートをトリップゾーン1に設定します。
Port GPIO13（トリップゾーン2）	GPIO13ポートをトリップゾーン2に設定します。
Port GPIO14（トリップゾーン3）	GPIO14ポートをトリップゾーン3に設定します。
Port GPIO15（トリップゾーン4）	GPIO15ポートをトリップゾーン4に設定します。
Port GPIO16（トリップゾーン5）	GPIO16ポートをトリップゾーン5に設定します。
Port GPIO17（トリップゾーン6）	GPIO17ポートをトリップゾーン6に設定します。

仕様（トリップゾーンステート）：

パラメータ	機能
PWM ソース	PWM生成器の出力の選択。 - PWM1~6、3-phase PWM123と3-phase PWM456、そしてPWM1~6が選択できます。

トリップゾーン割り込みは PWM 生成器のパラメータとして One-Shot モードまたは Cycle-by-Cycle モードを設定したときに使うことができます。

Cycle-by-Cycle モードでは、割り込みはそのときの PWM 周期内のみで有効になります。また、One-Shot モードでは入力信号が(0)になったときのみ割り込みによる Trip 時動作が有効になります。割り込みにより出力が一時停止した後、PWM 生成器は再起動を開始するので原則として PWM 出力は続けられます。

PWM 生成器から割り込みが発生させられたとき、トリップゾーン State ブロックはそれが One-Shot モード、Cycle-by-Cycle モードどちらの Trip 信号なのかを出力します。（1 のとき、One-Shot モード。0 のとき、Cycle-by-Cycle モード。）

トリップゾーンと接続して割り込みブロックを使う場合は、割り込みブロックの Device Name のパラメータはトリップゾーンブロック名ではなく、PWM 生成器の名前にしてください。例えば、PWM 生成器の名前が「PWM_G1」、トリップゾーン 1 のブロック名が「TZ1」ならば、割り込みブロックの名前は「TZ1」ではなく「PWM_G1」にしてください。（この場合は割り込みブロックの Channel Number のパラメータは使用しません。）

7.8 A/D 変換器

F2833x には 12 ビット 16 チャンネルの A/D 変換器があります。A/D 変換器は Group A と Group B の二つに分かれており、DSP への回路的な入力範囲は 0~+3V までです。

一般的に A/D 変換器から DSP へは、主回路の値（電圧、電流、速度など）が取り込まれます。主回路電圧を取り込む例を挙げると、ある主回路電圧値は、最初に電圧センサで制御信号に変換されます。その後、回路信号値を DSP 入力可能な範囲（0~+3V）に変換するため、スケーリング回路があり、必要によっては DC オフセット回路も使われます。DSP 内で信号はデジタル値へ変換され、スケーリングブロックで元の入力信号範囲に復元されます。（下図参照）

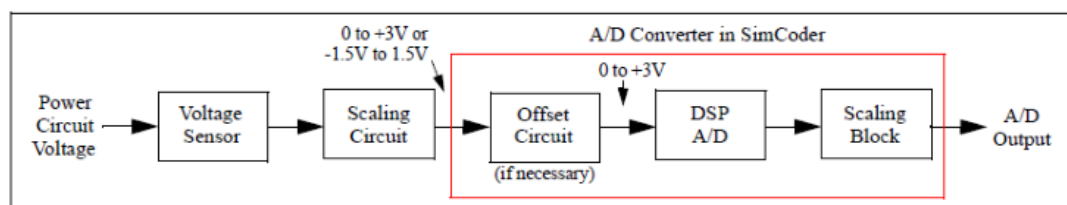


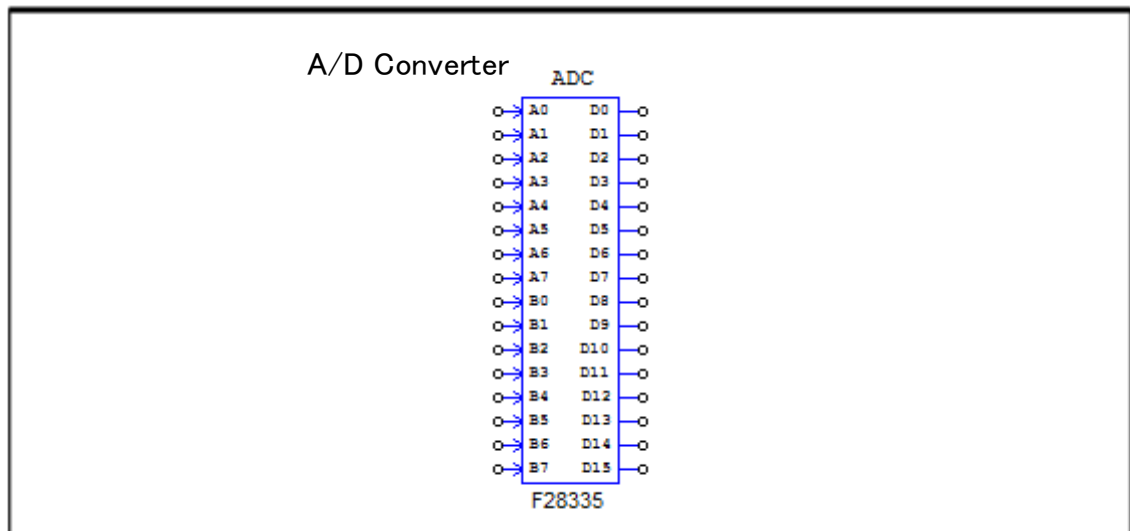
図 7.4 主回路から AD 出力までの処理

図より分かるように、SimCoder の A/D 変換器は、厳密には実際の DSP 搭載 A/D 変換器等価ではなく、オフセット機能、A/D 変換機能、スケーリング機能の組み合わせの構成を持っています。

これは AC システムアプリケーションが便利に使えるようデザインされています。

※以降は「A/D 変換器」という言葉を使うときは、DSP 搭載の A/D 変換器のことではなく、SimCoder の A/D 変換器ブロックのことを指すこととします。

シンボル:



仕様:

パラメータ	機能
ADC Mode	A/D変換器の動作モードを設定します。 - Continuous : 連続変換を行います。変換値の読み値は、最後に変換が行われた値になります。 - Start/Stop(8-Channel) : 片側のグループ（8チャンネル）のみを指令に従って一回だけ変換します。 - Start/Stop(16-Channel) : 全ての入力チャンネルを指令に従って一回だけ変換します。
Ch Ai or Bi Mode	各A/D変換チャンネルの入力モードの設定。 チャンネルにはAiとBiがあり、それぞれは0~7の値を取ります。 - AC : このオプションはシミュレーションのためのみです。-1.5Vから+1.5Vの入力が可能なac入力になります。このオプションはA/D変換へのオフセット回路を含んでいます。AC信号を0から3Vとする外部シフタが必要な場合に便利です。 - DC : 0Vから+3Vの入力が可能なdc入力になります。
Ch Ai or Bi Gain	各A/D変換チャンネルのゲインkの設定。 チャンネルにはAiとBiがあり、それぞれは0~7の値を取ります。

操作モード:

A/D 変換は Continuous モードに設定すると自律的に実行され、Start/Stop モードの場合は PWM 生成器からのトリガで実行されます。PWM を使って A/D 変換のトリガ機能をつける場合は次の制約があります。

- A/D 変換はPWM生成器のみによりトリガがかかります。すなわち複数のPWMがある場合は1つのPWM生成器を使って A/D 変換器にトリガをかけ、s

A/D 変換がトリガをかけることはできません。このグループの信号のいくつかは回路で使用されている時はできません。

これらの状態の場合は A/D 変換器は “Continuous” モードに設定しておいてください。

出力スケールリング:

出力値は下記の数式で与えられます。

$$V_o = k \times V_i$$

V_i は A/D 変換器の入力値です。

入力オフセットとスケールリング:

入力値は入力可能範囲内の値を設定してください。入力可能範囲外の値を入力した場合、出力は制限値でクランプされ、ワーニングメッセージを出力します。

A/D 変換器の入力ポートの信号は、DC 入力モードの場合は +3V が最大値としてスケールリングされ、AC 入力モードの場合は 1.5V がピークとしてスケールリングされます。

多くのアプリケーションではモニターされる回路変数は、特に AC モータードライブシステムの場合 AC 信号となります。このような AC 信号ではそれぞれオフセット回路がハードウェアの回路ボード上で 0 から 3V へ信号レベルをひき上げるためにオフセット回路が組み込まれないとなりません。

F2833xDSP の SimCoder の A/D 変換器はこのような場合に便利です。A/D 出力信号のレベルシフティングやスケールリングの代わりにオフセットポイントや SimCoder の A/D 変換器のスケールリングファクターを選択して使用できそれによってターゲットコードを生成できます。

以下に DC 入力設定時の構成例の図と AC 入力の場合の例を示しています。

A/D 変換器チャネル DC モードの例:

A/D 変換器は DC モードに設定されており、主回路電圧値が DC で 0V から 150V までの値を取るとして、実際に入力値 100V となった場合の例を取り下記に説明を行います。

センサのゲインを 0.01 だとすると実際に入力されてくる値は下記になります。

$$V_{I_max_S} = 150 \times 0.01 = 1.5V$$

$$V_{I_S} = 100 \times 0.01 = 1V$$

最大値を見たとき、センサからは 1.5V が来ますが、DSP 入力可能最大値は 3V なのでゲイン 2 を追加します。(回路としては $0.01 \times 2 = 0.02$ のゲインを持つことになります。) 調整後の入力値は下記のようになります。

$$V_{I_max_S_C} = 1.5 \times 2 = 3V$$

$$V_{I_S_C} = 1 \times 2 = 2V$$

A/D 変換後のスケールリングブロックの出力は、実際に主回路から来た電圧と合わせるとすると、A/D 変換器のゲインは 50 (電圧センサと調整用のゲインの逆数) になります。結果として A/D 変換器からの出力は下記になります。

$$V_{O_max} = 50 \times 3 = 150V$$

$$V_o = 50 \times 2 = 100V$$

回路図は以下になります。

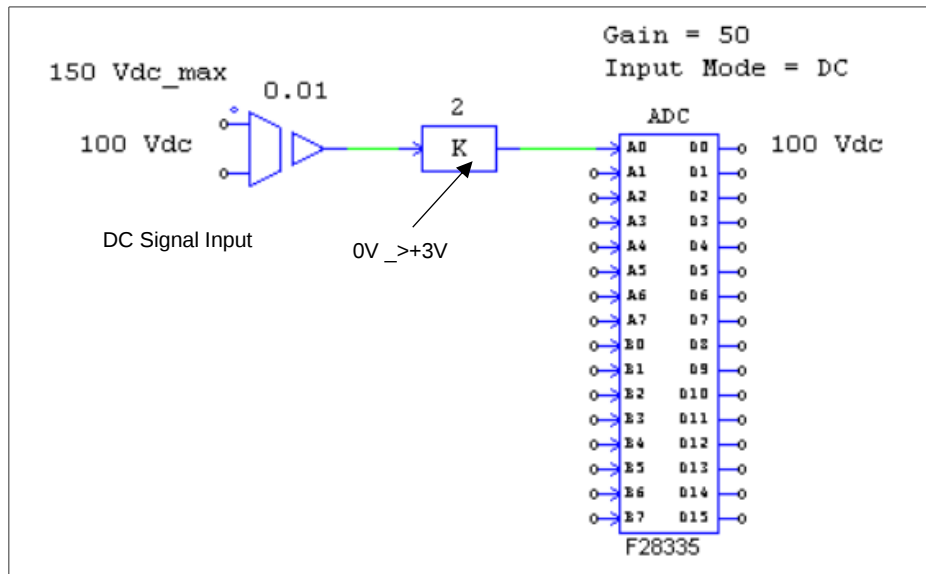


図 7.5 DC 入力場合の A/D 変換処理例

※上図では比例制御器ブロックのゲインを 2 から 1、A/D 変換器のゲインを 50 から 100 にしてもシミュレーション結果は同じ結果になりますが、コード生成としては適切に行われない可能性がある点をご注意ください。コードは+3V が入力の最大値としてスケーリングしますが、この場合、1.5V が最大値になります。入力値の最大値が+3V になるようにスケーリングするようにしてください。

AC モードの例：

A/D 変換器は AC モードに設定されており、主回路電圧値が DC で±75V までの値を取るとして、実際に入力値±50V となった場合の例を取り下記に説明を行います。

センサのゲインを 0.01 だとすると実際に入力されてくる値は下記になります。

$$V_{L_{max_S}} = +/-0.75V$$

$$V_{L_S} = +/-0.5V$$

ピーク値を見たとき、センサからは±0.75V が来ますが、DSP 入力可能ピーク値は±1.5V なのでゲイン 2 を追加します。調整後の入力値は下記のようにになります。

$$V_{L_{max_S_C}} = +/-1.5V$$

$$V_{L_S_C} = +/-1V$$

A/D 変換後のスケーリングブロックの出力は、実際に主回路から来た電圧と合わせるとすると、A/D 変換器のゲインは 50（電圧センサと調整用のゲインの逆数）になります。結果として A/D 変換器からの出力は下記になります。

$$V_{o_max} = \pm 75V$$

$$V_o = \pm 50V$$

回路図は以下になります。

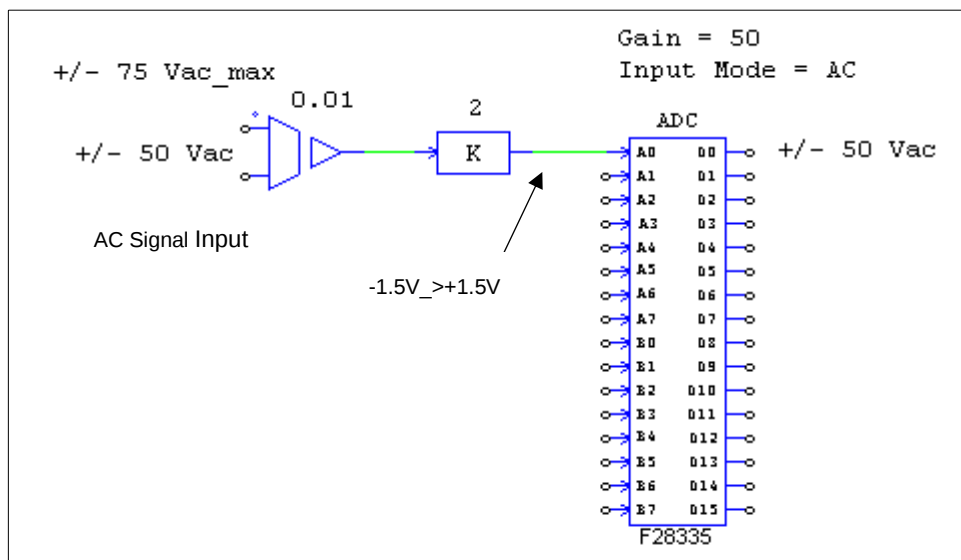


図 7.6 AC 入力場合の A/D 変換処理例

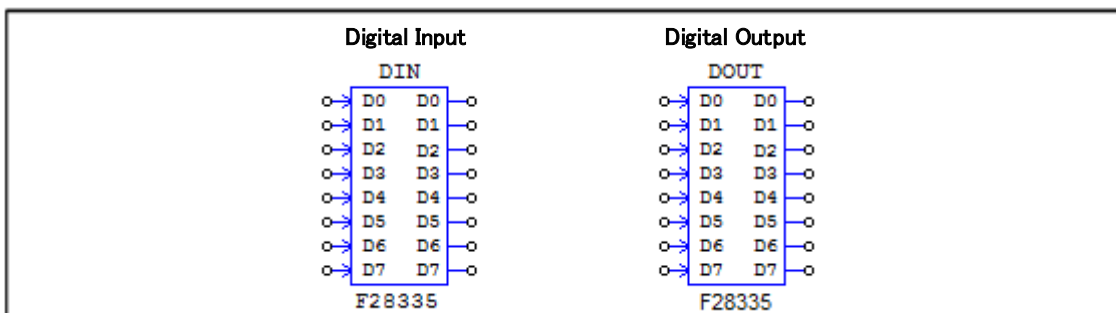
※PSIM の回路では直接 AC 信号を A/D 変換器に入力することができます。これは、ブロックの中にオフセット追加機能が含まれており、自動的に調整を行うためです。実際の DSP は 0V~3V までの入力となり、外部にオフセット回路を追加する必要がある点にご注意ください。

また、正確なコード生成を行うために A/D 変換器入力ポートのピーク値の最大値は 1.5V になるようにしてください。

7.9 デジタル入力とデジタル出力

F2833x は 88 本の汎用入出力ポート（GPIO）をデジタル入力、またはデジタル入出力に設定可能です。SimCoder では 8 チャンネルのデジタル入出力持つブロックを使うことができます。

シンボル:



仕様（デジタル入力）：

パラメータ	機能
入力ポート i の位置	ブロックのポートをGPIOピンへの設定。 iはブロックの入力で0~7です。GPIO0~87まで選択できます。
外部割込みとして 使用	外部割込み入力の設定。

仕様（デジタル出力）：

パラメータ	機能
出力ポート i の位置	ブロックのポートをGPIOピンへの設定。 iはブロックの出力で0~7です。GPIO0~87まで選択できます。

※GPIO ポートを入力ポートとした場合、そのポートは他のペリフェラルのポートとしては使えない点にご注意ください。例えば GPIO1 はデジタル入力ポートと PWM1 出力の両方の設定ができますが、同時に設定した場合、エラーが出力されます。

TMS320F2833x は最大で 7 つの外部割込み発生源を持っており、GPIO0~63 の中で使うことができます。（GPIO0~31 にて 2 つまで外部割込みを利用でき、GPIO32~63 で 5 つまで利用できます。外部割込みの優先順は GPIO0~31 の方が GPIO32~63 より高くなっています。）

7.10 UP/DOWN カウンタ

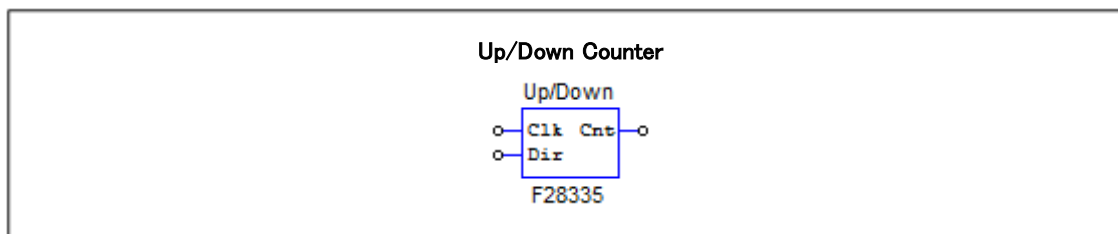
TMS320F28335 は UP/DOWN カウンタを二つ持っており、カウンタ 1 はポート GPIO20,21 または GPIO50,51 で利用でき、カウンタ 2 は GPIO24,25 で利用できます。

図内の Clk はクロック信号入力です。Dir は信号に対してカウンタをどちらへ進めるかの指令入力です。Dir が 1 のときカウンタは増加し、0 のときは減少します。

UP/DOWN カウンタブロックの出力はカウンタ値となります。

※GPIO20~21 と GPIO50~51 は同じ内部ブロックを使っているため、これらを同時に利用することはできません。

シンボル：



仕様:

パラメータ	機能
カウンターソース	カウンタとポートの設定。 - Counter1 (GPIO20, 21) : Counter1をGPIO20,21で使用します。 - Counter1 (GPIO50, 51) : Counter1をGPIO50,51で使用します。 - Counter2 (GPIO24, 25) : Counter2をGPIO24,25で使用します。

※“Clk”はクロック信号入力、“Dir”はカウンタがどちらへ進むかを指定します。“Dir”が 1 の時カウンタは増加し 0 の時は減少します。

※Clk 端子は割り振られた最初の GPIO ポート番号を使います、Dir 端子は二番目の GPIO ポート番号を使います。例えばカウンター1 を GPIO20,21 に設定した場合、Clk 端子が GPIO20 に、Dir 端子が GPIO21 に割り振られます。

※UP/DOWN カウンタブロックの出力はカウンタ値となります。

※GPIO ポートを UP/DOWN カウンタに設定した場合、そのポートはエンコーダなどのポートとしては使えない点にご注意ください。例えば、UP/DOWN カウンタ 1 とエンコーダ 1 を同時に設定した場合、エラーが出力されます。

7.11 エンコーダとエンコーダステート

F2833x はエンコーダ入力を二つ持っており、エンコーダ 1 はポート GPIO20,21 または GPIO50,51 で利用でき、エンコーダ 2 は GPIO24,25 で利用できます。

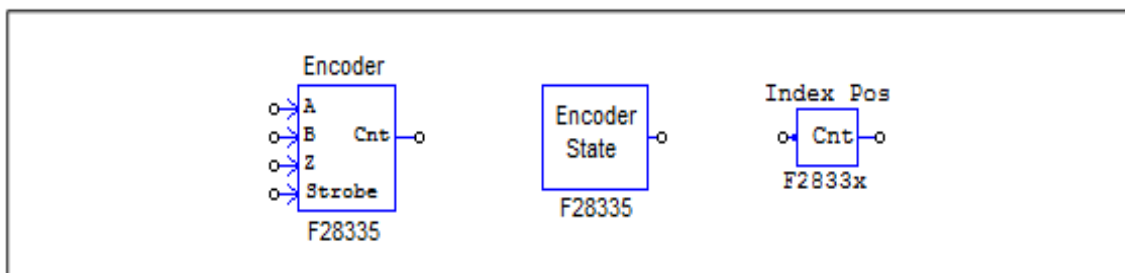
エンコーダステートブロックはどの入力信号（基準信号とストロブ信号）が割り込みを発生させる設定なのかを見ることに使います。

エンコーダステートブロックの出力が 0 のときは、Z 信号（基準信号）から割り込みが発生する設定になっていることを示し、1 のときはストロブ信号から割り込みが発生する設定になっていることを示します。

エンコーダインデックス/ストロボ位置ブロックはエンコーダの初期位置をラッチするのに使用されます。この素子の入力が 0 の時、エンコーダカウンタは 0 に設定されます。入力が 1 になるとエンコーダは、インデックス/ストロボイベントの発生を待ち、発生するとエンコーダのカウンタ値をラッチします。

※GPIO20~21 と GPIO50~51 は同じ内部ブロックを使っているため、これらを同時に利用することはできません。

シンボル:



仕様（エンコーダ）:

パラメータ	機能
エンコーダソース	エンコーダとポートの設定。 - <i>Encoder1 (GPIO20, 21)</i> : Encoder1をGPIO20,21で使用します。GPIO22はStrobe、GPIO23はIndex(Z)となります。 - <i>Encoder1 (GPIO50, 51)</i> : Encoder1をGPIO50,51で使用します。GPIO52はStrobe、GPIO53はIndex(Z)となります。 - <i>Encoder2 (GPIO24, 25)</i> : Encoder2をGPIO24,25で使用します。GPIO27はStrobe、GPIO26はIndex(Z)となります。
Z信号使用	Z信号（基準信号）の有効無効設定。 - <i>No</i>: 使用しない。 - <i>Yes(rising edge)</i>: 立上り信号エッジを使用します。 - <i>Yes(falling edge)</i>: 立下り信号エッジを使用します。
ストロブ信号使用	ストロブ信号の有効無効設定。 - <i>No</i>: 使用しない。 - <i>Yes(rising edge)</i>: 立ち上がり信号エッジを使用します。 - <i>Yes(rising/falling edge)</i>: 立ち上がり立下り両方の信号エッジを使用します。
カウンタ方向	カウンタの方向の設定。 - <i>Forward</i> : カウンタは増加方向に進みます。 - <i>Reverse</i> : カウンタは減少方向に進みます。
Z 信号極性	Z信号のトリガの極性 - <i>Active High</i> - <i>Active Low</i>
ストロブ信号極性	ストロブのトリガ極性 - <i>Active High</i> - <i>Active Low</i>
エンコーダ分解能	接続するエンコーダの分解能の設定。 4096に設定した場合、カウンタ値が4095になった次のタイミングで0にクリアされます。0に設定した場合、クリアは行われません。

仕様（エンコーダステート）:

パラメータ	機能
エンコーダソース	状態を観測するエンコーダ番号の設定。 - <i>Encoder1 (GPIO20, 21)</i> : GPIO20,21のEncoder1に対して利用します。 - <i>Encoder1 (GPIO50, 51)</i> : GPIO50,51のEncoder1に対して利用します。 - <i>Encoder2 (GPIO24, 25)</i> : GPIO24,25のEncoder2に対して利用します。

仕様（エンコーダステート）：

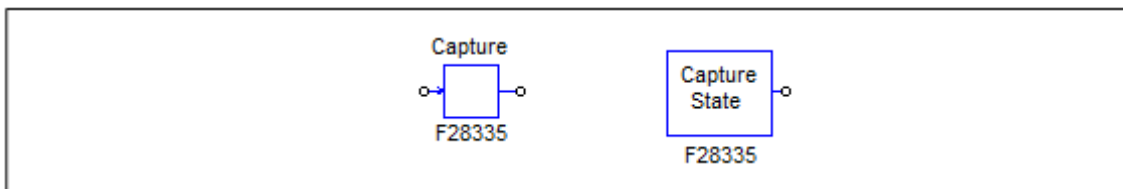
パラメータ	機能
エンコーダ ソース	状態を観測するエンコーダ番号の設定。 - Encoder1 (GPIO20, 21) : GPIO20,21のEncoder1に対して利用します。 - Encoder1 (GPIO50, 51) : GPIO50,51のEncoder1に対して利用します。 - Encoder2 (GPIO24, 25) : GPIO24,25のEncoder2に対して利用します。 Encoder1(GPIO20, 21)とEncoder1 (GPIO50, 51)が同じ内部機能ブロックで使用される場合同時に使うことはできません。
位置タイプ	ラッチカウンタタイプ - Index Pos:Z方向信号が使用されます。 - Strobe Pos:Strobe信号が使用されます。
Positopn Type	ラッチ位置 - 最初のラッチ位置 - 現在のラッチ位置

7.12 キャプチャとキャプチャステート

F2833x はキャプチャ入力を 6 つ持っており、割り込みを発生させることができます。割り込みブロックで割り込みトリガモードの設定ができます。

キャプチャステートブロックは 1 か 0 を出力し、1 のときは立ち上がりエッジを示し、0 のときは立下りエッジを示します。

シンボル:



仕様（キャプチャ）：

パラメータ	機能
キャプチャ ソース	キャプチャとポートの設定。 6つのキャプチャと14の設定可能なGPIOポートがあります。 - Capture1 (GPIO5, GPIO 24, GPIO 34) - Capture2 (GPIO7, GPIO 25, GPIO 37) - Capture3 (GPIO9, GPIO 26) - Capture4 (GPIO11, GPIO 27) - Capture5 (GPIO3, GPIO 48) - Capture6 (GPIO1, GPIO 49)
イベント フィルタ	イベントフィルタのプリスケール設定。 入力信号は選択したプリスケールで分けられます。
Timer モード	キャプチャカウンタタイマモード設定。 Absolute timeまたはTime differeneのどちらかを設定できます。

仕様（キャプチャステート）：

パラメータ	機能
キャプチャソース	状態を観測するキャプチャ番号の設定。 Capture1から6まで選択できます。

キャプチャステートブロック出力は1か0で、1は立ち上がり信号エッジ、0は立ち下がり信号エッジとなります。

7.13 シリアル通信インターフェース（SCI）

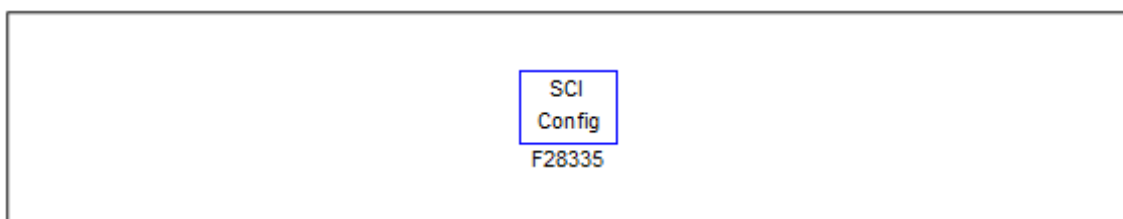
F2833x は、シリアルコミュニケーションインターフェース（SCI）のための関数が用意されています。SCI を介して、DSP 内部のデータは、外部 RS-232 ケーブルを使用してコンピュータに転送することができます。PSIM は、DSP とコンピュータ両方のデータの送受信およびコンピュータ上にデータを表示するための必要なすべての関数を提供します。これは DSP のコードをリアルタイムでモニタ、デバッグ、および調整するために非常に便利な方法を提供します。

以下に説明しますが、SimCoder では SCI 設定、SCI 入力、および SCI 出力の 3 つの SCI の機能ブロックが用意されています：

7.13.1 SCI 設定

SCI 設定ブロックは、SCI ポート、通信速度、パリティチェックのタイプ、およびデータバッファサイズを定義します。

シンボル：



仕様：

パラメータ	機能
SCI ポート選択	SCIポートの設定。SCIに使用できる7セットのGPIOポートがあります。 - SCIA (GPIO28, GPIO35) (GPIO29, GPIO36)との組み合わせ - SCIB (GPIO9, GPIO14, GPIO18, GPIO22)と(GPIO11, GPIO15, GPIO19, GPIO23)との組み合わせ - SCIC (GPIO62, GPIO63)
通信速度(bps)	SCI通信速度[bps]。プリセット速度は、200000、115200、57600、38400、19200、9600 [bps]が用意されています。または手動で他の速度を指定することができます。

パリティ チェック	通信エラーチェック用のパリティチェックの設定。 無効、奇数、偶数のいずれかを選択できます。
出力バッファ サイズ	SCI用にDSPに割り当てられたデータバッファのサイズ。バッファはRAM領域に位置しており、各バッファの要素は3つの16ビットワード（データポイントごとに、6バイト、または48ビットである）から成る1つのデータポイントを格納しています。

バッファサイズを正しく選択しなければならないことに注意してください。

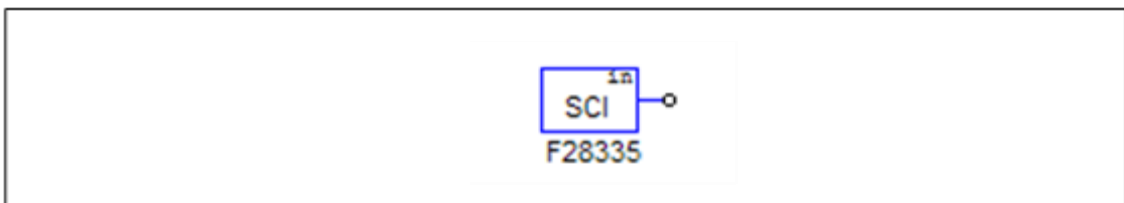
より多くの変数を長い期間にわたって監視することができるように、より多くのデータポイントを収集するためには大きなバッファが好ましい。一方、内部 DSP のメモリは限られており、バッファは通常の DSP の動作を妨げるほど大きすぎはいけません。バッファサイズを選択する方法の詳細については、"Tutorial - Using SCI for Waveform Monitoring.pdf"を参照してください。

7.13.2 SCI 入力

SCI 入力ブロックは、ハングアップしうる DSP コードで変数を定義するために使用されます。SCI 入力変数の名前は、DSP のオシロスコープ（ユーティリティのメニューの下）に表示され、その値は SCI を介して実行時に変更することができます。

SCI 入力ブロックは、例えば、指令値の変更、コントローラパラメータの微調整のための便利な方法を提供します。

シンボル:



仕様:

パラメータ	機能
初期値	SCI入力変数の初期値。

回路図では、SCI 入力は定数として動作。コードが DSP 上で実行され、実行時にハングアップしうる間、値はシミュレーションの初期値に固定されます。

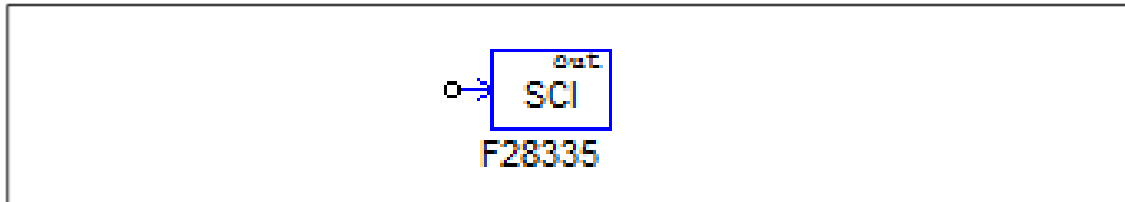
7.13.3 SCI 出力

SCI 出力ブロックは、表示用の変数を定義するために使用されます。

SCI 出力ブロックがノードに接続されている場合、SCI 出力ブロックの名前は、DSP のオシロスコープ（ユーティリティのメニューの下）に表示され、この変数のデータは、実行時に SCI を介してコンピュータに DSP から送信することができ、波形は DSP オシロスコープで表示することができます。

SCI の出力ブロックは、DSP 波形を観測するための便利な方法を提供します。

シンボル:



仕様:

パラメータ	機能
データポイント ステップ	データの収集頻度を定義します。[Data Point Step]が1の場合、すべてのデータポイントが収集され送信されます。例えば、[Data Point Step]が10の場合、10ポイントのうちの1ポイントのみが収集され送信されます。

[Data Point Step]が小さすぎると、データポイントが非常に多くなる可能性があり、すべてを送信できない場合があることに注意してください。この場合、いくつかのデータポイントがデータの送信時に破棄されます。

また、[Data Point Step]のパラメータは DSP オシロスコープが連続モードの時のみ使用されます。スナップショットノードにある場合、このパラメータは無視され、すべてのポイントを収集して送信されます。

シミュレーションでは、SCI 出力は電圧プローブとして動作します。

7.14 シリアルペリフェラルインターフェース(SPI)

F2833x DSP はシリアルペリフェラルインターフェース (SPI) のための関数が用意されています。TI F833x ターゲットライブラリの SPI ブロックによって、容易かつ便利に外部の SPI デバイス（外付け A/D および D/A コンバータなど）と通信する機能を実装することができます。SPI デバイスのために手動でコードを書くことは、しばしば時間がかかり、大変な作業です。SPI をサポートする機能によって PSIM は大幅に簡素化し、コーディングとハードウェアの実装プロセスをスピードアップします。

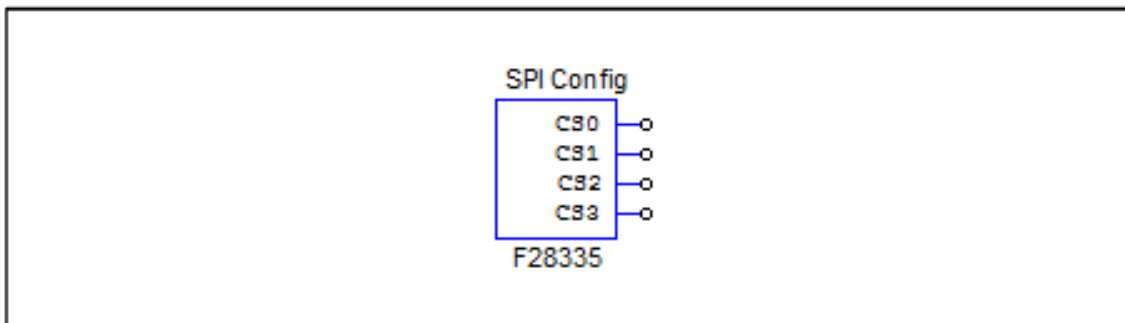
SPI ブロックを使用する方法についての詳細な説明については、ドキュメントを参照してください" *Tutorial - Using SCI for Real-Time Monitoring.pdf* "。

SimCoder では SPI 設定、SPI デバイス、SPI 入力、SPI 出力の 4 つの SCI の機能ブロックが提供されています。

7.14.1 SPI 設定

SPI 設定ブロックは、SPI ポート、チップ選択ピン、および SPI のバッファサイズを定義します。SPI が使用されている回路図に必要で、ブロックはメイン回路図内に配置される必要があります。

シンボル:



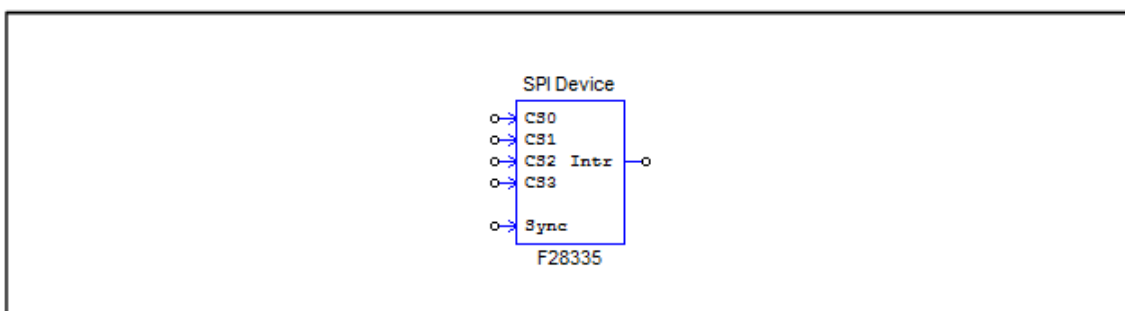
仕様:

パラメータ	機能
SPI port	SPIポートの設定。SPIポートはGPIO16-19またはGPIO54-57があります。
Chip Select Pin 0,1,2,3	チップセレクトピンのGPIOポート。PSIMは、チップセレクトピンPin0～Pin3で定義されている4つのGPIOピンを選択する必要がある16 SPIデバイスに対応しています。これらのGPIOポートとSPIスレーブ送信許可ピンSPISTEは、チップセレクト信号を生成するために使用されます。
SPI Buffer Size	SPIコマンドのバッファサイズ。バッファの各メモリセルは、SPIコマンドのインデックスが保存されます。通常は、すべてのSPIの入力/出力要素の中でSPIコマンド（例えば、Start Conversion Command, Receiving Data Command, Sending Data Command, and Sync. Command）の数に1を足したバッファサイズを指定します。

7.14.2 SPI デバイス

SPI デバイスのブロックは、対応する SPI ハードウェアデバイスの情報を定義します。回路図の SPI デバイスのブロック数は、SPI ハードウェアデバイスの数と同じでなければなりません。

シンボル:



仕様:

パラメータ	機能
Chip Select Pins	SPIデバイスに対応するチップセレクトピンの状態。チップセレクトピンがこの状態にあるときは、このSPIデバイスが選択されます。
Communication	SPIの通信速度（MHz）

Speed (MHz)	
Clock Type	SPIハードウェアデバイスによって決定されるSPIクロックの種類。 -立ち上がりエッジ（遅れなし）： クロックは通常はLOWで、データはクロックの立ち上がりエッジでラッチされます。 -立ち上がりエッジ（遅れあり）： クロックは通常はLOWでデータは遅延付きのクロックの立ち上がりエッジでラッチされます。 -立ち下がりエッジ（遅れなし）： クロックは通常はHIでデータはクロックの立ち下がりエッジでラッチされます。 -立ち下がりエッジ（遅れあり）： クロックは通常はHIでデータは遅延付きのクロックの立ち下がりエッジでラッチされます。
Command Word Length	SPI通信コマンドのワード長、または有効ビットの長さ。1～16ビットにすることができます。
Sync. Active Mode	SPIデバイスの同期信号のトリガモード。立ち上がりエッジまたは立ち下がりエッジのいずれかです。
SPI Initial Command	SPIデバイスの初期コマンドです。
Hardware Interrupt Mode	SPIデバイスが生成する割り込み信号の種類を指定します。SPIデバイスの割り込み出力ノードがデジタル出力要素の入力に接続されている場合にのみ有効です。次のいずれかになります。 - ハードウェア割り込みなし - 立ち上がりエッジ - 立ち下がりエッジ
Interrupt Timing	変換完了時のSPIデバイスの割り込み生成方法を指定します。次のいずれかになります。 - 割り込みなし 割り込みは生成されません。この場合、DSPIはSPI入力デバイスにコマンドを送信します。デバイスは変換を開始して同じコマンドで結果を返します - 連続多重割り込み： 多重割り込みは、各変換の後に連続で生成されます。 1つのA/D変換ユニットと複数の入力チャンネルを持つSPIデバイス用です。この場合、DSPが最初に変換のコマンドを送信し、SPIデバイスの変換が開始されます。変換が完了すると、SPIデバイスが割り込みを生成します。割り込みサービスルーチンでは、DSPが変換結果をフェッチするためにコマンドを送信し、同じSPI入力デバイスの別のチャンネルの新しい変換を開始します。 - ワンタイム割り込み： 変換終了後に1回だけ割り込みが生成されます。1つのリクエストで複数のチャンネルの変換を実行可能なSPIデバイス用です。この場合、DSPIはSPI入力デバイスにコマンドを送信し、SPIデバイスは複数の入力チャンネルの変換を完了します。すべての変換が完了すると、SPIデバイスが割り込みを生成します。
Command Gaps (ns)	2つのSPIコマンドの間隔[nsec]です。
Conversion Sequence	変換シーケンスを決定するためのコマンドで区切られたSPIの入力要素の名前を定義します。このパラメータは、SPIデバイスが連続で複数の割り込みを生成する場合にのみ有効であることに注意してください。

回路図では、チップセレクトロジックが実装される方法を定義せずに、すべての SPI デバイスのチップセレクトピンは SPI 設定ブロックのチップセレクト端子に接続されています。

しかし、実際のハードウェアでは、対応するチップセレクトロジックに合わせて実装する必要があります。SPI のコマンドは、カンマで区切られた 16 ビットの一連の数字で構成されています。16 ビットの数値では、下位ビットだけがコマンドによって使用される有効ビットです。例えば、コマンドワード長が 8 の場合、ビット 0～7 はコマンドであり、ビット 8～15 は使用されません。

SPI デバイスは、入力デバイスまたは出力デバイスのいずれかです。

例えば、外部 A/D コンバータは、入力デバイスです。通常、DSP はデバイスに 1 つまたは複数の A/D 変換のコマンドを送信し、変換を開始するために同期信号を設定します。同期信号は、同じデバイスの次回のコマンドでリセットされます。

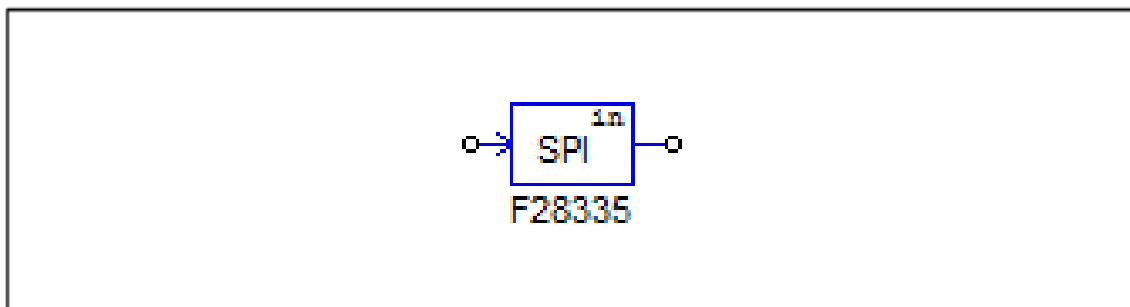
同期信号を使用した SPI の入力デバイスは、通常、割り込みサービスルーチンを入力して DSP をトリガする割り込みピンを必要とします。

一方、外部 D/A コンバータは出力デバイスです。通常、DSP はデバイスに 1 つまたは複数の D/A 変換のコマンドを送信し、変換を開始するために同期信号を設定します。同期信号は、同じデバイスの次回のコマンドでリセットされます。

7.14.3 SPI 入力

SPI 入力デバイスは、複数の入力チャンネルを持つことができます。SPI 入力ブロックは、SPI 通信用の入力チャンネル、および 1 つの入力チャンネルに対応する 1 つの SPI 入力ブロックのプロパティを定義するために使用されます。

シンボル:



仕様:

パラメータ	機能
Device Name	SPI入力デバイスの名前です。
Start Conversion Command	コンマで区切られた16進の数字で、変換を開始するコマンドです。 (例 ; 0x23、0x43、0x00)
Receiving Data Command	コンマで区切られた16進の数字で、受信を開始するコマンドです。 (例 ; 0x23、0x43、0x00)
Data Bit Position	データビットが受信データの文字列のどこにいるか定義します。形式は次のとおりです。 <i>ElementName</i> = { <i>Xn</i> [<i>MSB.. LSB</i>]} ここで、 - “ <i>ElementName</i> ”はSPIの入力デバイスの名前です。もし現在のSPI入力デバイスの場合は、代わりに <i>y</i> を使用してください。 - {}は、括弧内の項目が複数回繰り返されることを意味します。 - <i>Xn</i> はSPIの入力装置から受信した <i>n</i> 番目の単語であり、 <i>n</i> は0から始まります。

	- MSB.. LSBは、ワードの有効ビットの位置を定義します。
Input Range	入力範囲を定義するパラメータVmaxを指定します。このパラメータは、SPIデバイスがA/Dコンバータの場合にのみ有効です。デバイスの変換モードがDCの場合、入力範囲は0～Vmax。デバイスの変換モードがACの場合、入力範囲は-Vmax/2～Vmax/2。
Scale Factor	出力スケールファクタのKscale。スケールファクタが0の場合、SPIデバイスは、A/Dコンバータではなく、結果はDSPがSPI通信で受信したものとまったく同じになります。そうでない場合は、SPIデバイスはA/Dコンバータであり、結果はこの係数とA/D変換モードに基づいてスケーリングされます。
ADC Mode	デバイスのA/D変換モード。変換モードはDCまたはACのどちらでもかまいません。このパラメータはデバイスがA/Dコンバータの場合にのみ有効であることに注意してください。
Initial Value	入力の初期値です。

データのビット位置の式は、SPI 入力デバイスのデータ長を定義します。例えば、

y= x1[3..0]x2[7..0]は、データ長が 12 であることを意味し、結果は 2 ワード目の下位 4 ビットと 3 番目のワードの下位 8 ビットです。受信データの文字列は 0x12、0x78、0xAF の場合、結果は 0x8AF です。

スケールファクタが 0 でない場合、出力は以下に基づいて拡大縮小されます。

DC 変換モードの時：

-シミュレーションの場合

$$Output = Input \cdot K_{scale}$$

-実際のハードの場合

$$Output = \frac{Result \cdot V_{max} \cdot K_{scale}}{2^{Data_Length}}$$

AC 変換モードの時：

-シミュレーションの場合

$$Output = Input \cdot K_{scale}$$

-実際のハードの場合

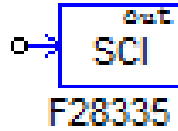
$$Output = \frac{(Result - 2^{Data_Length-1}) \cdot V_{max} \cdot K_{scale}}{2^{Data_Length-1}}$$

パラメータ Data_length はデータのビット位置の計算式で計算されます。

7.14.4 SPI 出力

SPI 出力デバイスは、複数の出力チャンネルを持つものもあります。SPI 出力ブロックは、SPI 通信用の出力チャンネル、1 つの出力チャンネルに対応する 1 つの SPI 出力ブロックのプロパティを定義するために使用されます。

シンボル:



仕様:

パラメータ	機能
Device Name	SPI出力デバイスの名前です。
Scale Factor	出力スケールファクタのKscale。スケールファクタが0の場合、SPIデバイスは、D/Aコンバータではなく、結果はDSPがSPI通信で受信したものとまったく同じになります。そうでない場合は、SPIデバイスはD/Aコンバータであり、結果はこの係数とD/A変換モードに基づいてスケーリングされます。
Output Range	出力範囲を定義するパラメータVmaxを指定します。このパラメータは、SPIデバイスがD / Aコンバータの場合にのみ有効です。デバイスの変換モードがDCの場合、入力範囲は0～Vmax、デバイスの変換モードがACの場合、入力範囲は-Vmax/2～Vmax/ 2。
DAC Mode	デバイスのD/ A変換モード。変換モードはDCまたはACのどちらでもかまいません。このパラメータはデバイスがD / Aコンバータの場合にのみ有効であることに注意してください。
Sending Data Command	コンマで区切られた16進の数字で、出力データを送信するコマンドです。 (例 ; 0x23, 0x43, 0x00)
Data Bit Position	データビットが送信データの文字列のどこにいるか定義します。形式は次のとおりです。 <i>ElementName</i> = { <i>Xn</i> [<i>MSB.. LSB</i>]} ここで、 - “ <i>ElementName</i> ”はSPI出力デバイスの名前です。もし現在のSPI出力デバイスの場合は、代わりに <i>y</i> を使用してください。 - {}は、括弧内の項目が複数回繰り返されることを意味します。 - <i>Xn</i> はSPI出力デバイスに送信した <i>n</i> 番目の単語であり、 <i>n</i> は0から始まります。 - <i>MSB.. LSB</i> は、ワードの有効ビットの位置を定義します。
Sync. Command	コンマで区切られた16進の数字で、SPI出力デバイスの出力チャネルと同期させるコマンドです。(例 ; 0x23, 0x43, 0x00) このコマンドは、SPI出力デバイスが同期信号を持っていない場合に使用されます。

データのビット位置の式は、SPI 出力デバイスのデータ長を定義します。例えば、

$y=x1[3..0]x2[7..0]$ は、データ長が 12 であることを意味し、結果は 2 ワード目の下位 4 ビットと 3 番目のワードの下位 8 ビットです。送信データの文字列が 0x12, 0x78, 0xAF の場合、データは 0x8AF です。

スケールファクタが 0 でない場合、出力は以下に基づいて拡大縮小されます。

DC 変換モードの時 :

-シミュレーションの場合 $Output = Input \cdot K_{scale}$

-実際のハードの場合 $Output = \frac{Result \cdot K_{scale} \cdot 2^{Data_Length}}{V_{max}}$

AC 変換モードの時：

-シミュレーションの場合 $Output = Input \cdot K_{scale}$

-実際のハードの場合 $Output = 2^{Data_Length} + \frac{Result \cdot K_{scale} \cdot 2^{Data_Length-1}}{V_{max}}$

パラメータ Data_length はデータのビット位置の計算式で計算されます。

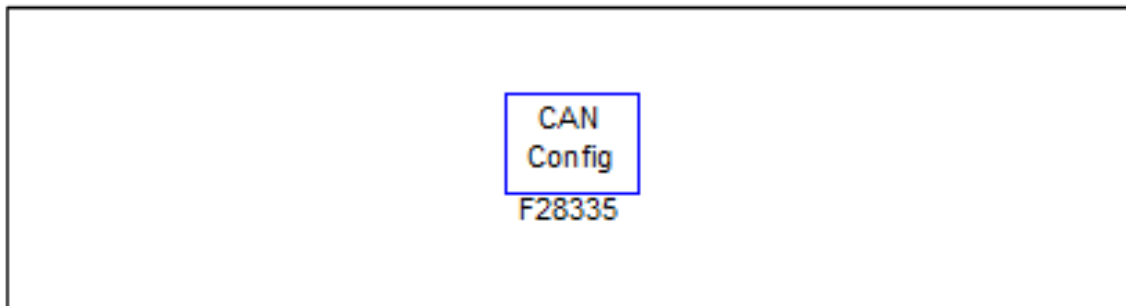
7.15 Controller Area Network (CAN) Bus

F2833x は CAN バスコミュニケーションの機能をもっています。PSIM では CAN バスで必要な機能を備えています。SimCoder には *CAN Configuration*、*CAN Input*、*CAN output* の 3 つの機能があります。

7.15.1 CAN Configuration

CAN Configuration ブロックは CAN バスソースでデータバイト順や他の設定を定義します。

シンボル:



仕様:

パラメータ	機能
CAN 速度 (MHz)	通信速度 (単位はMHz)。 次の周波数、125kHz、250kHz、500kHz、1MHzのどれか1つを選択するか もしくは手入力で設定できます。手入力の場合は1MHzを超えないようご注意ください。
Dataバイト順 Order	データバイトの順番になります。次の 1 つとなります。 - Least significant byte 1st - Most significant byte 1st

	“Least significant byte 1st” は最下位バイトが最初に転送され最上位バイトが最後に転送されます。一方 “Most significant byte 1st”は最上位バイトが最初に転送され最下位バイトが最後に転送されます。
入力メールボックス数	入力mailboxの数
受信メール紛失検査	もしEnable(有効)に設定されると受信メールが無くなった場合、エラーレポート関数の “_ProcCanAErrReport(nErr)” (CAN Aに対して) が呼びだされます。 この関数はエラー状態nErrの数値を返します。 もし、Disable(無効)に設定されるとエラーレポート関数は呼ばれません。
バスオフ検査	もしEnable(有効)に設定されると受信メールが無くなった場合、エラーレポート関数の “_ProcCanAErrReport(nErr)” (CAN Aに対して) が呼びだされます。 この関数はエラー状態nErrの数値を返します。 もし、Disable(無効)に設定されるとエラーレポート関数は呼ばれません。
エラー検査モード	エラーチェックモードを受動か能動かのどちらかを選択できます。受動の場合、エラーカウンタが128になったときに割込みが発生します。能動の場合割り込みが毎回発生します。

関数 “_ProcCanAErrReport(nErr)” (CAN A に対して) の戻り状態 nErr は 32 ビットの整数です。

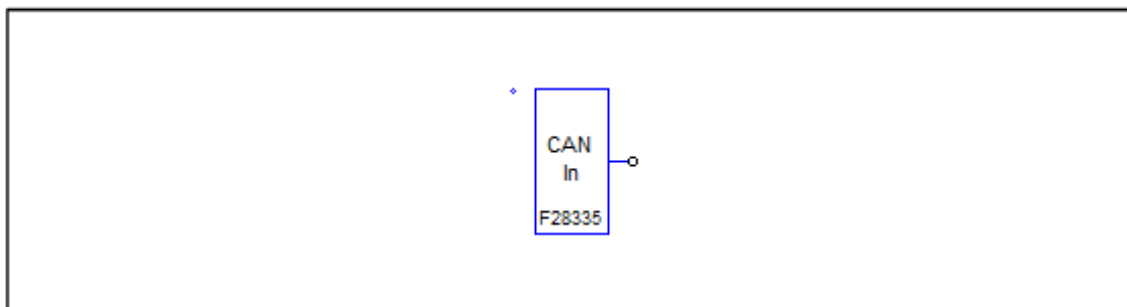
エラーとレジスタCANES 状態からの値を含んでいます。関数 “_ProcCanAErrReport(nErr)”からの値が戻るとレジスタCANES はクリアされます。

もし、特定のエラーに対応したい場合は “_ProcCanAErrReport(nErr)”関数に独自のコードを追加することができます。

7.15.2 CAN Input

CAN 入力ブロックは CAN バスから CAN メッセージを受け取ります。

シンボル:



仕様:

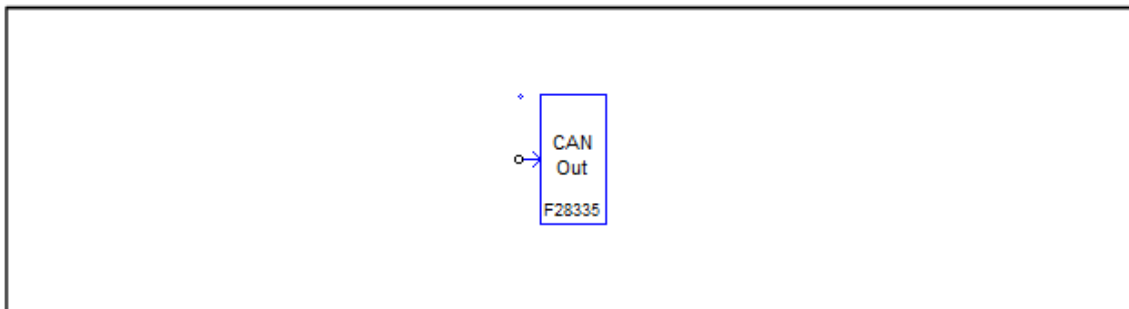
パラメータ	機能
Number of Input	CAN入力数です。8入力まで可能です。
CAN ソース	CANソースはCAN AかCAN Bとなります。
拡張ID使用	Yesに設定された場合、メッセージのIDは29-bitの整数となります。 Noに設定された場合、メッセージのIDは11-bitの整数となります。
メッセージ ID	メッセージのIDです。整数で例えば0x23のようになります。
ローカルマスク	メッセージに対するMASKとなります。整数です。もしメッセージIDのビットがMASKのビットと一致するとメッセージを受信します。そうでない場合は無視され

	ます。例えばmaskが0x380でメッセージIDが0x389の場合、このメッセージが受信されます。もし、メッセージIDが0x480の場合はメッセージは無視されます。
上書き保護フラグ	Allow overwriteかDo not allow overwriteに設定できます。 例えばメッセージ受信用にメールボックスが設定され、新しいメッセージがCANバス経由で受信されたが古いメッセージがまだメールボックスにあり、何も処理されていない場合、Override flagが"Allow overwrite"に設定されると新しいメッセージが受信され古いメッセージは上書きされます。Flagが"Do not allow overwrite"に設定されていると新しいメッセージは受信されずに古いものはそのまま残ります。
受信頻度	特定期間に入力ブロックで受信したメッセージの数。例えば、最初の入力ブロックで1秒間に20のメッセージを、2番目の入力ブロックで1秒間に30のメッセージを受信する2つの入力ブロックがあるとします。その場合"Receive Message Rate"は最初のブロックに20、2番目のブロックに30と設定されます。
入力 <i>i</i> Gain	<i>i</i> 番目の入力に対するゲインで <i>i</i> は1から8で設定できます。出力は入力のゲインの値をかけた値になります。
入力 <i>i</i> データ開始位置	メッセージは8データポイントまで可能です。データポイントは1bitから32bitsまで可能です。これはメッセージの現在のデータポイントの開始位置を定義します。
入力 <i>i</i> データ終了位置	これはメッセージの現在のデータポイントの終了位置を定義します。
入力 <i>i</i> データタイプ	データタイプは、浮動、整数、またはIQ1からIQ30までとなります。
入力 <i>i</i> デフォルトデータ	SCI入力変数の初期値となります。

7.15.3 CAN Output

CAN 出力ブロックは CAN メッセージを CAN バスへ送ります。

シンボル:



仕様:

パラメータ	機能
Number of Outputs	CAN出力の数です。8出力まで可能です。
CAN ソース	CANソースはCAN AかCAN Bとなります。
拡張ID使用	Yesに設定された場合、メッセージのIDは29-bitの整数となります。 Noに設定された場合、メッセージのIDは11-bitの整数となります。
メッセージ ID	メッセージのIDです。整数で例えば0x23のようになります。
トリガタイプ	トリガのタイプは次のようになります。 - No trigger : トリガなし。 - Rising edge : トリガソースの立ち上がりエッジでトリガがかかります。

	- <i>Falling edge</i> : トリガソースの立下りエッジでトリガがかかります。 - <i>Riding/Falling edge</i>:トリガソースの立ち上がりと立下りのエッジでトリガがかかります。 立ち上がり／立下りエッジはトリガソースの現在の値と、最終のトリガの値が1以上で差がある場合に適用されます。
トリガソース	トリガソースはトリガタイプによるグローバル変数の定数にもあります。 トリガタイプが “No Trigger”に設定されるとトリガソースはカウンターリミットを定義します。例えばトリガソースが定数かグローバル変数で値が5の時、5回に1回トリガがかかることを意味しています。すなわちデータが5サイクル毎に送られます。 もしトリガタイプがエッジトリガに設定された場合、トリガソースはグローバル変数のみです。グローバル変数がトリガタイプに依存して立ち上がりか立下りもしくは両方のエッジとなる時にトリガがかかります。
出力 <i>i</i> ゲイン	<i>i</i> 番目の出力ゲインでは1から8で設定できます。出力は入力のゲインの値をかけた値になります。
出力 <i>i</i> データ開始位置	メッセージは8個のデータポイントまで可能です。データポイントは1bitから32bitsまで可能です。これはメッセージの現在のデータポイントの開始位置を定義します。
出力 <i>i</i> データ終了位置	これはメッセージの現在のデータポイントの終了位置を定義します。
Output <i>i</i> Data Type	データタイプは、浮動、整数、またはIQ1からIQ30までとなります。

7.16 割込み時間

割込み時間ブロックは割込みサービスルーチンの時間間隔を測定するのに使われます。

仕様:

パラメータ	機能
Time Output Method	割込み時間測定の定義を行います。 -SCI(time used : 使用時間): SCIを使用します。DSPクロックカウントで割込みサービスルーチンとして使われる時間はSCIを介して送信されます。 -SCI(time remaining : 残り時間): SCIを使用します。割込みサービスルーチンの残り時間です。DSPクロック数で測定されSCIを介して送信します。time remainingは現在の最後の割込みから次の割込みの先頭までの時間として定義されます。 -GPIO0 to GPIO87: GPIOポートを使います。パルスは特定されたGPIOポートで生成されます。パルスは割込みに入る時high(1)にセットされます。割込みが終了する時にはlow(0)にセットされます。オシロスコープでパルス幅を測定できます。
Sampling Frequency	割込みサービスルーチンのサンプリング周波数 (単位はHz)

SCI が使われる場合、値は DSP クロックでカウントされます。例えば値が 7500 の時 150MHz の DSP クロックの割込み時間は $7500/150M=50 \mu s$ となります。

7.17 プロジェクト設定とメモリ配置

TI F2833x Hardware Target でコード生成をしようとしたとき、SimCoder は開発環境 TI Code Composer Studio(CCS)のプロジェクトファイルも作成できますので、コンパイル、リンク、DSP へのアップロードが容易に行えます。

現在、Code Composer Studio のバージョン 3.3 に対応しています。PSIM の回路ファイル名が「test.sch」である場合、コード生成後、「test(C code)」というフォルダが回路ファイルと同じ場所に作成されます。作成されたフォルダには下記のファイルが生成されています。

- test.c	: 生成された C コード
- PS_bios.h	: SimCoder F2833x のヘッダファイル
- passwords.asm	: DSP コードパスワードファイル
- test.pjt	: CCS プロジェクトファイル
- F2833x_Headers_nonBIOS.cmd	: ペリフェラルレジスタリンクコマンドファイル
- F28335_FLASH_Lnk.cmd	: フラッシュメモリリンクコマンドファイル
- F28335_FLASH_RAM_Lnk.cmd	: フラッシュ RAM リンカコマンドファイル
- F28335_RAM_Lnk.cmd	: RAM メモリリンクコマンドファイル

注): リンカコマンドファイルの名前は F28335 でない場合はターゲットハードウェアによりアサインされます。例えばもしターゲットハードウェアが F28334 の場合、ファイル名はそれに応じて F28334_FLASH_Lnk.cmd, F28334_FLASH_RAMLnk.cmd, F28334RAM Lnk.cmd となります。

更に下記のファイルが必要となります。

- PS_bios.lib	: PSIM フォルダ内の SimCoder F2833x ライブラリ
- C28x_FPU_FastRTS_beta1.lib	: PSIM の¥lib フォルダ内の TI 高速浮動小数点ライブラリ

これら二つのファイルはコード生成時に自動的にプロジェクトフォルダにコピーされます。

※.c ファイルと.pjt ファイルはコード生成実行時に毎回生成及び書き換えが行われます。もしこれらのファイルを自分で別途変更を加えたい場合は、まずこれらのファイルを別のフォルダにコピーしてから行ってください。次回コード生成実行時に、強制的に上書きが行われ、手動での変更が失われる可能性があります。

プロジェクト設定:

CCS プロジェクトファイルには 4 つの設定があります。

- RAM Debug	: デバッグモードでコンパイル RAM メモリへ書き込みの設定
- RAM Release	: リリースモードでコンパイル RAM メモリへ書き込みの設定
- Flash Release	: リリースモードでコンパイルフラッシュメモリへ書き込みの設定
- Flash RAM Release	: デバッグモードでコンパイル RAM メモリへ書き込みの設定

RAM Debug または RAM Release 設定が選択されたとき、CCS はリンクコマンドファイル F28335_RAM_Lnk.cmd でプログラムとデータ空間の配置を行います。

Flash Release 設定が選択されたとき、CCS はリンクコマンドファイル F28335_FLASH_Lnk.cmd でプログラムとデータ空間の配置を行います。

Flash RAM Release 設定が選択されたとき、CCS はリンカコマンドファイル F28335_FLASH_RAM_Lnk.cmd でプログラムとデータ空間の配置を行います。メモリ配置は RAM Release と同じです。

リリースモードでのコンパイルはデバッグモードに比べると早く済みます。また RAM Release または Flash RAM Release が最も早いです。RAM Debug は遅く、Flash Release が最も遅いですが、開発段階では一般的にデバッグのし易さから RAM Debug から始めて、RAM Release へそして Flash Release や RAM Release へ移行していきます。

メモリ配置：

生成されたリンカファイルではメモリ配置は下記のように定義されます。

RAM Debug、RAM Release、Flash RAM Release の設定の場合：

RAM Memory 0x0000_0x07FF(2K) Interrupt vectors Stack 0x8000_0xFFFF(32K*) Program and data space

Flash Release の設定の場合：

RAM Memory 0x0000_0x07FF(2K) Interrupt vectors Stack 0x8000-0xFFFF(32K*) data space	Flash Memory 0x300000_0x33FFFF(256K**) Program Password etc.
---	---

注意：

* RAM Debug や RAM Release では、SimCoder はプログラム空間やデータ空間を RAM メモリの 0x8000 から 0xFFFF まで定義します。

- F28335,F28334 : 0x8000 から 0xFFFF(32K)
- F28332 : 0x8000 から 0xDFFF(16K)

もしプログラムとデータ空間を合わせたものが RAM 空間を越えた場合は Flash Release 設定を選択する必要があります。

* * SimCoder によりあらかじめ定義されているプログラム空間に対する flash memory は

- F28335 : 0x300000 から 0x33FFFF(256K)
- F28334 : 0x320000 から 0x33FFFF(128K)
- F28332 : 0x330000 から 0x33FFFF(64K)

となります。

8 TI F2803x Hardware Target

8.1 概要

オプションモジュールTI F2803x Hardware TargetとSimCoderの組み合わせによって、PSIMの制御回路図からTexas Instruments社製固定小数点型DSP F2803xを搭載した基板で汎用的に使用できるCコードを生成することができます。

TI F2803xハードウェアターゲットはF2803x全てのパッケージに対応しています。

TI F2803x Hardware Targetには下記の機能があります。

- 3-phase、2-phase、1-phase、とAPWM(Single PWM)生成機能
- 可変周波数PWM
- PWM生成のStart、Stop機能
- トリップゾーン、トリップゾーンステート機能
- A/D変換機能
- コンパレータ入力、出力とDAC
- デジタル入力出力機能
- Up/Downカウンタ機能
- エンコーダ、エンコーダステート機能
- キャプチャ、キャプチャステート機能
- SCI基板設定機能、入力出力
- SPI基板設定機能、デバイス、入力出力
- CAN設定機能、入力出力
- 割り込み時間機能
- DSP クロック機能
- ハードウェア基板設定機能

複数のサンプリング周波数でコードを生成する回路の場合、SimCoder は優先的に PWM 割り込みの周波数に使用します。他のサンプリング周波数では、タイマ 1 割り込み、タイマ 2 割り込みの順に使われます。3 つ以上のサンプリング周期がある場合は対応する割り込み関数をメイン関数内で実行する形にします。

TI F2803x では PWM 生成部からハードウェア割り込みを発生させることができますので、SimCoder は PWM 生成ブロックに繋がっていて、PWM 生成ブロックを同じ周波数を持つ全ての素子を検索してグループ化します。各素子ブロックは自動的に配置され、出力コード内の割り込み関数として実行されます。

また、ハードウェア割り込みはデジタル入力、エンコーダ、キャプチャ、そしてトリップゾーンにもあります。各ハードウェア割り込みは割り込みブロックから生成され、(5.4 章参照) 割り込みブロックは割り込み関数と繋げる必要があります。(割り込み関数はサブ回路で作ります。) 二つ以上の割り込みを使うと

きはそれぞれの機能に割り込みブロックと割り込み関数を設定します。

本章では TI F2803x Hardware Target ライブラリ内の各素子の使い方について解説します。次の図は F2803x 80-pin PN QFP ポートアサイン解説します。

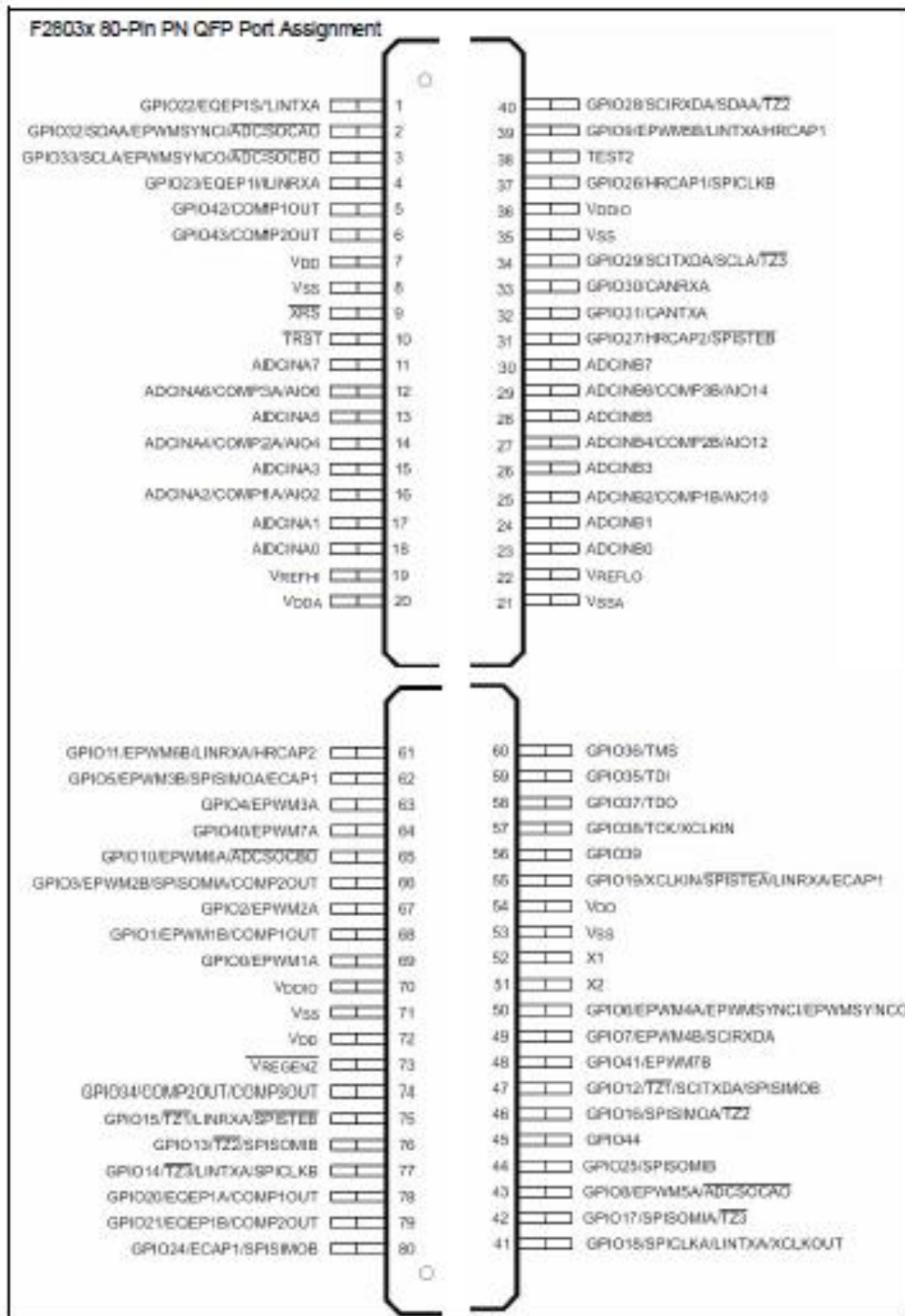


図 8- 1 F2803x 80-pin PN QFP ポートアサイン図

8.2 ハードウェア構成

F2803xには16チャンネルのアナログ入力と独立した45のプログラムマルチプレクサのGPIOポートがあります。GPIOポートの大部分はいくつかの機能の1つに割り当てられます。例えばGPIO1のポートはデジタル入力と出力またはPWM出力として使用できます。16A/Dチャンネルのいくつかも同様にデジタル入力、出力、コンパレータ入力として定義できます。それゆえPSIM回路図の中で正しくGPIOポートの機能を定義しないといけません。

シンボル:

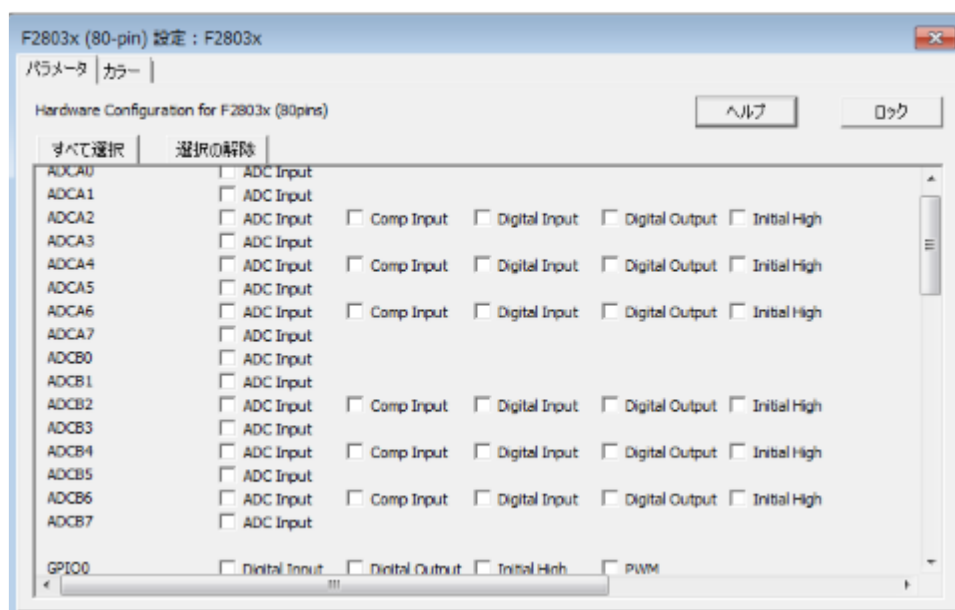
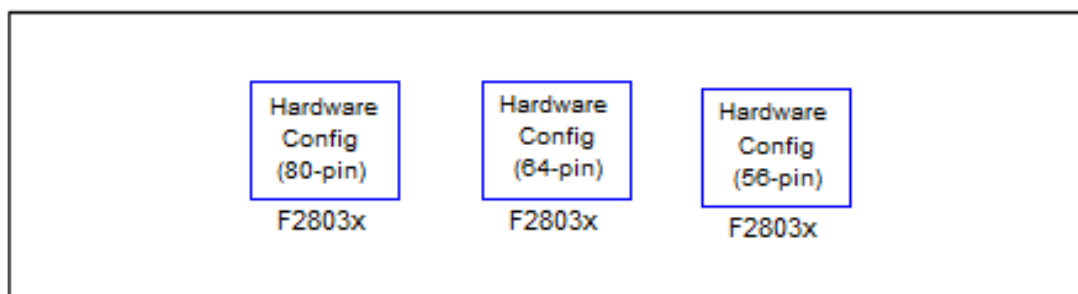


図 8-2 ブロックのダイアログウィンドウ

ハードウェア構成ブロックはF2803xハードウェアのI/Oポートを定義します。ポートは正しくアサインしてください。使用しないポートはチェックを入れないようにしてください。

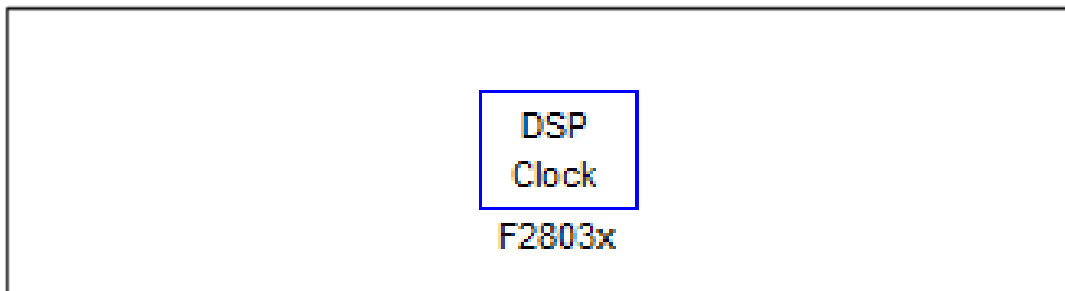
各々のGPIOポートのチェックボックスにある機能が使用可能です。SimCoderではこのボックスがチェックされている機能だけが使用できます。例えばポートGPIO1では“Digital Input”がチェックされるとこのポートは“Digital Input”となります。他の機能としては使用できません

PSIM 回路中で PWM 出力として使用されるとエラーメッセージが出ます。

8.3 DSP クロック

DSP クロックブロックは外部クロック周波数と F2803x DSP の周波数の設定ができます。

シンボル:



仕様:

パラメータ	機能
DSP クロック ソース	F2803xは次の5つの方法でシステムクロックを提供します。 ・ Internal Oscillator1(内部発信器1) ・ Internal Oscillator2(内部発信器2) ・ External Oscillator(外部発信器) ・ External clock(GPIO19) ・ External clock(GPIO38)
外部クロック (MHz)	DSPボード上の外部クロックの周波数(単位はMHz) 周波数は整数値である必要があり最大値は10MHzです。このパラメータはDSPクロックソースが内部オシレータ1か2と選択された場合は無視されます。
DSP速度 (MHz)	DSP速度 (単位はMHz) 。 入力するDSP速度は整数値でなければなりません。外部クロック周波数の整数倍 (1~12倍) でなければなりません。60MHzが最大値です。

DSP クロックブロックが回路中で使用されていない場合は DSP ブロックのデフォルト値が使用されます。

8.4 PWM 生成器

F2803x には、

- PWM1(GPIO0、GPIO1)
- PWM2(GPIO2、GPIO3)
- PWM3(GPIO4、GPIO5)
- PWM4(GPIO6、GPIO7)
- PWM5(GPIO8、GPIO9)
- PWM6(GPIO10、GPIO11)
- PWM7(GPIO40、GPIO41)

のように合計で 7 組の PWM 出力があります。各組は互いに相補的になっており、特別な設定の場合を除いて、PWM1A が正の出力ならば PWM1B は逆の負の出力になります。

SimCoder では 7 つの PWM を下記のように使うことができます。

-3-phase PWM 生成器 : 2 種類あります。PWM1、2、3 と PWM4、5、6 という組み合わせです。

4. -2-phase PWM 生成器 : 7 種類あります。PWM1~7 を相補的な動作ではなく、設定によって幾つかの特別なモードで動作させることができます。

5. -1-phase PWM 生成器 : PWM1~7 互いに相補的な二つの出力をもっています。

-位相シフト付 1-phase PWM 生成器 : PWM2~6 は互いに相補的な 2 つの出力をもっています。

これらの PWM 生成器は A/D 変換のトリガを掛けることや、トリップゾーン信号を用いることもできます。

上記で説明した PWM 生成器の他にもう一点、6 つの APWM 生成器があります。これは機能的には他のブロックに比べて制限されているものとなっており、A/D 変換トリガやトリップゾーンが使えず、更にピンがキャプチャと共用になっているため、キャプチャ機能と同時に使うことができません。

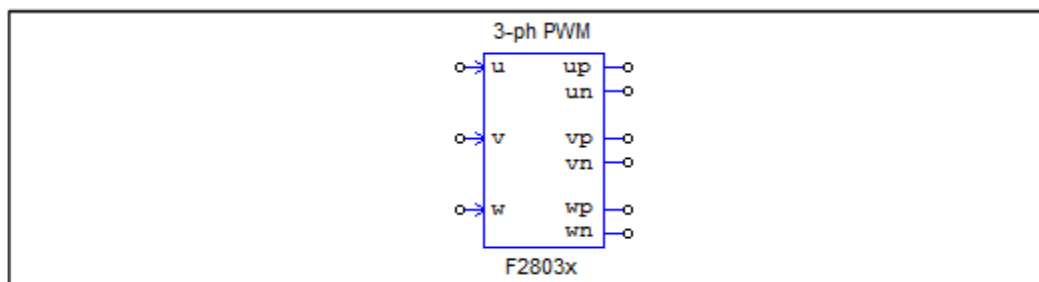
SimCoder の PWM 生成器は内部にスイッチング一回分の遅れを持っており、PWM の出力は入力が入ってから 1 サイクル分遅れて出力されます。これは実際の DSP の持つ遅れをシミュレーションに反映するためです。

PWM 周波数倍率は F2803x の制限では 1~3 です。(設定は 1 から 100 までできます。3 より大きくしたい場合は、PSIM が未使用の PWM を使って、制御周波数で割り込みを引き起こします。この未使用の PWM は周期的な割り込みを生成するためにしか使用されません、PWM の出力端子は、他の機能に設定することができます。もしシステム内に未使用 PWM がない場合、CPU タイマーが使用されます。

8.4.1 3-phase PWM 生成器 :

三相 PWM 生成器の図中では、u、v、w は三相を示しています。(a、b、c 相とも呼ばれます。) p は正の出力を示し、n は負の出力を示しています。例えば、三相 PWM123 では、"up" は PWM1A、"un" は PWM1B となります。

シンボル:



仕様:

パラメータ	機能
PWM ソース	PWM生成器の出力ピン設定。「3-phase PWM 123」(PWM1~3)か「3-phase PWM 456」(PWM4~6)のどちらかを選択します。
デッドタイム	PWM生成器のデッドタイム(Td)の設定。単位は[sec]
サンプリング周波数	PWM生成器のサンプリング周波数の設定。単位は[Hz] PWM信号のデューティサイクルはここで設定した周波数で更新されます。
PWM周波数倍率	PWM周波数とサンプリング周波数の倍率の設定。 1~100の設定ができます。 例えばサンプリング周波数が50kHzでスケーリングファクターが3の場合はPWMスイッチング周波数は150kHzになります。ゲーティング信号は50kHzでもしくは3回につき1回のスイッチングサイクルで更新を行います。
キャリア波形タイプ	キャリア波形タイプの設定。 「Triangular(start low)」: キャリアを三角波、PWM出力の初期値はLow。 「Triangular(start high)」: キャリアを三角波、PWM出力の初期値はHIGH。 「Sawtooth(start low)」: キャリアをのこぎり波、PWM出力の初期値はLow。 「Sawtooth(start high)」: キャリアをのこぎり波、PWM出力の初期値はHIGH。
A/D変換トリガ	A/D変換をトリガするPWMジェネレータの有無を設定します。 以下のいずれかとなります。 「Do not trigger ADC」: PWMはA/D変換へのトリガを無効にします。 「Trigger ADC」: A/D変換器へのトリガを有効にします。 -
A/D変換トリガ位置	A/D変換トリガを出力する位置の設定。値は0~1未満まで設定可能です。 (例) 0の場合、PWM周期の始まりでA/D変換トリガを掛けます。 0.5の場合、PWM周期の180°の位置でA/D変換トリガを掛けます。
トリップゾーンi使用	PWM生成器と繋がる6つの各トリップゾーンの設定。 「Disable Trip-Zone i」: 対応するトリップゾーン信号を無効にします。 「One shot」: One shotモードでトリップゾーン信号を使用します。トリップゾーン信号が検出されると、PWM出力は手動で開始させる必要があります。 「Cycle by cycle」: cycle-by-cycleモードでトリップゾーン信号を使用します。そのときの周期内でトリップゾーン信号が有効になり、次の周期でPWMは自動的に再出力されるようになります。
PWMA DC Trip Src1(DCAH)	PWMA用のDigital compare(DC)トリップソースDCAHです。PWMAは3つのトップスイッチに対して出力"up","vp","wp"を参照します。 PWMチャンネルはDCAHとDCALの2つのDCトリップソースを持っています。 トリップソースは次の中の1つとなります。 - Do not use: PWMはこの信号を使用しません。 - Trip-zone 1: PWMはtrip-zone 1をdigital compare input signalとして使用します。 - Trip-zone 2: PWMはtrip-zone 2をdigital compare input signalとして使用します。 - Trip-zone 3: PWMはtrip-zone3をdigital compare input signalとして使用します。 - Comparator 1: PWM はComparator 1をdigital compare input signalとして使用します。 - Comparator 2: PWM はComparator 2をdigital compare input signalとして使用します。

	- <i>Comparator 3</i>: PWM は <i>Comparator 3</i> を digital compare input signal として使用します。 .
PWMA DC Trip Src1(DCAL)	PWMA用のDigital compare(DC)トリップソースDCALです。トリップソースは次のどれかになります。 - <i>Do not use</i>: PWM はこの信号を使用しません。 - <i>Trip-zone 1</i>: PWMはtrip-zone1をdigital compare入力信号として使用します。 - <i>Trip-zone 2</i>: PWMはtrip-zone2をdigital compare入力信号として使用します。 - <i>Trip-zone 3</i>: PWMはtrip-zone3をdigital compare入力信号として使用します。 - <i>Comparator 1</i>: PWMはComparator1をdigital compare入力信号として使用します。 - <i>Comparator 2</i>: PWMはComparator2をdigital compare入力信号として使用します。 - <i>Comparator 3</i>: PWMはComparator3をdigital compare入力信号として使用します。
PWMA 1-shot Evt(DCAEVT1)	ワンショット信号がPWMAのDCトリップ信号にどのように使われるかを定義します。DCトリップ信号のアクティブレベルは以下の中から選択できます。 - <i>Do not use</i>: PWMはこの信号を使用しません。 - <i>Trip source 1 is low</i>: PWMはソース信号1がlowの場合tripされます。 - <i>Trip source 1 is high</i>: PWMはソース信号1がhighの場合tripされます。 - <i>Trip source 2 is low</i>: PWMはソース信号2がlowの場合tripされます。 - <i>Trip source 2 is high</i>: PWMはソース信号2がhighの場合tripされます。 - <i>Trip source 1 low & source 2 high</i>: PWMはソース1信号がlowで同時にソース2信号がhighの場合tripされます。
PWMA CBC Evt(DCAEVT1)	cycle-by-cycle信号がPWMAのDCトリップ信号にどのように使われるかを定義する。DCトリップ信号のアクティブレベルは以下の中から選択できます。 - <i>Do not use</i>: PWMはこの信号を使用しません。 - <i>Trip source 1 is low</i>: PWMはソース信号1がlowの場合tripされます。 - <i>Trip source 1 is high</i>: PWMはソース信号1がhighの場合tripされます。 - <i>Trip source 2 is low</i>: PWMはソース信号2がlowの場合tripされます。 - <i>Trip source 2 is high</i>: PWMはソース信号2がhighの場合tripされます。 - <i>Trip source 1 low & source 2 high</i>: PWMはソース1信号がlowで同時にソース2信号がhighの場合tripされます。
PWMB DC Trip Src1(DCBH)	PWMBのデジタルコンペア (DC) トリップソースDCBH。3相PWM発電機の場合PWMBは、3つのボトムスイッチの出力 "un"、"vn"、"wn"を指します。PWMチャンネルは、最大で2つのDCトリップソースを持つことができます。DCBHとDCBLです。トリップソースは以下のいずれかになります。 - <i>Do not use</i>: PWMはこの信号を使用しません。 - <i>Trip-zone 1</i>: PWMはtrip-zone1をdigital compare入力信号として使用します。 - <i>Trip-zone 2</i>: PWMはtrip-zone2をdigital compare入力信号として使用します。 - <i>Trip-zone 3</i>: PWMはtrip-zone3をdigital compare入力信号として使用します。 - <i>Comparator 1</i>: PWMはComparator1をdigital compare入力信号として使用します。 - <i>Comparator 2</i>: PWMはComparator2をdigital compare入力信号として使用します。 - <i>Comparator 3</i>: PWMはComparator3をdigital compare入力信号として使用します。
PWMB DC Trip Src1(DCBL)	PWMBのデジタルコンペア (DC) トリップソースDCBL。トリップソースは以下のいずれかです。 - <i>Do not use</i>: PWMはこの信号を使用しません。 - <i>Trip-zone 1</i>: PWMはtrip-zone1をdigital compare入力信号として使用します。 - <i>Trip-zone 2</i>: PWMはtrip-zone2をdigital compare入力信号として使用します。

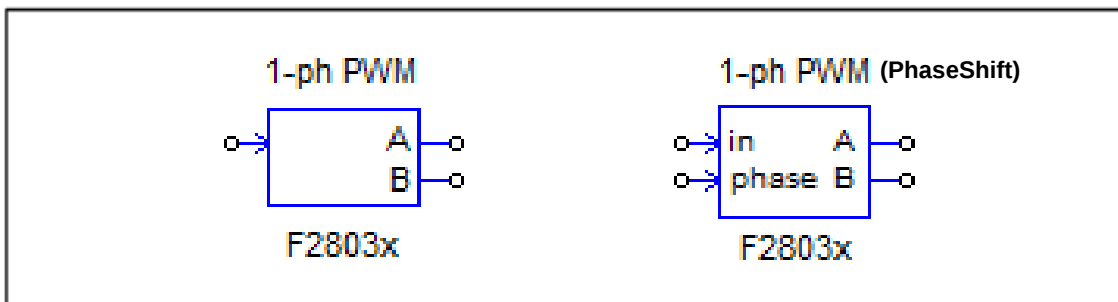
	- <i>Trip-zone 3</i>: PWMはtrip-zone3をdigital compare入力信号として使用します。 - <i>Comparator 1</i>: PWMはComparator1をdigital compare入力信号として使用します。 - <i>Comparator 2</i>: PWMはComparator2をdigital compare入力信号として使用します。 - <i>Comparator 3</i>: PWMはComparator3をdigital compare入力信号として使用します。
PWMB 1-shot Evt (DCBEVT1)	PWMBのDCトリップ信号にワンショット信号を使用する方法を定義します。DCトリップ信号のアクティブレベルを以下の中から選択できます。 - <i>Do not use</i>: PWMはこの信号を使用しません。 - <i>Trip source 1 is low</i>: PWMはソース信号1がlowの場合tripされます。 - <i>Trip source 1 is high</i>: PWMはソース信号1がhighの場合tripされます。 - <i>Trip source 2 is low</i>: PWMはソース信号2がlowの場合tripされます。 - <i>Trip source 2 is high</i>: PWMはソース信号2がhighの場合tripされます。 - <i>Trip source 1 low & source 2 high</i>: PWMはソース1信号がlowで同時にソース2信号がhighの場合tripされます。
PWMB CBC Evt (DCBEVT1)	PWMBのDCトリップ信号に、サイクルバイサイクル信号をどのように使用するかを定義します。DCトリップ信号のアクティブレベルは以下の中から選択できます。 - <i>Do not use</i>: PWMはこの信号を使用しません。 - <i>Trip source 1 is low</i>: PWMはソース信号1がlowの場合tripされます。 - <i>Trip source 1 is high</i>: PWMはソース信号1がhighの場合tripされます。 - <i>Trip source 2 is low</i>: PWMはソース信号2がlowの場合tripされます。 - <i>Trip source 2 is high</i>: PWMはソース信号2がhighの場合tripされます。 - <i>Trip source 1 low & source 2 high</i>: PWMはソース1信号がlowで同時にソース2信号がhighの場合tripされます。
DC Event Filter Source	デジタルコンペアイイベントフィルターのソース。以下のいずれかになります。 - <i>Do not use</i>: DC event filterを使用しません。 - <i>DCAEVT1</i>: DCAEVT1をfilter sourceとして使用します。 - <i>DCAEVT2</i>: DCAEVT2をfilter sourceとして使用します。 - <i>DCBEVT1</i>: DCBEVT1をfilter sourceとして使用します。 - <i>DCBEVT2</i>: DCBEVT2をfilter sourceとして使用します。 - <i>DCAEVT2</i>: ソース信号1がHighの場合、PWMがトリップします。
Blanking Window Pos(us)	PWM周期におけるBlanking windowの開始位置を設定します。(単位 : us)
Blanking Window Width (us)	Blanking windowの幅を指定します。(単位 : us) 幅はハードウェアによって制限され、255/CPU周波数 として計算されます。 例えばCPUのスピードが90MHzの場合幅の範囲は0から2.83usとなります。
Blanking Window Range	Blanking windowどう適用するかを定義します。 - <i>In the window</i>: ブランキング動作の適用範囲はウィンドウ(指定開始位置から指定幅のウィンドウ)内になります - <i>Out of window</i>: ブランキング動作の適用範囲は指定ウィンドウ外となります。
Applying Event Filtering	イベントフィルタリングをデジタル比較イベントにどう適用するかを定義します。イベントフィルタリングは次のデジタル比較イベントのいずれかの組み合わせに対応します。 : DCAEVT1, DCAEVT2, DCBEVT1, and DCBEVT2
トリップ	トリップゾーン信号にトリガされた時のPWMモジュールの出力を設定します。

アクション	- 「High Impedance」：PWMの両出力端子はハイインピーダンスに設定されます。 - 「PWM A high & PWM B low」：PWMのAの出力はハイレベル、Bの出力はローレベルに設定されます。 - 「PWM A low & PWM B high」：PWMのAの出力はローレベル、Bの出力はハイレベルに設定されます。 - 「No action」：何も動作を設定しません。
ピーク間電圧	キャリア波のピーク間電圧Vppの設定。
オフセット電圧	キャリア波のオフセット電圧Voffsetの設定。
初期入力値u, v, w	u、v、w三相入力ノードの初期値の設定。
最初からPWM信号出力	開始からPWM信号出力をするかどうかの設定。 - 「Start」：プログラム開始時にPWM出力を許可します。 - 「Do not start」：「Start PWM」関数を実行するまでPWM出力をしません。
シミュレーション出力モード	シミュレーションの出力モードはSwitching modeかAverage modeに設定されます。“Switching mode”へ設定された時にはPWMブロックの出力はPWM信号となります。“Average mode”へ設定された時にはPWMブロックの出力はAverage mode信号となります。 Average modeでは、搬送波が負から正に向かっており、負のピークと正のピークの絶対値が等しい場合（例えば、搬送波が-1から+1、または-5から+5の場合）、その変調はac信号の変調とみされます。それ以外の場合は、dc信号による変調とみなします。例えば、3相インバータや単相インバータでの変調はac信号変調であり、単相インバータでの変調はdc信号変調となります。 ac信号変調では、PWMブロックの入力uをVuとすると、Average modeの出力upとunは $V_{up} = V_u / (V_{pp} + V_{offset})$ $V_{un} = -V_{up}$ となります。この場合、Vuは-(Vpp+Voffset)からVpp+Voffsetの間、Vupは-1から+1の間となります。 直流信号変調では、アベレージモードでの出力upとunは、次のようになります。 $V_{up} = (V_u - V_{offset}) / V_{pp}$ $V_{un} = 1 - V_{up}$ この場合、VuはVoffsetからVpp+Voffsetの間、Vupは0から+1の間となります。 Average modeに設定すると、PWMブロックの出力をAverage modeモデルのコンバータ/インバータに接続することができます。

8.4.2 1-phase PWM 生成器（外部位相シフト機能付きを含む）：

1-phase PWM 生成器と外部入力位相シフト機能付き 1-phase PWM 生成器はほぼ同じ属性を持っています。違いとしては1相PWMブロックはパラメータを通して位相を定義し、1相PWM(位相シフト)ブロックは外部入力からの位相シフトを読みます(シンボルのラベルPhase部分が該当します)。

シンボル:



仕様:

パラメータ	機能
PWM ソース	PWM生成器の出力ピン設定。位相シフトを使用しない場合は「PWM1」から「PWM6」まで、位相シフトを使用する場合は「PWM2」から「PWM7」を選択します。
出力モード	PWM生成器の出力モードの設定。 - 「Use PWM A&B」 : PWM出力のAとBを両方使います。また互いに相補的になります。 - 「Use PWM A」 : PWM出力Aのみ使用します。 - 「Use PWM B」 : PWM出力Bのみ使用します。
デッドタイム	PWM生成器のデッドタイム (Td) の設定。単位は[sec]
サンプリング周波数	PWM生成器のサンプリング周波数の設定。単位は[Hz] PWM信号のデューティサイクルはここで設定した周波数で更新されます。
PWM 周波数倍率	PWM周波数とサンプリング周波数の倍率の設定。 1~100倍に設定ができます。 例えばサンプリング周波数が50kHzでスケーリングファクターが3の場合はPWMスイッチング周波数は150kHzになります。ゲーティング信号は50kHzで、または3回につき1回のスイッチングサイクルで更新を行います。
キャリア波形タイプ	キャリア波形タイプとPWM出力信号の初期状態の設定。 - 「Triangular(start low)」 : キャリアは三角波で、PWM出力の初期値はLow(0)。 - 「Triangular(start high)」 : キャリアは三角波で、PWM出力の初期値はHIGH(1)。 - 「Sawtooth(start low)」 : キャリアはのこぎり波で、PWM出力の初期値はLow(0)。 - 「Sawtooth(start high)」 : キャリアはのこぎり波で、PWM出力の初期値はHIGH(1)。
A/D変換トリガ	A/D変換へのトリガの設定。 - 「Do not trigger ADC」 : A/D変換トリガ機能を無効にします。 - 「Trigger ADC」 : PWMは複数のA/Dチャンネルをトリガします。
A/D変換トリガ位置	A/D変換トリガを出力する位置の設定。値は0~1まで設定可能です。 (例) 0の場合、PWM周期の始まりでA/D変換トリガを掛けます。 0.5の場合、PWM周期の180° の位置でA/D変換トリガを掛けます。
トリップゾーン i 使用	PWM生成器と繋がる6つの各トリップゾーンの設定。 - 「Disable Trip-Zone i」 : 対応するトリップゾーン信号を無効にします。

	- 「One shot」 : One shotモードでトリップゾーン信号を使用します。トリガがかかるとPWM出力は手動で開始させなければなりません。 - 「Cycle by cycle」 : cycle-by-cycleモードでトリップゾーン信号を使用します。そのときの周期内でトリップゾーン信号が有効になり、次の周期でPWMは自動的に再出力されるようになります。
PWMA DC Trip Src1(DCAH)	PWMA用のDigital compare(DC)トリップソースDCAHです。PWMAは3つのトリップスイッチに対して出力"up","vp","wp"を参照します。 PWMチャネルはDCAHとDCALの2つのDCトリップソースを持っています。トリップソースは次の中の1つとなります。 - Do not use: PWMはこの信号を使用しません。 - Trip-zone 1: PWMはtrip-zone 1をdigital compare input signalとして使用します。 - Trip-zone 2: PWMはtrip-zone 2をdigital compare input signalとして使用します。 - Trip-zone 3: PWMはtrip-zone3をdigital compare input signalとして使用します。 - Comparator 1: PWM はComparator 1をdigital compare input signalとして使用します。 - Comparator 2: PWM はComparator 2 をdigital compare input signalとして使用します。 - Comparator 3: PWM はComparator 3 をdigital compare input signalとして使用します。
PWMA DC Trip Src2(DCAL)	PWMA用のDigital compare(DC)トリップソースDCALです。トリップソースは次のどれかになります。 - Do not use: PWM はこの信号を使用しません。 - Trip-zone 1: PWMはtrip-zone1をdigital compare入力信号として使用します。 - Trip-zone 2: PWMはtrip-zone2をdigital compare入力信号として使用します。 - Trip-zone 3: PWMはtrip-zone3をdigital compare入力信号として使用します。 - Comparator 1: PWMはComparator1をdigital compare入力信号として使用します。 - Comparator 2: PWMはComparator2をdigital compare入力信号として使用します。 - Comparator 3: PWMはComparator3をdigital compare入力信号として使用します。
PWMA 1-shotEvt(DCAEVT1)	- Do not use: PWMはこの信号を使用しません。 - Trip source 1 is low: PWMはソース信号1がlowの場合tripされます。 - Trip source 1 is high: PWMはソース信号1がhighの場合tripされます。 - Trip source 2 is low: PWMはソース信号2がlowの場合tripされます。 - Trip source 2 is high: PWMはソース信号2がhighの場合tripされます。 - Trip source 1 low & source 2 high: PWMはソース1信号がlowで同時にソース2信号がhighの場合tripされます。
PWMA CBC Evt(DCAEVT2)	- Do not use: PWMはこの信号を使用しません。 - Trip source 1 is low: PWMはソース信号1がlowの場合tripされます。 - Trip source 1 is high: PWMはソース信号1がhighの場合tripされます。 - Trip source 2 is low: PWMはソース信号2がlowの場合tripされます。 - Trip source 2 is high: PWMはソース信号2がhighの場合tripされます。 - Trip source 1 low & source 2 high: PWMはソース1信号がlowで同時にソース2信号がhighの場合tripされます。
PWMB DC Trip Src1(DCBH)	- Do not use: PWMはこの信号を使用しません。 - Trip-zone 1: PWMはtrip-zone1をdigital compare入力信号として使用します。

	- <i>Trip-zone 2</i>: PWMはtrip-zone2をdigital compare入力信号として使用します。 - <i>Trip-zone 3</i>: PWMはtrip-zone3をdigital compare入力信号として使用します。 - <i>Comparator 1</i>: PWMはComparator1をdigital compare入力信号として使用します。 - <i>Comparator 2</i>: PWMはComparator2をdigital compare入力信号として使用します。 - <i>Comparator 3</i>: PWMはComparator3をdigital compare入力信号として使用します。
PWMB DC Trip Src1(DCBL)	- <i>Do not use</i>: PWMはこの信号を使用しません。 - <i>Trip-zone 1</i>: PWMはtrip-zone1をdigital compare入力信号として使用します。 - <i>Trip-zone 2</i>: PWMはtrip-zone2をdigital compare入力信号として使用します。 - <i>Trip-zone 3</i>: PWMはtrip-zone3をdigital compare入力信号として使用します。 - <i>Comparator 1</i>: PWMはComparator1をdigital compare入力信号として使用します。 - <i>Comparator 2</i>: PWMはComparator2をdigital compare入力信号として使用します。 - <i>Comparator 3</i>: PWMはComparator3をdigital compare入力信号として使用します。
PWMB 1-shot Evt (DCBEVT1)	- <i>Do not use</i>: PWMはこの信号を使用しません。 - <i>Trip source 1 is low</i>: PWMはソース信号1がlowの場合tripされます。 - <i>Trip source 1 is high</i>: PWMはソース信号1がhighの場合tripされます。 - <i>Trip source 2 is low</i>: PWMはソース信号2がlowの場合tripされます。 - <i>Trip source 2 is high</i>: PWMはソース信号2がhighの場合tripされます。 - <i>Trip source 1 low & source 2 high</i>: PWMはソース1信号がlowで同時にソース2信号がhighの場合tripされます。
PWMB CBC Evt (DCBEVT2)	- <i>Do not use</i>: PWMはこの信号を使用しません。 - <i>Trip source 1 is low</i>: PWMはソース信号1がlowの場合tripされます。 - <i>Trip source 1 is high</i>: PWMはソース信号1がhighの場合tripされます。 - <i>Trip source 2 is low</i>: PWMはソース信号2がlowの場合tripされます。 - <i>Trip source 2 is high</i>: PWMはソース信号2がhighの場合tripされます。 - <i>Trip source 1 low & source 2 high</i>: PWMはソース1信号がlowで同時にソース2信号がhighの場合tripされます。
DC Event Filter Source	- <i>Do not use</i>: DC event filterを使用しません。 - <i>DCAEVT1</i>: DCAEVT1をfilter sourceとして使用します。 - <i>DCAEVT2</i>: DCAEVT2をfilter sourceとして使用します。 - <i>DCBEVT1</i>: DCBEVT1をfilter sourceとして使用します。 - <i>DCBEVT2</i>: DCBEVT2をfilter sourceとして使用します。
Blanking Window Pos(us)	PWM周期におけるBlanking windowの開始位置を指定します。(単位: us)
Blanking Window Width (us)	ブランキングウィンドウの幅(単位: us) 幅はハードウェアによって制限され、255/CPUとして計算されます。例えばCPUのスピードが90MHzの場合幅の範囲は0から2.83usとなります。
Blanking Window Range	ブランキング動作の適用範囲を定義します。 - <i>In the window</i>: ブランキング動作の適用範囲はウィンドウ(指定開始位置から指定幅のウィンドウ)内になります。 - <i>Out of window</i>: ブランキング動作の適用範囲は指定ウィンドウ外となります。
Applying Event Filtering	イベントフィルタリングをデジタル比較イベントに適用される方法を指定します。イベントフィルタリングは次のデジタル比較イベントの組み合わせに適用できます。 : DCAEVT1, DCAEVT2, DCBEVT1, and DCBEVT2

トリップ アクション	トリップゾーン信号に対するPWM生成器の動作の設定。 - 「High Impedance」：PWM出力部はハイインピーダンスになります。 - 「PWM A high & PWM B low」：PWMのA端子がハイレベル、B端子がローレベルになります。 - 「PWM A low & PWM B high」：PWMのA端子はローレベル、B端子がハイレベルになります。 - 「No action」：何も動作を設定しません。
ピーク間電圧	キャリア波のピーク間電圧Vppの設定。
オフセット電圧	キャリア波のオフセット電圧Voffsetの設定。
位相シフト	PWM生成器の出力に対してシフトする位相の値[deg] (外部位相シフト入力なしの1-phase PWM生成器)
初期入力値	入力ノードの初期値の設定。
Use HRPWM	高分解能PWMの定義を行います。 <i>Do not use HRPWM</i> : 高分解能PWMを使用しません。 <i>Use HRPWM</i> (校正なし)：校正なしで高分解能PWMを使用します。 <i>Use HRPWM</i> (校正あり)：校正ありで高分解能PWMを使用します。
最初からPWM信号	開始からPWM信号出力をするかどうかの設定。 - 「Start」：プログラム開始時にPWM出力を許可します。 - 「Do not start」：「Start PWM」関数を実行するまでPWM出力をしません。
シミュレーション 出力モード	シミュレーションの出力モードはSwitchingモードかAverageモードに設定されます。 “Switchingモード”へ設定された時にはPWMブロックの出力はPWM信号となります。 “Averageモード”へ設定された時にはPWMブロックの出力は平均モード信号となります。 例えばPWMブロックの入力をVuとすると出力のupとunは $V_{up} = V_u / (V_{pk_pk} + V_{offset})$ $V_{un} = -V_u / (V_{pk_pk} + V_{offset})$ となります。Average mode に設定されるとPWMブロックの出力はaverage mode modelのインバータへ接続されます。

Phase Shift

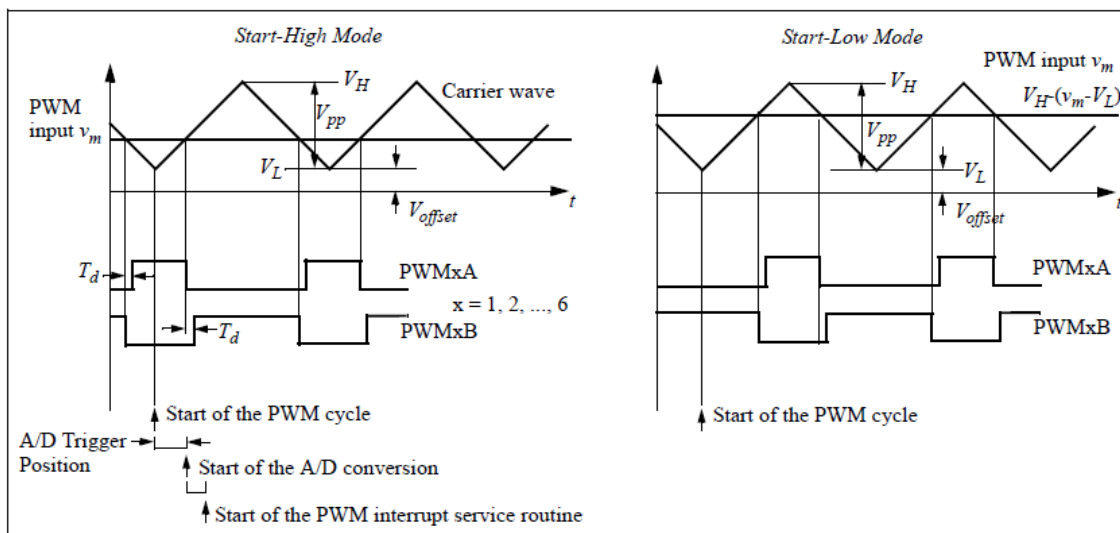
1-phase PWM 生成器は、別の PWM 信号に対して位相をシフトさせた PWM 信号を生成することができます。例えば F2803xDSP では PWM 位相シフトは次のようになります。

- どのような PWM でもその前の PWM も PSIM 回路に位相シフト PWM として使用されることが必要になります。例えば、PWM1、2、3 は正しい位相シフト系列ですが、PWM1、3、4 は正しい位相シフト系列ではありません。
- 基準となる PWM とシフトされる PWM は、組み合わせ内で連続している必要があります。つまり、PWM1 を基準として PWM3、または PWM5 または PWM6 を位相シフトすることはできません。

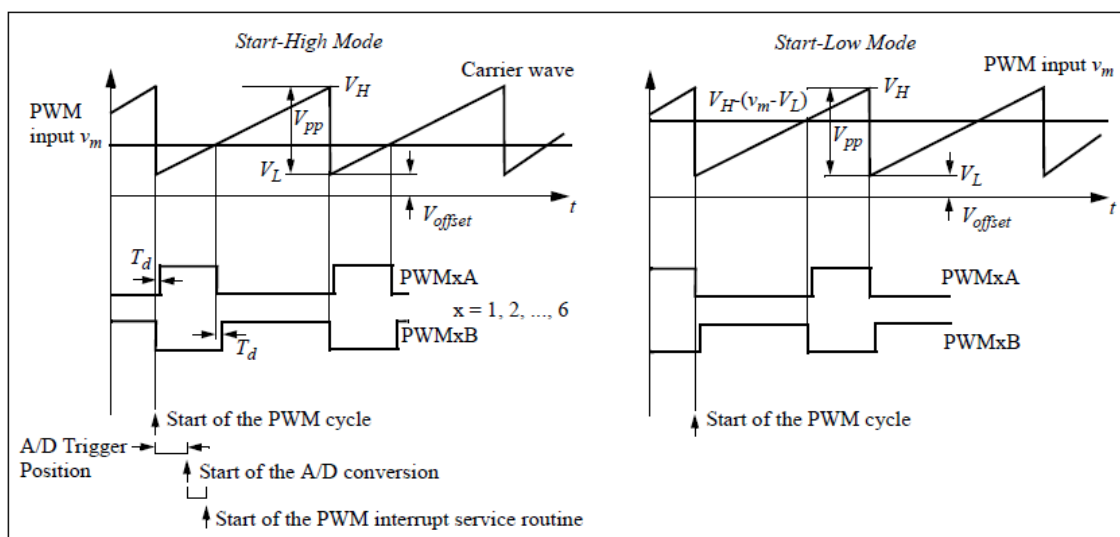
位相シフトの値は、度[deg]になります。値が-30° の場合、出力は基準の PWM 生成器の出力に対してスイッチングサイクルの 30° 右に（遅れ）にシフトされます。これは、30° 右に PWM の搬送波をシフトすることと同じです。位相の値が 30° の場合、出力は 30° 左（進み）にシフトされます。

搬送波

キャリア波形には、三角波（立ち上がりと立ち下りの間隔が等しい）とノコギリ波の 2 種類があります。また、動作モードには、次のようにスタート・ロー・モードとスタート・ハイ・モードの 2 種類があります。三角波搬送波を用いた PWM 生成器の入力波形と出力波形を次に示します。



ノコギリ搬送波の PWM 生成器の入力波形と出力波形は次のとおりです。



前図は、デッドタイムの定義と、PWM 発生器が A/D コンバータをトリガするときのタイムシーケンスを示しています。

A/D コンバータのトリガが選択された場合、PWM サイクルの開始から、A/D トリガの位置で定義される一定の遅延の後、A/D 変換が開始されます。

A/D 変換が完了した後、PWM 割り込みサービスルーチンが開始されます。

PWM 生成器が A/D コンバータを動作させない場合は、PWM サイクルの開始時に PWM 割り込みルーチンが開始されます。

前図は、“start-High”および“start-low”モードの動作方法を示しています。PWM 入力を v_m 、搬送波の最小値を V_L 、最大値を V_H とします。“start-high”モードでは、PWM 正出力 PWMA は、スイッチングサイクルの開始時に high レベルとなり、入力 v_m が搬送波より大きい間、high レベルを維持します。例えば v_m が搬送波よりも大きい間、ハイ状態を維持します。搬送波が $0 \sim 1$ 、つまり $V_L=0$ 、 $V_H=1$ の時、 V_m が 0.2 の場合、PWM 出力 PWMA は搬送波が 0.2 より小さい間は high を持続します。

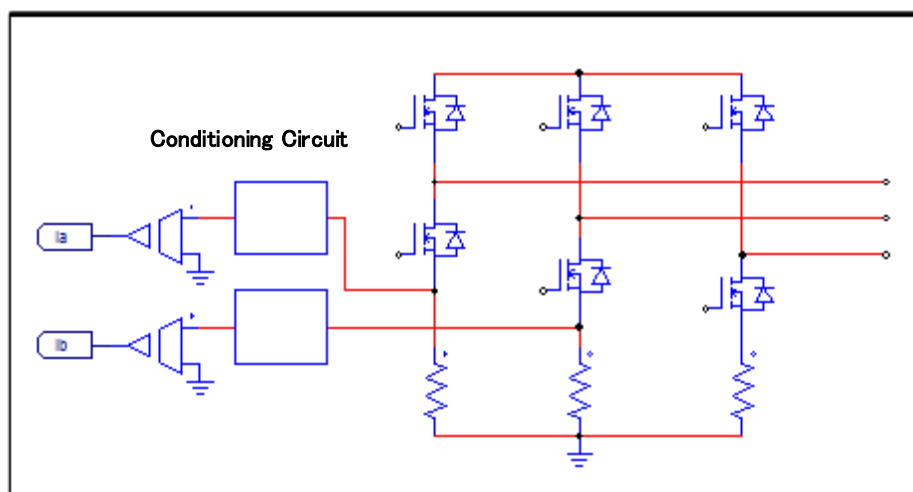
一方、“start-low”モードでは、PWM 正出力 PWMA はスイッチングサイクルの開始時に low となり、搬送波が「 $V_H - (V_m - V_L)$ 」の値より大きい際に high となります。例えば、搬送波が $0 \sim 1$ 、つまり $V_L=0$ 、 $V_H=1$ の時、 V_m が 0.2 の場合、PWM 出力 PWMA は搬送波が 0.8 より大きい間は high となります。

キャリア開始モードはスイッチ電流の測定方法によりかわります。3 相インバータの場合はスイッチ電流が測定され、スタートハイモードが選択されます。これは上のスイッチが周期の最初にトップスイッチゲーティング信号が高く、電流が流れていることを示します。

一方低いスイッチ電流が測定されているとスタートローモードが選択されます。これは周期の最初に

トップスイッチゲーティング信号が低く、下のスイッチゲーティング信号が高く、電流が流れていることとなります。

次のような回路の場合、位相 A と B の下のスイッチ電流が測定されます。この場合はキャリアスタートローモードを選択すべきです。

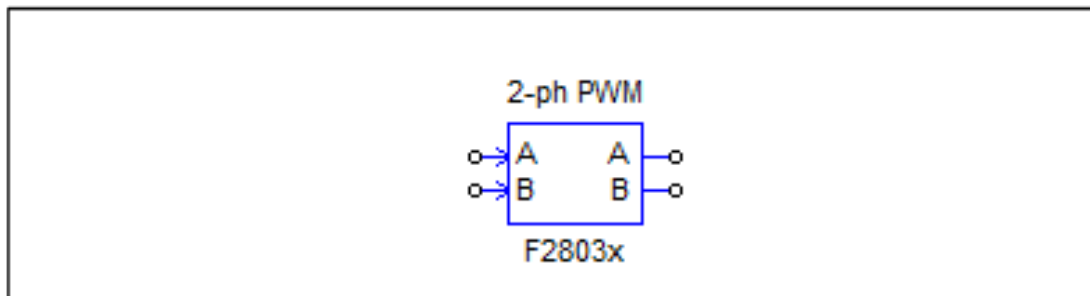


注：“Start-Low”モードでは、PWM 信号を生成するために搬送波と比較する前に PWM 入力“ V_m ”は“ $V_H - (V_m - V_L)$ ”に変換されます。変換によって、“Start-Low”と“Start-High”の両方のモードで同じデューティ比の表現となります。例えば、 $V_L=0$ 、 $V_H=1$ のノコギリ波、または $V_L=-V_H$ の三角波の場合、PWMA 出力のデューティ比 D は“Start-Low”と“Start-High”の両方で「 $D=V_m/V_H$ 」となります。

8.4.3 2-phase PWM 生成器：

2 相 PWM ブロックは 6 つの動作モードがあります。

シンボル:



仕様:

パラメータ	機能
PWM ソース	PWM生成器の出力ピン設定。 PWM1~6-*+までを選択します。
モード・タイプ	PWM生成器の動作モードの設定。 6つのモードから一つを選択します。各モードの動作の詳細は次の図を参照してください。
サンプリング 周波数	PWM生成器のサンプリング周波数の設定。単位は[Hz] PWM信号のデューティサイクルはここで設定した周波数でアップデートされます。
PWM 周波数倍率	PWM周波数とサンプリング周波数の倍率の設定。 1~100倍に設定ができます。例えばサンプリング周波数が50kHzでスケール倍率が3の場合、PWMスイッチング周波数は150kHzとなります。つまりスイッチは150kHzで動作します。がゲーティング信号は50kHzで、または3スイッチングサイクル毎に1回更新されます。
A/D変換トリガ	A/D変換へのトリガの設定。 - 「Do not trigger ADC」 : A/D変換トリガ機能を無効にします。 - 「Trigger ADC」 : A/D変換のトリガ機能を有効にします。 -
A/D変換トリガ 位置	A/D変換チャネルのPWMモジュールによるトリガされた変換位置。 トリガされた変換位置は搬送波の開始時にまたは搬送波の途中のいずれかに設定できます。途中トリガ位置は動作モード4,5、及び6の場合に有効です。もしPWMがADCをトリガしない場合はこの位置はPWMの割り込みの位置になります。
トリップゾーン <i>i</i> 使用	トリップゾーン <i>i</i> 信号の使用方法の設定。「Do not use」 : PWMはこの信号を使用しません。「One shot」 : PWMはトリップゾーン <i>i</i> をOne shot信号として使用します。 - 「Cycle by cycle」 : PWMはトリップゾーン<i>i</i>をcycle-by-cycle信号として使用します。モードでトリップゾーン信号を使用します。そのときの周期内でトリップゾーン信号が有効になり、次の周期でPWMは自動的に再出力されるようになります。
PWMA DC Trip Src1(DCAH)	PWMA用のDigital compare(DC)トリップソースDCAHです。PWMAは3つのトックスイッチに対して出力"up","vp","wp"を参照します。 PWMチャネルはDCAHとDCALの2つのDCトリップソースを持っています。 トリップソースは次の中の1つとなります。 - Do not use: PWMはこの信号を使用しません。 - Trip-zone 1: PWMはtrip-zone 1をdigital compare input signalとして使用します。 - Trip-zone 2: PWMはtrip-zone 2をdigital compare input signalとして使用します。

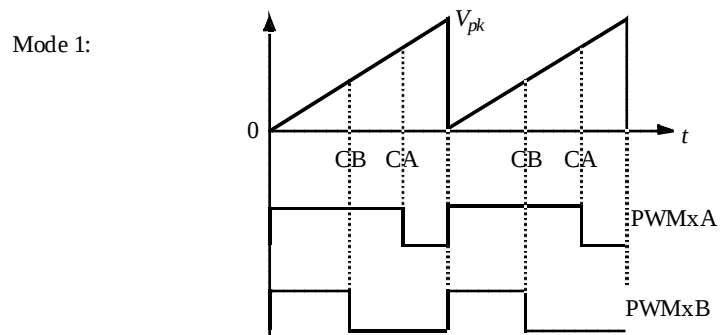
	- Trip-zone 3: PWMはtrip-zone3をdigital compare input signalとして使用します。 - Comparator 1: PWM はComparator 1をdigital compare input signalとして使用します。 - Comparator 2: PWM はComparator 2をdigital compare input signalとして使用します。 - Comparator 3: PWM はComparator 3をdigital compare input signalとして使用します。
PWMA DC Trip Src2(DCAL)	PWMA用のDigital compare(DC)トリップソースDCALです。トリップソースは次のどれかになります。 - Do not use: PWMはこの信号を使用しません。 - Trip-zone 1: PWMはtrip-zone1をdigital compare入力信号として使用します。 - Trip-zone 2: PWMはtrip-zone2をdigital compare入力信号として使用します。 - Trip-zone 3: PWMはtrip-zone3をdigital compare入力信号として使用します。 - Comparator 1: PWMはComparator1をdigital compare入力信号として使用します。 - Comparator 2: PWMはComparator2をdigital compare入力信号として使用します。 - Comparator 3: PWMはComparator3をdigital compare入力信号として使用します。
PWMA 1-shotEvt(DCAEVT1)	- Do not use: PWMはこの信号を使用しません。 - Trip source 1 is low: PWMはソース信号1がlowの場合tripされます。 - Trip source 1 is high: PWMはソース信号1がhighの場合tripされます。 - Trip source 2 is low: PWMはソース信号2がlowの場合tripされます。 - Trip source 2 is high: PWMはソース信号2がhighの場合tripされます。 - Trip source 1 low & source 2 high: PWMはソース1信号がlowで同時にソース2信号がhighの場合tripされます。
PWMA CBC Evt(DCAEVT2)	- Do not use: PWMはこの信号を使用しません。 - Trip source 1 is low: PWMはソース信号1がlowの場合tripされます。 - Trip source 1 is high: PWMはソース信号1がhighの場合tripされます。 - Trip source 2 is low: PWMはソース信号2がlowの場合tripされます。 - Trip source 2 is high: PWMはソース信号2がhighの場合tripされます。 - Trip source 1 low & source 2 high: PWMはソース1信号がlowで同時にソース2信号がhighの場合tripされます。
PWMB DC Trip Src1(DCBH)	- Do not use: PWMはこの信号を使用しません。 - Trip-zone 1: PWMはtrip-zone1をdigital compare入力信号として使用します。 - Trip-zone 2: PWMはtrip-zone2をdigital compare入力信号として使用します。 - Trip-zone 3: PWMはtrip-zone3をdigital compare入力信号として使用します。 - Comparator 1: PWMはComparator1をdigital compare入力信号として使用します。 - Comparator 2: PWMはComparator2をdigital compare入力信号として使用します。 - Comparator 3: PWMはComparator3をdigital compare入力信号として使用します。
PWMB DC Trip Src1(DCBL)	- Do not use: PWMはこの信号を使用しません。 - Trip-zone 1: PWMはtrip-zone1をdigital compare入力信号として使用します。 - Trip-zone 2: PWMはtrip-zone2をdigital compare入力信号として使用します。 - Trip-zone 3: PWMはtrip-zone3をdigital compare入力信号として使用します。 - Comparator 1: PWMはComparator1をdigital compare入力信号として使用します。 - Comparator 2: PWMはComparator2をdigital compare入力信号として使用します。

	- <i>Comparator 3</i> : PWMはComparator3をdigital compare入力信号として使用します。
PWMB 1-shot Evt (DCBEVT1)	- <i>Do not use</i> : PWMはこの信号を使用しません。 - <i>Trip source 1 is low</i> : PWMはソース信号1がlowの場合tripされます。 - <i>Trip source 1 is high</i> : PWMはソース信号1がhighの場合tripされます。 - <i>Trip source 2 is low</i> : PWMはソース信号2がlowの場合tripされます。 - <i>Trip source 2 is high</i> : PWMはソース信号2がhighの場合tripされます。 - <i>Trip source 1 low & source 2 high</i> : PWMはソース1信号がlowで同時にソース2信号がhighの場合tripされます。
PWMB CBC Evt (DCBEVT2)	- <i>Do not use</i> : PWMはこの信号を使用しません。 - <i>Trip source 1 is low</i> : PWMはソース信号1がlowの場合tripされます。 - <i>Trip source 1 is high</i> : PWMはソース信号1がhighの場合tripされます。 - <i>Trip source 2 is low</i> : PWMはソース信号2がlowの場合tripされます。 - <i>Trip source 2 is high</i> : PWMはソース信号2がhighの場合tripされます。 - <i>Trip source 1 low & source 2 high</i> : PWMはソース1信号がlowで同時にソース2信号がhighの場合tripされます。
DC Event Filter Source	- <i>Do not use</i> : DC event filterを使用しません。 - <i>DCAEVT1</i> : DCAEVT1をfilter sourceとして使用します。 - <i>DCAEVT2</i> : DCAEVT2をfilter sourceとして使用します。 - <i>DCBEVT1</i> : DCBEVT1をfilter sourceとして使用します。 - <i>DCBEVT2</i> : DCBEVT2をfilter sourceとして使用します。
Blanking Window Pos(us)	PWM周期におけるBlanking windowの開始位置を指定します。(単位: us)
Blanking Window Width (us)	ブランキングウィンドウの幅(単位: us) 幅はハードウェアによって制限され、255/CPUとして計算されます。例えばCPUのスピードが90MHzの場合幅の範囲は0から2.83usとなります。
Blanking Window Range	ブランキング動作の適用範囲を定義します。 - <i>In the window</i> : ブランキング動作の適用範囲はウィンドウ(指定開始位置から指定幅のウィンドウ)内になります。 - <i>Out of window</i> : ブランキング動作の範囲範囲は指定ウィンドウ外となります。
Applying Event Filtering	イベントフィルタリングをデジタル比較イベントに適用される方法を指定します。イベントフィルタリングは次のデジタル比較イベントの組み合わせに適用できます。 : DCAEVT1, DCAEVT2, DCBEVT1, and DCBEVT2
トリップアクション	トリップゾーン信号に対するPWM生成器の動作の設定。 - 「High Impedance」 : PWM出力部はハイインピーダンスになります。 - 「PWM A high & PWM B low」 : PWMの正の出力はハイレベル、負の出力はローレベルになります。 - 「PWM A low & PWM B high」 : PWMの正の出力はローレベル、負の出力はハイレベルになります。 - 「No action」 : 何も動作を設定しません。
ピーク間電圧	キャリア波のピーク電圧Vpkの設定。
初期入力値A, B	A、B入力ノードの初期値の設定。
最初からPWM信号	開始からPWM信号出力をするかどうかの設定。 - 「Start」 : 開始からPWM出力を許可します。 - 「Do not start」 : 「Start PWM」関数を実行するまでPWM出力をしません。

2-phase PWM 生成器の出力形式は動作モード設定によって変わります。(以降の図参照) モードによってキャリアは三角波とのこぎり波のどちらかになります。また、キャリア値は0からVpkまでの間の値を取り、DC オフセットの設定はできません。

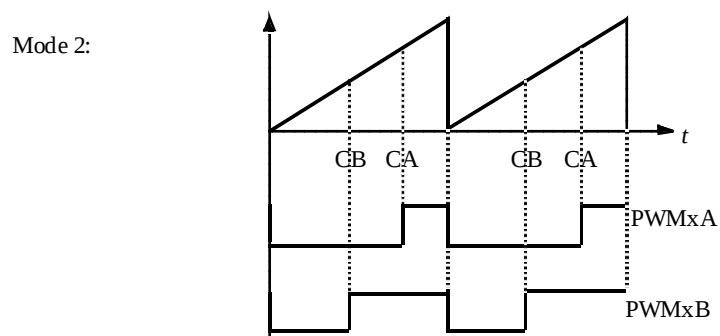
動作モード 1:

図の CA と CB は 2-phase PWM 生成器の二つの入力 A、B になり、各出力のターンオフタイミングを決めます。



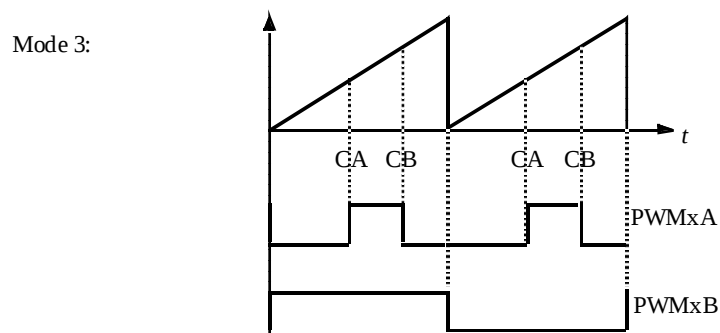
動作モード 2:

動作モード 1 と違い、各出力のターンオンタイミングを決めるモードです。



動作モード 3:

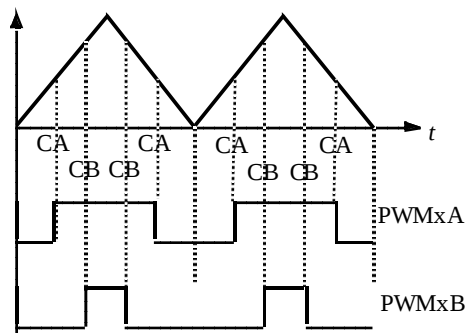
入力 A は PWM A 出力のターンオンのタイミングを決め、入力 B は PWM B 出力のターンオフのタイミングを決めます。PWM B 出力は PWM 周期の間オンになり、次の周期の間はオフになる、を繰り返します。



動作モード 4 :

キャリアは三角波になります。各入力は対応する出力のターンオンとターンオフのタイミングを決めます。

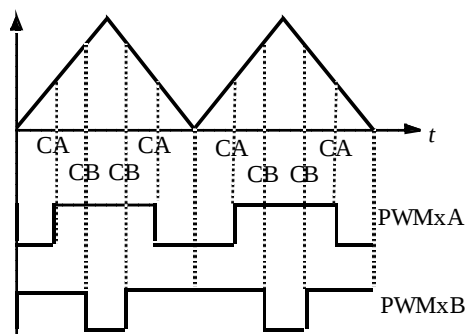
Mode 4:



動作モード 5 :

動作モード 4 と違い、PWM A 出力と PWM B 出力のターンオフとターンオンのタイミングが反対に動作します。

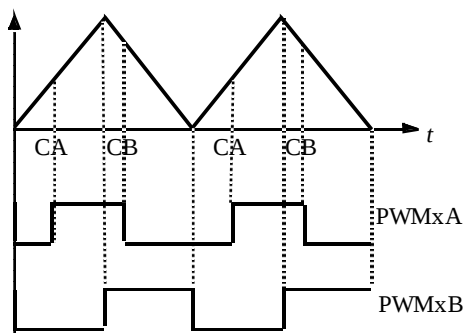
Mode 5:



動作モード 6 :

入力 A は PWM A のターンオンのタイミングを決め、入力 B は PWM A のターンオフのタイミングを決めます。PWM B は PWM 周期の最初の半周期でオンになり、次の半周期でオフになる動作をします。

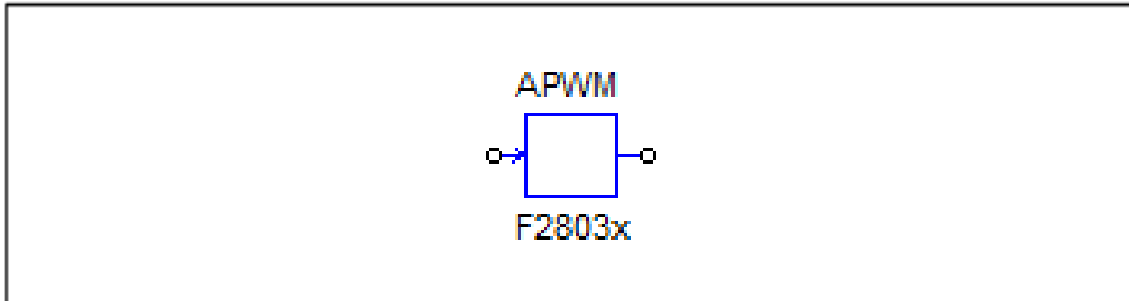
Mode 6:



8.4.4 Single PWM 生成器(APWM) :

SinglePWM 生成器は APWM とも言われキャプチャと同じリソースを共有しています。

シンボル:



仕様:

パラメータ	機能
PWM ソース	Single PWM生成器(APWM生成器)はキャプチャと共用で3つの指定されたGPIOポートを使用できます。 - APWM1(GPIO5, GPIO19, GPIO24) -
PWM 周波数	PWM生成器の周波数の設定。単位は[Hz]
キャリア波形タイプ	キャリア波形タイプの設定。 次のいずれかになります。: - のこぎり波 (低でスタート): キャリアをのこぎり波、PWM出力の初期値はLOW。 - のこぎり波 (高でスタート): キャリアをのこぎり波、PWM出力の初期値はHIGH。
スツプ出力	PWMジェネレータが停止している時の出力設定。 - 出力 低: PWM出力は低に設定されます。 - 出力 高: PWM出力は高に設定されます。
ピーク間電圧	キャリア波のピーク間電圧の設定。
オフセット値	キャリア信号のオフセット電圧の設定。
位相シフト	PWM生成器の出力に対しての位相シフト値 (単位deg)
初期入力値	入力の初期値の設定。
最初からPWM信号	開始からPWM信号出力をするかどうかの設定。 - 「Start」 : 開始からPWM出力を許可します。 - 「Do not start」 : 「Start PWM」関数を実行するまでPWM出力をしません。

1 相 PWM 生成器と同様に APWM 生成器は他の PWM 生成器に関して位相シフトをもつ PWM 信号を生成します。位相シフトのルールは 1 相 PWM 生成器と同じです。

Single PWM 生成器は機能的には他のブロックに比べて制限されており、A/D 変換トリガやトリップゾーン信号が利用できません。

8.4.5 PWM ブロック間の同期

PWM の 3 つのタイプで同期が可能です。位相シフトはお互いの間で定義できます。単相 PWM、単相 PWM(位相シフト付)と APWM(もしくは単層 PWM(キャプチャと共用)です。

単相 PWM ブロックは別の PWM 信号で位相シフトした PWM 信号を生成できます。通常の PWM ブロックには PWM1、2、3 と PWM1、4、5、6 の 2 つのシリーズがあります。

同様に APWM ブロックには位相シフトのための 2 つのシリーズがあります。それは PWM1、APWM1、2、3 と PWM1、APWM4、5、6 です。

位相シフトのための PWM ブロックの定義は次のようになります。

ー基準 PWM と位相シフトされる PWM は同じシリーズでなければなりません。すなわち PWM1 が基準、PWM2、3 か PWM4、5、6 が PWM1 に関して位相シフトできます。または、PWM2 が基準となる場合は PWM3 が PWM2 について位相シフトできます。

同様に PWM4 (または 5) は基準となり、PWM5 (または 6) は PWM4 (または 5) に関して位相シフトできます。しかし、PWM4、5、6 に対して PWM2 または 3 を基準として使うことはできません。

これはまた APWM についても同様です。例えば PWM1 が基準となる場合、APWM1、2、3 は位相シフトされます。同様に PWM1 が基準の場合、APWM4、5、6 が位相シフトできます。または、APWM4 が基準の場合、APWM5 と 6 は位相シフトできます。

ー基準 PWM と位相シフトされる PWM は省略された PWM が使用されないようにシリーズの中で連続でなければなりません。例えばもし PWM1、2、3 すべてが使用されているが PWM2 が PWM1 と 3 と同期していない場合は使用できません。しかしもし PWM2 が回路中で使用されていない場合は PWM1 を基準、位相シフト PWM3 として使用することができます。

例えば正しい設定は次のようになります。最初が基準でその後のブロックが位相シフトとなります。

PWM 1 (reference), PWM 2, PWM 3, PWM 4, PWM 5, PWM 6, PWM 7

PWM 2 (reference), PWM 3

PWM 4 (reference), PWM 5

PWM 5 (reference), PWM 6

PWM 6 (reference), PWM 7

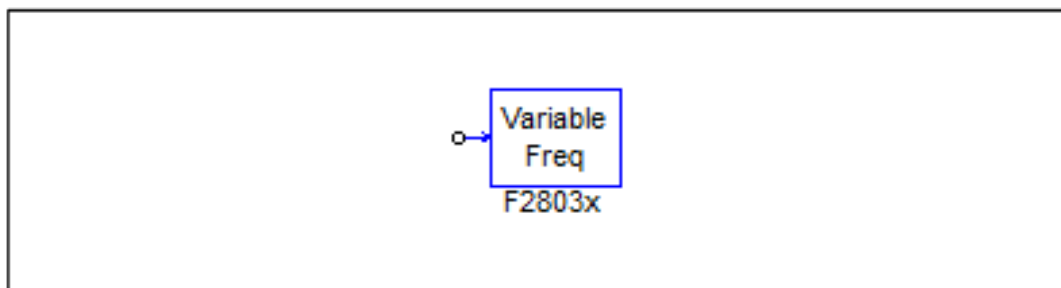
PWM 1 (reference), PWM 2, PWM 3, PWM 4, PWM 5, PWM 6, PWM 7, APWM 1

PWM 1 (reference), APWM 1

8.5 可変周波数 PWM

可変周波数 PWM ブロックでは PWM 生成器のサンプリング周波数の変更ができます。

シンボル：



仕様

パラメータ	機能
PWM ソース	PWM 生成器のソース PWW1～7、3 相 PWM123 と PWM456 を選択できます。
割り込み位置調整	PWM 周波数が調整する時、割り込み位置を調整するかどうかを指定します。 - Do not adjust(調整しない): 割り込み位置はベース周波数で計算された初期位置を維持します。 - Adjust(調整する): 割り込み位置は、新しい周波数に基づき再計算されます、次の PWM 周期で適用されます。
ランプ補正調整	PWM 周波数を調整する時、コンパレータ DAC ブロックのランプ補償を調整するかどうかを指定します。 - Do not adjust(調整しない): ランプ補正はベース周波数で計算された初期補正値を維持します。 - Adjust(調整する): ランプ補正は新しい周波数に基づき再計算されます。次の PWM 周期で適用されます。

PWM ブロックと一致するサンプリング周波数は次のような PWM 周期の最初に変更されます。

$$PWM_Frequency = PWM_Base_Frequency / Input_Value$$

ここで $PWM_Base_Frequency$ は PWM_block に一致したサンプリング周波数です。

$Input_Value$ はこのブロックの入力値です。

割り込み位置を調整する場合、各サイクルで割り込み位置が再計算されます。割り込み位置調整は時間がかかるので周波数変化が小さい場合はこの割り込み位置の調整は行わないでください。

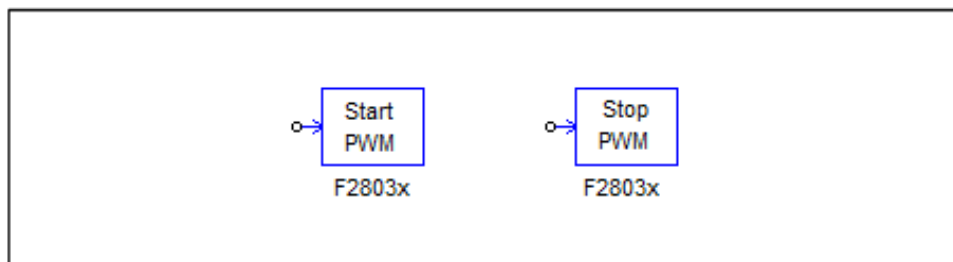
同様にランプ補正を調整する場合、各サイクルで割り込み位置が再計算されます。ランプ補正の調整には時間がかかるので周波数変化

が小さい場合はこのランプ補正の調整は行わないでください。

8.6 Start PWM と Stop PWM

Start PWM と Stop PWM ブロックは、PWM 生成器の動作を開始する、または停止する機能があります。ブロック図とパラメータは下記に示します。

シンボル:



仕様:

パラメータ	機能
PWM ソース	PWM生成器の出力の選択。 PWM1~7、3-phase PWM123と3-phase PWM456、そしてCapture 1が選択できます。

8.7 トリップゾーンとトリップゾーンステート

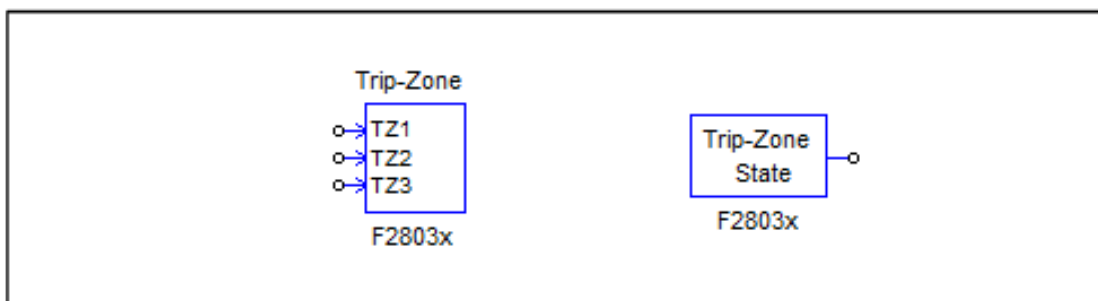
F2803xDSP は 6 つのトリップゾーン入力信号があり、GPIO ポートが 3 つ、コンパレータが 3 つとなっています。コンパレータはトリップ PWM へのデジタル比較器にしか使用できません。

トリップゾーンは外部障害またはトリップ状態を処理するために使用されます。各 PWM 出力は対応した動作をするようにプログラムされます。

一つのトリップゾーン入力信号は複数の PWM 生成器に使用でき、一つの PWM 生成器は 6 つまでのトリップゾーン入力信号を使うことができます。割り込みブロックと共に使えばトリップゾーン信号は割り込み生成用に使えます。

三つの入 GPIO トリップゾーン信号がロー (0) になったとき、トリップ動作をトリガします。デジタル比較器で使うトリップ・ゾーン信号は、指定された (高または低) アクティブレベルでのトリップ動作をトリガします。

シンボル:



仕様 (トリップゾーン) :

パラメータ	機能
トリップゾーン <i>i</i> 使用	トリップゾーン <i>i</i> の使用を定義します。
トリップゾーン <i>i</i> 用のGPIOポート	トリップゾーン入力信号のGPIOポートを指定します。 トリップゾーン1に対するGPIOポート : GPIO12 かGPIO15を選択 します。 トリップゾーン2に対するGPIOポート : GPIO13、GPIO16 かGPIO28を選択しま す。 トリップゾーン3に対するGPIOポート : GPIO14かGPIO17 かGPIO29 を選択しま す。
コンパレータ <i>i</i> 使用	コンパレータ1~3がトリップ・ゾーン入力信号 <i>i</i>

として使用されることを指定します。

仕様（トリップゾーンステート）：

パラメータ	機能
PWM ソース	PWMソースの指定。 - PWM1~7、3-phase PWM123と3-phase PWM456が選択できます。

トリップゾーン割り込みは PWM 生成器のパラメータ通りに One-Shot モードまたは Cycle-by-Cycle モードで使用できます。

Cycle-by-Cycle モードでは、割り込みはそのときの PWM 周期内のみで有効になります。また、One-Shot モードでは入力信号が(0)になったときのみ割り込みによる Trip 時動作が有効になります。割り込みにより出力が一時停止した後、PWM 生成器は再起動を開始するので原則として PWM 出力は続けられます。

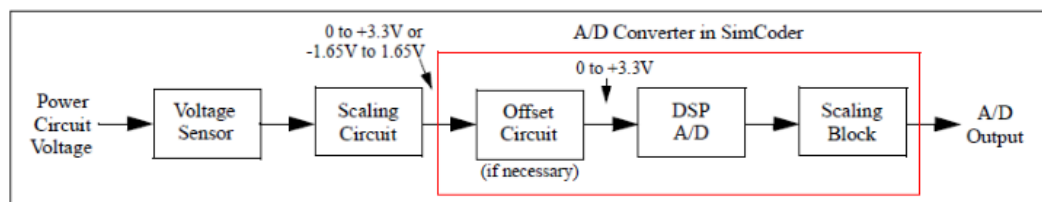
PWM 生成器から割り込みが発生させられたとき、トリップゾーン State ブロックはそれが One-Shot モード、Cycle-by-Cycle モードどちらの Trip 信号なのかを出力します。1 のとき One-Shot モード、0 のとき Cycle-by-Cycle モードとなります。

トリップゾーンと接続して割り込みブロックを使う場合は、割り込みブロックの Device Name のパラメータはトリップゾーンブロック名ではなく、PWM 生成器の名前にしてください。例えば、PWM 生成器の名前が「PWM_G1」、トリップゾーン 1 のブロック名が「TZ1」ならば、割り込みブロックの名前は「TZ1」ではなく「PWM_G1」にしてください。（この場合は割り込みブロックの Channel Number のパラメータは使用しません。）

8.8 A/D 変換器

F2803x には 12 ビット 16 チャンネルの A/D 変換器があります。

一般的に A/D 変換器から DSP へは、主回路の値（電圧、電流、速度など）が取り込まれます。主回路電圧を取り込む例を挙げると、ある程度の値になった主回路電圧は、最初に電圧センサで制御信号に変換されます。その後、回路信号を DSP 入力可能な範囲（0~+3.3V）に変換するため、スケーリング回路があり、必要によっては DC オフセット回路も使われます。DSP 内で信号はデジタル値へ変換され、スケーリングブロックで元の入力信号範囲に復元されます。（下図参照）

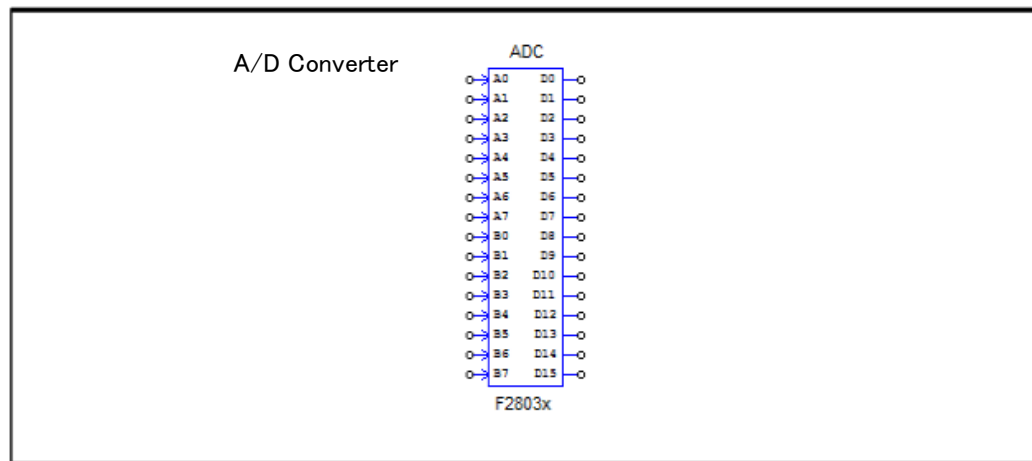


図より分かるように、SimCoder の A/D 変換器は、厳密には実際の DSP 搭載 A/D 変換器と等価ではなく、オフセット機能、A/D 変換機能、スケーリング機能の組み合わせの構成を持っています。これは AC システムアプリケーションが便利に使えるようデザインされています。

多くのアプリケーションで回路変数、特に AC モニタードライブシステムでは AC 信号でモニターされます。これら AC 信号は信号レベルを許容範囲 0 から+3.3V とするために DSP アナログインプットの入力でオフセット回路が回路基板のハードウェアに組み込まれなければなりません。SimCoder の A/D 変換器はそのような場合に便利に使えます。A/D 出力信号のレベルシフトやスケーリングの代わりに SimCoder A/D 変換器のオフセットオプションとスケーリングファクター、の使用を選択するとそれに伴いターゲットコードが生成されます。

※以降は「A/D 変換器」という言葉を使うときは、DSP 搭載の A/D 変換器のことではなく、SimCoder の A/D 変換器ブロックのことを指すこととします。

シンボル:



仕様:

パラメータ	機能
Ch Ai or Bi モード	各A/D変換チャンネルの入力モードの設定。 チャンネルにはAiとBiがあり、それぞれiは0~7の値を取ります。 - AC : このオプションはシミュレーションのためのみです。-1.65Vから+1.65Vの入力が可能なac入力になります。このオプションはA/D変換へのオフセット回路を含んでいます。AC信号を0から3.3Vとする外部シフトが必要な場合に便利です。 - DC : 0Vから+3.3Vの入力が可能なdc入力になります。
Ch Ai or Bi ゲイン	各A/D変換チャンネルAiとBiのゲインkの設定。
Conversion Order	A/D変換器の順番となります。もし定義せずに空欄とした場合は変換はA/Dチャンネルの番号に沿って実行されます。例えばA0,A2, A4, B1とB3が使用されている場合A0,A2, A4, B1とB3の順に実行されます。もし特定のチャンネルを先に実行したい場合、例えばA4,A0,A2,B3,Bとしたい場合はこの順番で設定すると、A0の前にA4、A2の前にA0というように、定義された順番で変換が行われます。
ADCINT1 PIE 選択	ADCINT1割り込みがPIEグループ1或いはPIEグループ10を指定します。
ADCINT2 PIE 選択	ADCINT2割り込みがPIEグループ1或いはPIEグループ10を指定します。

A/D 変換器は 16 チャンネルまであります。SimCoder はそれらのサンプリングレートにしたがってグループに分けられます。最高のサンプリングレートのグループは ADCINT1 割り込みに割り当てられ、2 番目は ADCINT2 割り込みといったように割り当てられていきます。最高サンプリングレートを持つ 2 つの ADC グループは、PIE (周辺割り込み拡張) ベクトルを選択することができます。PIE グループ 1 は PIE グループ 10 よりも高い割り込み優先順位を持ちます。例えば、PWM 割り込みが PIE グループ 3 である場合その割り込み優先順位はグループ 10 より高く PIE グループ 1 よりも低くなります。ユーザは、PWM 割り込み優先順位をある ADC グループの割り込みよりも高く設定したい場合、この ADC グループを PIE グループ 10 に設定する必要があります。

トリガソース :

A/D 変換器は複数のソースからトリガをかけられます。複数の A/D 変換チャンネルは同じトリガソースを共用できます。各々の A/D チャンネルは次のものからトリガをかけることができます。

- PWM 生成器の 1 つ
- Timer1 か Timer2
- 1 つ以上のトリガソース

回路図でもし A/D チャンネルが PWM 生成器と関連していない場合は SimCoder がトリガソースとしてタイマーを選択するよう ZOH ブロックを A/D チャンネルの出力に入れなければなりません。

1 つのソースによってトリガされる A/D 変換器チャンネルはできません。しかし出力信号は違ったサンプリングレートをもつ回路の一部で使用されます。

出力スケールリング :

出力値は下記の数式で与えられます。

$$V_o = k \times V_i$$

V_i は A/D 変換器の入力値です。

入力オフセットとスケールリング :

A/D 変換器の入力値は入力可能範囲内の値を設定してください。入力可能範囲外の値を入力した場合、出力は制限値でクランプされ、ワーニングメッセージを出力します。

A/D 変換器の入力ポートの信号は、DC 入力モードの場合は +3.3V が最大値としてスケールリングされ、AC 入力モードの場合は ±1.65V をピークとしてスケールリングされます。

A/D 変換器の例として DC 入力と AC 入力の 2 つの例を示します。

A/D 変換器チャンネル DC モードの例 :

A/D 変換器は DC モードに設定されており、主回路電圧値が DC で $V_{i_min}=0V$ から $V_{i_max}=150V$

までの値を取ります。A/D 変換機は DC に設定され入力のレンジは 0 から 3.3V となります。実際に入力値 $V_i=100V$ となった場合の例を取り下記に説明を行います。

センサのゲインを 0.01 だとすると実際に入力されてくる値は下記になります。

$$V_{L_max_s} = 150 \times 0.01 = 1.5V$$

$$V_{L_s} = 100 \times 0.01 = 1V$$

最大値を見たとき、センサからは 1.5V が来ますが、DSP 入力可能最大値は 3.3V なのでゲイン 2.2 を追加します。（回路としては $0.01 \times 2.2 = 0.022$ のゲインを持つことになります。）調整後の入力値は下記のようになります。

$$V_{L_max_s_c} = 1.5 \times 2.2 = 3.3V$$

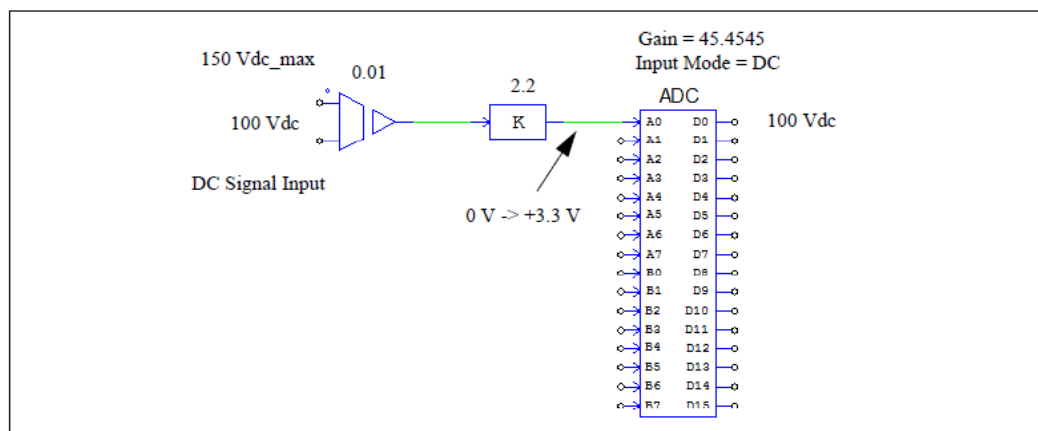
$$V_{L_s_c} = 1 \times 2.2 = 2.2V$$

DSP A/D 変換後のスケーリングブロックの出力は、実際に主回路から来た電圧と合わせるとすると、A/D 変換機のゲインは 45.4545（電圧センサと調整用のゲインの逆数）になります。結果として A/D 変換器からの出力は下記になります。

$$V_{O_max} = 45.4545 \times 3.3 = 150V$$

$$V_O = 45.4545 \times 2.2 = 100V$$

回路図は以下になります。



※上図では比例制御器ブロックのゲインを 2.2 から 1.1、A/D 変換機のゲインを 45.4545 から 90.909 にしてもシミュレーション結果は同じ結果になりますが、コード生成としては適切に行われない可能性がある点ご注意ください。コードは +3.3V が入力の最大値としてスケーリングしますが、この場合、1.5V が最大値になります。入力値の最大値が +3.3V になるようにスケーリングするようにしてください。

AC モードの例：

A/D 変換器は AC モードに設定されており、主回路電圧値が DC で $V_{\text{imax}} = \pm 75\text{V}$ までの値を取るとして、実際に入力値 $\pm 50\text{V}$ となった場合の例を取り下記に説明を行います。

センサのゲインを 0.01 だとすると実際に入力されてくる値は下記になります。

$$V_{L_{\text{max}_S}} = \pm 0.75\text{V}$$

$$V_{L_S} = \pm 0.5\text{V}$$

ピーク値を見たとき、センサからは $\pm 0.75\text{V}$ が来ますが、DSP 入力可能ピーク値は $\pm 1.65\text{V}$ なのでゲイン $2.2(1.65/0.75=2.2)$ を追加します。調整後の入力値は下記のようになります。

$$V_{L_{\text{max}_S_C}} = \pm 1.65\text{V}$$

$$V_{L_{S_C}} = \pm 1.1\text{V}$$

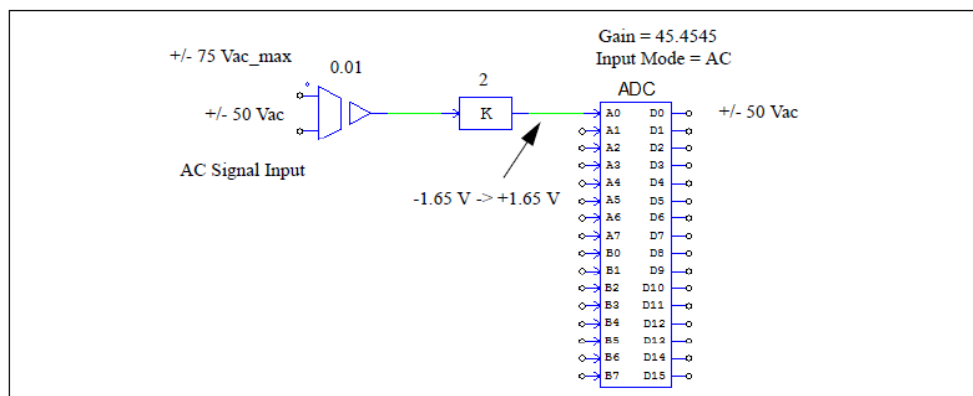
A/D 変換後のスケーリングブロックの出力は、実際に主回路から来た電圧と合わせるとすると、A/D 変換器のゲインは 45.4545（電圧センサと調整用のゲインの逆数）になります。結果として A/D 変換器からの出力は下記になります。

$$V_{O_{\text{max}}} = \pm 75\text{V}$$

$$V_O = \pm 50\text{V}$$

PSIM の A/D チャンネルのゲインは 45.4545 に設定されます。

回路図は以下になります。



※PSIM の回路では直接 AC 信号を A/D 変換器に入力することができます。これは、A/D 入力モードが AC に設定された時、入力範囲は $\pm 1.65\text{V}$ となり、DC オフセットを実行する調整回路の機能は A/D 変換ブロックにすでに含まれています。実際のハードウェアの回路では AC 信号は A/D 変換器からの要求により 0 から 3.3V 範囲内の値となるため AC 信号はスケーリング及びオフセットされます。

8.9 コンパレータ

F2803x は 3 つのコンパレータモジュールがあります。各々のコンパレータブロックは 2 つの外部アナログ入力か、1 つの外部アナログ入力と他の入力となる内部の DAC リファレンスに使用されています。コンパレータ出力は PWM トリップゾーンと GPIO 出力となります。

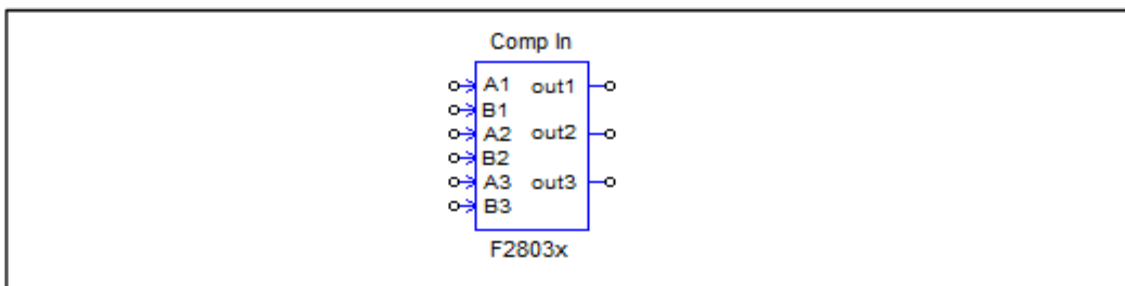
8.9.1 コンパレータ入力

F2803x では 3 つのコンパレータがあります。入力の 3 つのペアは次のような ADC 入力チャネルをもつ ADC/AIP ポートを共用しています。

- Comparator 1 input A - Port ADCA2/AIO2
- Comparator 1 input B - Port ADCB2/AIO10
- Comparator 2 input A - Port ADCA4/AIO4
- Comparator 2 input B - Port ADCB4/AIO12
- Comparator 3 input A - Port ADCA6/AIO6
- Comparator 3 input B - Port ADCB6/AIO14

各々のポートに対して 1 つの機能が対応しています。SimCoder はポートがコンパレータ入力として定義されているにもかかわらず、同じ PSIM 回路図の A/D 変換器のチャネルや AIO として使用されるならば SimCoder はエラーを報告します。

シンボル:



仕様:

パラメータ	機能
コンパレータ <i>i</i>	i番目のコンパレータの出力方法の設定。 - <i>Do not use</i>: 使用しません。 - <i>As a normal comparator</i>: 標準のコンパレータとして使用します。 - <i>As a Trip-Zone signal</i>: trip-zone信号として使用します。
出力論理 <i>i</i>	コンパレータ出力ロジックの設定。 - <i>High when A > B</i>: 入力AがBより大きい時出力はhigh. - <i>High when A < B</i>: 入力AがBより小さい時出力はhigh.
比較方法 <i>i</i>	コンパレータ入力Aと比較する入力Bの設定。 - <i>Compare to Input B</i>: 入力Aを入力Bと比較します。入力Bは外部アナログ信号です。 - <i>Compare to Constant Value</i>: 入力Aを定数と比較します。 - <i>Compare to DAC output</i>: 入力Aを内部信号のDAC出力と比較します。
定数 <i>i</i>	比較方法で定数と比較を選択した場合に有効となります。 定数の範囲は0から3.3Vまでとなります。

すべてのコンパレータで入力 A は常に外部アナログ入力となります。

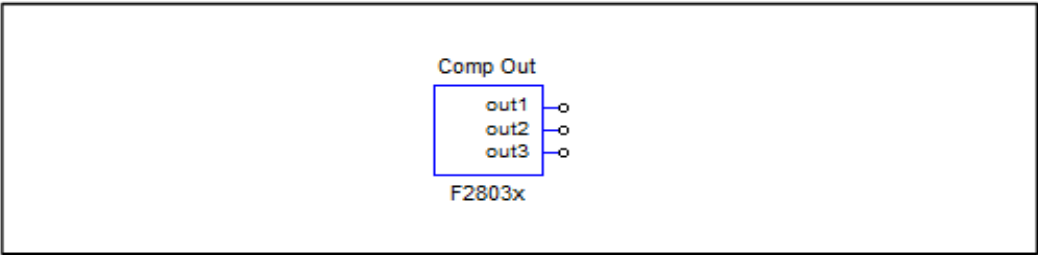
もし入力 B も外部アナログ入力の場合（パラメータの *Compare Method* が *Compare to Input B* として設定される場合）、一致するポートは Hardware Configuration block のコンパレータ入力として設定されなければなりません。

もし、定数値と比較する場合、入力 B は回路図でグランドへ接続する必要があります。DAC 出力と比較する場合は入力 B は DAC 出力と接続されます。どちらの場合も対応した ADC/AIP ポートは他の機能に使用することができます。

8.9.2 コンパレータ出力

コンパレータ出力は GPIO 出力と同様に PWMTrip-zone 信号として使用できます。

シンボル:



仕様:

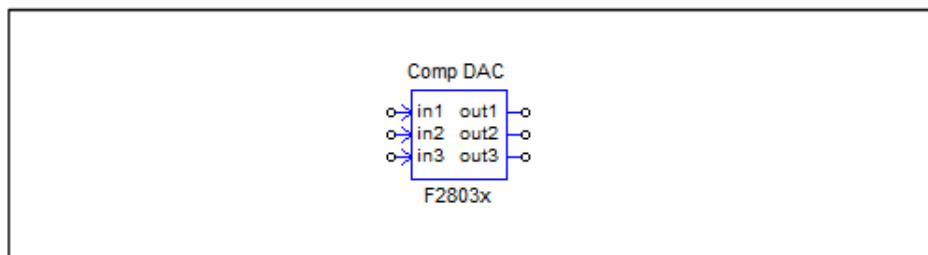
パラメータ	機能
コンパレータ <i>i</i> 出力ポート	<i>i</i> 番目のコンパレータの出力ポート位置。 - コンパレータ1の場合: GPIO1, 20, or 42 - コンパレータ2の場合: GPIO3, 21, 34, or 43 - コンパレータ3の場合: GPIO34

コンパレータ出力ブロックは入力ブロックと一緒に使用されなければなりません。コンパレータ出力チャネルが使用されると対応するコンパレータ入力チャネルも設定されます。

8.9.3 コンパレータ DAC

F2803x の各コンパレータブロックはコンパレータの反転入力(入力 B)により使用可能な 10bit の DAC コンバータを持っています。

シンボル:



仕様:

パラメータ	機能
DAC i レンジ	i番目のコンパレータDACの入力信号範囲：下限値は0です。
Use Ramp Generator	ランプジェネレーターを使用します。 ランプジェネレーターはコンパレータDACがランプジェネレータ用に設定されません。
Total Ramp Compensation	1つのPWM周期におけるランプジェネレーターの総補正值。1サイクルでのランプ総減少を表わしています。

コンパレータ DAC の出力はコンパレータ入力ブロックの相応反転入力(InputB)に接続できません。ノード out1 はコンパレータ入力ブロックのノード B1 に、ノード out2 はノード B2 へノード out3 はノード B3 へのみ接続できます。

コンパレータ DAC ブロックはコンパレータ入力ブロックと連携して使用してください。単独では使用できません。

ランプジェネレーター機能を使用しない場合、入力値は直接 DAC 出力へ瞬時に適用されます。出力範囲は 0 から 3.3V で出力は次のように計算されます。

DAC 出力信号 = DAC 入力信号 * 3.3 / DAC レンジ
ランプジェネレーター機能を使用する場合、入力値は保存され次の PWM 周期の初期出力値として使用されます。コンパレータ DAC 出力は PWM 期間内に直線的に減少し、総減少量は総ランプ補償値と同等となります

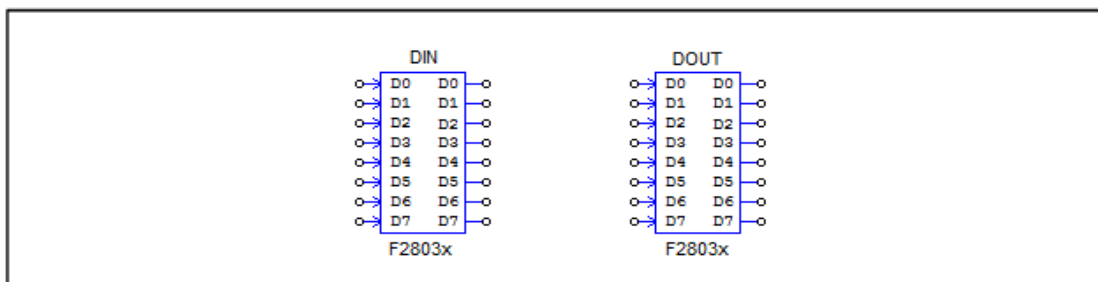
ランプジェネレーター機能を使用する場合、コンパレータ DAC 入力に関連したサンプリングレートはコンパレータに使用されている PWM generator の周波数と同じでなければなりません。

8.10 デジタル入力とデジタル出力

F2803x は 45 本の汎用入出力ポート (GPIO0~44) をデジタル入力、またはデジタル出力に設定可能です。6 本のデジタル入力と出力に使用できるデジタル・アナログ入出力ポート (AIO2,4,6,10,12 と 14) があります。

SimCoder では 8 チャンネルのデジタル入出力持つブロックを使うことができます。複数の 8 チャンネルのデジタル入出力ブロックは同じ回路上で使用できます

シンボル:



仕様（デジタル入力）:

パラメータ	機能
入力ポート <i>i</i> の位置	入力iのポート位置で、iは0~7です。 GPIOポート0~44かAIOポート2,4,6,10,12,14から選択できます。
外部割込みとして使用	外部割込み入力の設定。

仕様（デジタル出力）:

パラメータ	機能
出力ポート <i>i</i> の位置	出力ポートiのポート位置（0~7）です。GPIOポート0~44かAIOポート2,4,6,10,12,14から選択できます。

※GPIO ポートを入力ポートとした場合、そのポートは他のペリフェラルのポートとしては使えない点にご注意ください。もし IO ポートがデジタル入力として使用される場合、他のデジタル出力やリフェラルのポートとしては使用できません。例えば GPIO1 はデジタル入力ポートと PWM1 出力の両方の設定ができますが、同時に設定した場合、エラーが出力されます。

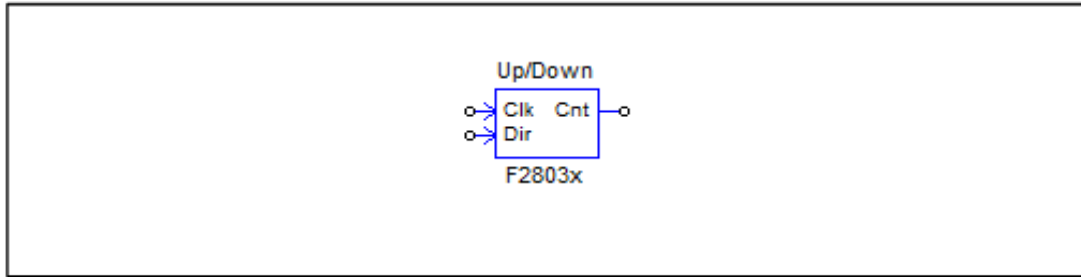
F2803xDSP は 3 つのマスクされた外部割込み(XINT1~XINT3)をサポートしています。外部割込みの専用のピンはありません。X XINT1~XINT3 の割り込みは GPIO0~31 ピンの入力として使用できます。

8.11 UP/DOWN カウンタ

F2803x は UP/DOWN カウンタを持っています。入力ポートは

- GPIO20 はクロック信号を指定します。
- GPIO21 はカウント方向を指定します。

シンボル:



仕様:

“Clk”はクロック信号入力、“Dir”はカウンタがどちらへ進むかを指定します。“Dir”が1のときカウンタは増加し0の時は減少します。

UP/DOWN カウンタブロックの出力はカウンタ値となります。

UP/DOWN カウンタはエンコーダと同じリソースを使用しているため同じ GPIO ポートは使用できません。例えば UP/DOWN カウンタ1 とエンコーダ1 は同時に使用できません。

8.12 エンコーダとエンコーダステート

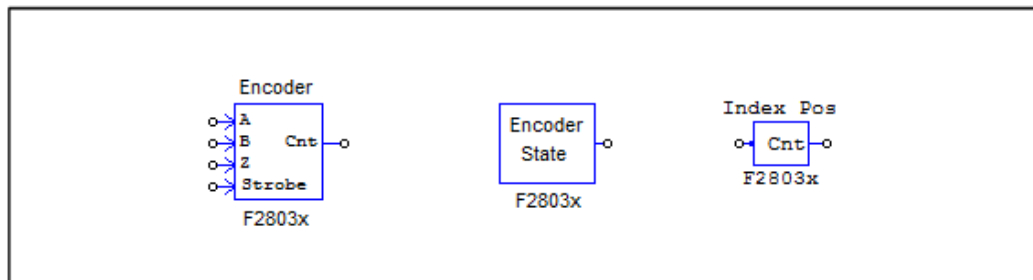
F2803x はエンコーダモジュールを利用できます。

- GPIO 20 クロック入力
- GPIO 20 方向入力
- GPIO 20 Z(もしくは基準信号)入力
- GPIO 20 ストローブ信号入力

エンコーダステートブロックはどの入力信号（基準信号とストローブ信号）が割り込みを発生させる設定なのかを見ることに使います。ハードウェアの割り込みはZ信号（基準信号）とストローブ信号から生成され、エンコーダステートの出力はどの信号が割り込み発生させているかを示しています。エンコーダステートブロックの出力が0のときは、Z信号（基準信号）から割り込みが発生する設定になっていることを示し、1のときはストローブ信号から割り込みが発生する設定になっていることを示します。

エンコーダインデックス/ストロボ位置ブロックはエンコーダの初期位置をラッチするのに使用されます。この素子の入力が0の時、エンコーダカウンタは0に設定されます。入力が1になるとエンコーダは、インデックス/ストロボイベントの発生を待ち、発生するとエンコーダのカウンタ値をラッチします。

シンボル:



仕様（エンコーダ）:

パラメータ	機能
Z信号使用	Z信号（基準信号）の有効無効設定。 - No: 使用しない。 - Yes(rising edge): 立上り信号エッジを使用します。 - Yes(falling edge): 立下り信号エッジを使用します。
ストロブ信号使用	ストロブ信号の有効無効設定。 - No: 使用しない。 - Yes(rising edge): 立ち上がり信号エッジを使用します。 - Yes(rising/falling edge): 立ち上がり立下り両方の信号エッジを使用します。
カウンタ方向	カウンタの方向の設定。 - Forward : カウンタは増加方向に進みます。 - Reverse : カウンタは減少方向に進みます。
Z 信号極性	Z信号のトリガの極性 - Active High : 正論理 - Active Low : 負論理
ストロブ信号極性	ストロブのトリガ極性 - Active High : 正論理 - Active Low : 負論理
エンコーダ分解能	接続するエンコーダの分解能の設定。 0に設定した場合、リセットされません。4096に設定した場合、カウンタ値が4095になった次のタイミングで0にリセットされます。

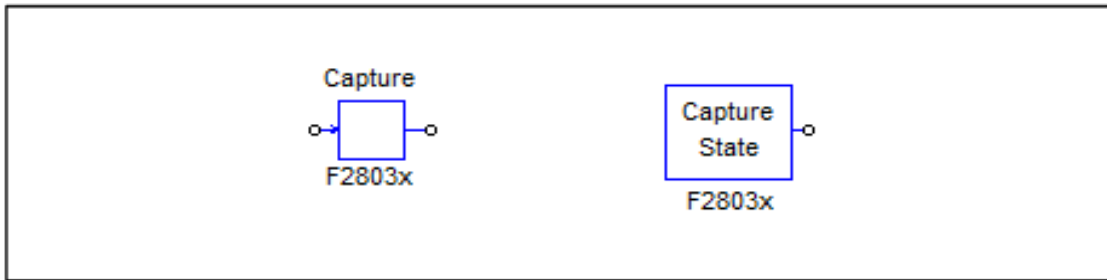
仕様（エンコーダインデックス／ストロボ位置）:

パラメータ	機能
ラッチ位置	ラッチカウンタタイプの選択 - Index Pos: Z方向信号が使用されます。 - Strobe Pos: Strobe信号が使用されます。
位置タイプ	ラッチ位置の選択 - 最初のラッチ位置 - 現在のラッチ位置

8.13 キャプチャとキャプチャステート

F2803x は強化されたキャプチャモジュールを持っています。割り込みを発生させることができます。割り込みブロックで割り込みトリガモードの設定ができます。

シンボル:



仕様（キャプチャ）:

パラメータ	機能
キャプチャソース	キャプチャとポートの設定。次の3つのGPIOポートの1つをソースとして設定できます。 - Capture1 (GPIO5) - Capture2 (GPIO19) - Capture3 (GPIO24)
イベントフィルタ	イベントフィルタのプリスケール設定。 入力信号は選択したプリスケールで分けられます。
タイマモード	キャプチャカウンタタイマモード設定。 Absolute time(絶対時間)またはTime difference(時間差)のどちらかを設定できます。

仕様（キャプチャステート）:

パラメータ	機能
キャプチャソース	キャプチャソースです。 Capture1が利用可能な唯一のソースです。

キャプチャステートブロック出力は1か0で、1は立ち上がり信号エッジ、0は立ち下がり信号エッジとなります。

8.14 シリアル通信インターフェース（SCI）

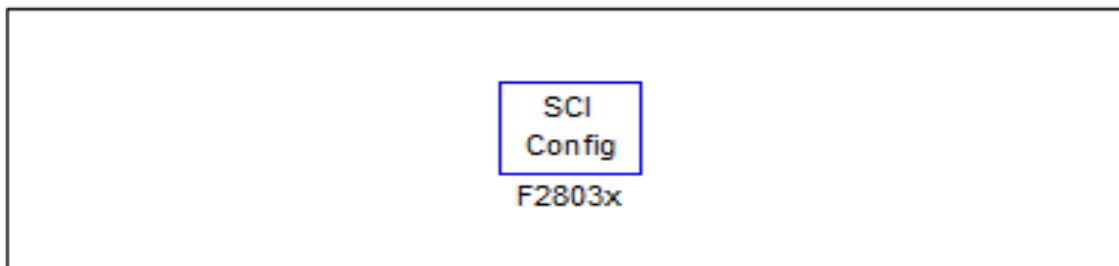
F2803x は、シリアルコミュニケーションインターフェース（SCI）のための関数が用意されています。SCI を介して、DSP 内部のデータは、外部 RS-232 ケーブルを使用してコンピュータに転送することができます。PSIM は、DSP とコンピュータ両方のデータの送受信およびコンピュータ上にデータを表示するための必要なすべての関数を提供します。これは DSP のコードをリアルタイムでモニタ、デバッグ、および調整するために非常に便利な方法を提供します。

以下に説明しますが、SimCoder では SCI 設定、SCI 入力、および SCI 出力の 3 つの SCI の機能ブロックが用意されています:

8.14.1 SCI 設定

SCI 設定ブロックは、SCI ポート、通信速度、パリティチェックのタイプ、およびデータバッファサイズを定義します。

シンボル:



仕様:

パラメータ	機能
ポート選択	SCIポートの設定。SCIに使用できる次のGPIOポートがあります。 - SCIA : GPIO28とGPIO7に対する組み合わせとしてGPIO29とGPIO12があります。
通信速度(bps)	SCI通信速度[bps]。プリセット速度は、200000、115200、57600、38400、19200、9600 [bps]が用意されています。または手動で他の速度を指定することができます。
パリティチェック	通信エラーチェック用のパリティチェックの設定。 None(無効)、Odd(奇数)、Even(偶数)のいずれかを選択できます。
出力バッファサイズ	SCI用にDSPに割り当てられたデータバッファのサイズ。バッファはRAM領域に位置しており、各バッファの要素は3つの16ビットワード（データポイントごとに、6バイト、または48ビットである）から成る1つのデータポイントを格納しています。

バッファサイズを正しく選択しなければならないことに注意してください。

より多くの変数を長い期間にわたって監視できるように、より多くのデータポイントを収集するためには大きなバッファが好ましい。一方、内部 DSP のメモリは限られており、バッファは通常の DSP の動作を妨げるほど大きすぎはいけません。

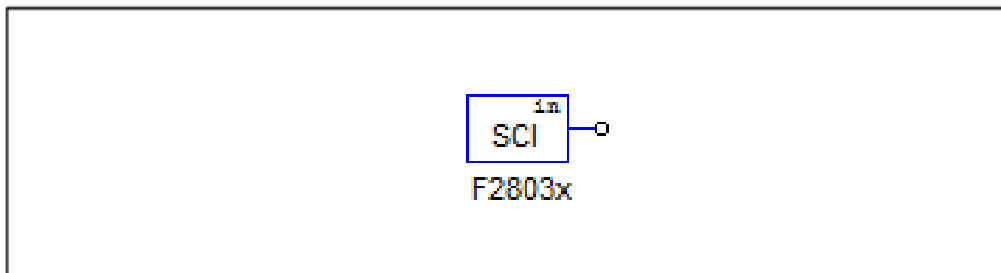
バッファサイズを選択する方法の詳細については、"Tutorial - Using SCI for Waveform Monitoring.pdf"を参照してください。

8.14.2 SCI 入力

SCI 入力ブロックは、変更可能な DSP コードで変数を定義するために使用されます。SCI 入力変数の名前は、DSP のオシロスコープ（ユーティリティのメニューの下にあります）に表示され、その値は SCI を介して実行時に変更することができます。

SCI 入力ブロックは、例えば、リファレンスの変更、制御パラメータの微調整のための便利な方法を提供します。

シンボル:



仕様:

パラメータ	機能
初期値	SCI入力変数の初期値。

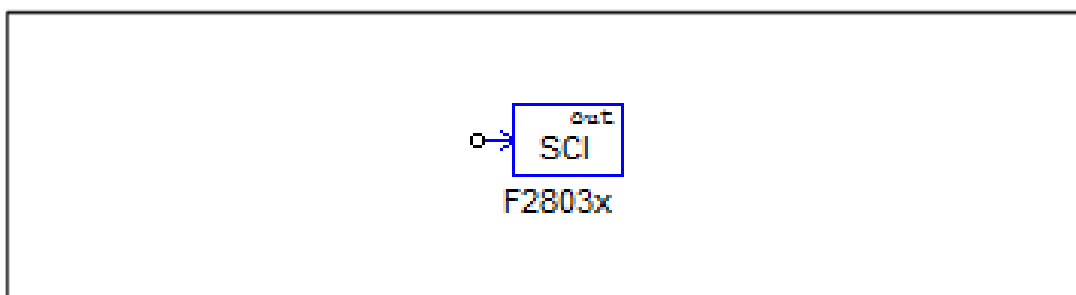
回路図では、SCI 入力は定数として動作します。コードは DSP 上で実行されている間に、ユーザーはこの値を変更することができ、値はシミュレーションの初期値として固定されます。

8.14.3 SCI 出力

SCI 出力ブロックは、表示用の変数を定義するために使用されます。SCI 出力ブロックがノードに接続されている場合、SCI 出力ブロックの名前は、DSP のオシロスコープ（ユーティリティのメニューの下にあります。）に表示され、この変数のデータは、実行時に SCI を介してコンピュータに DSP から送信することができます。波形は DSP オシロスコープで表示することができます。

SCI の出力ブロックは、DSP 波形を観測するための便利な方法を提供します。

シンボル:



仕様:

パラメータ	機能
データポイントステップ	データの収集頻度を定義します。[Data Point Step]が1の場合、すべてのデータポイントが収集され送信されます。例えば、[Data Point Step]が10の場合、10ポイントのうちの1ポイントのみが収集され送信されます。

[Data Point Step]が小さすぎると、データポイントが非常に多くなる可能性があり、すべてを送信できない場合があることに注意してください。この場合、いくつかのデータポイントがデータの送信時に破棄されます。

また、[Data Point Step]のパラメータは DSP オシロスコープが連続モードの時のみ使用されます。スナップショットモードにある場合、このパラメータは無視され、すべてのポイントを収集して送信されます。

シミュレーションでは、SCI 出力は電圧プローブとして動作します。

8.15 シリアルペリフェラルインターフェース(SPI)

F2803x はシリアルペリフェラルインターフェース (SPI) のための関数が用意されています。TI F803x ターゲットライブラリの SPI ブロックによって、容易かつ便利に外部の SPI デバイス (外付け A / D および D / A コンバータなど) と通信する機能を実装することができます。SPI デバイスのために手動でコードを書くことは、しばしば時間がかかり、大変な作業です。SPI をサポートする機能によって PSIM は大幅に簡素化し、コーディングとハードウェアの実装プロセスをスピードアップします。

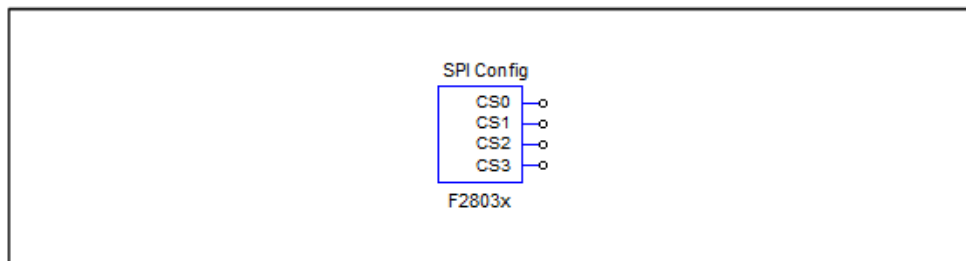
SPI ブロックを使用する方法についての詳細な説明については、ドキュメントを参照してください"
Tutorial - Using SCI for Real-Time Monitoring.pdf".

SimCoder では SPI 設定、SPI デバイス、SPI 入力、SPI 出力の 4 つの SCI の機能ブロックが提供されています。

8.15.1 SPI 設定

SPI 設定ブロックは、SPI ポート、チップ選択ピン、および SPI のバッファサイズを定義します。SPI が使用されている回路図に必要で、ブロックはメイン回路図内に配置される必要があります。

シンボル:



仕様:

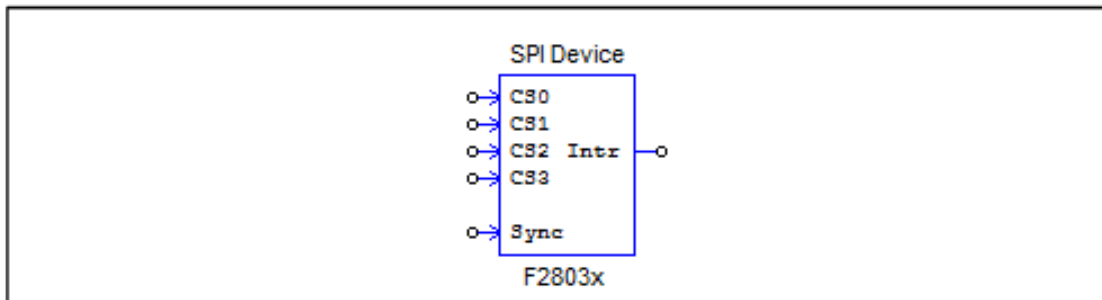
パラメータ	機能
SPI ポート選択	SPIポートの設定。 -SPIA (GPIO 16-19) -SPIA (GPIO 3,5,18,19) -SPIB (GPIO 12-15) -SPIB (GPIO 24-27)
チップセレクト端子 0,1,2,3	チップセレクトピンのGPIOポート。PSIMは、最大で16のSPIデバイスを使用することができます。これらはチップセレクトピンPin0～Pin3で定義されている4つのGPIOピンを選択する必要があります。これらのGPIOポートとSPIスレーブ送信許可ピンSPISTEは、チップセレクト信号を生成するために使用されます。

キューサイズ (SPI Buffer Size)	SPI通信バッファサイズ。バッファの各メモリセルは、SPIコマンドのインデックスが保存されます。通常は、すべてのSPIの入力/出力要素の中でSPIコマンド（例えば、Start Conversion Command, Receiving Data Command, Sending Data Command, and Sync. Command）の数に1を足したバッファサイズを指定します。
--------------------------------	--

8.15.2 SPI デバイス

SPI デバイスのブロックは、対応する SPI ハードウェアデバイスの情報を定義します。回路図の SPI デバイスのブロック数は、SPI ハードウェアデバイスの数と同じでなければなりません。

シンボル:



仕様:

パラメータ	機能
チップセレクト 端子	SPIデバイスに対応するチップセレクトピンの状態。チップセレクトピンがこの設定と一致する場合にこのSPIデバイスが選択されます。
通信速度 (MHz)	SPIの通信速度 (MHz)
クロックタイプ	SPIハードウェアデバイスによって決定されるSPIクロックの種類。 -立ち上がりエッジ（遅れなし）： クロックは通常はLOW(0)で、データはクロックの立ち上がりエッジでラッチされます。 -立ち上がりエッジ（遅れあり）： クロックは通常はLOW(0)でデータは遅延付きのクロックの立ち上がりエッジでラッチされます。 -立ち下がりエッジ（遅れなし）： クロックは通常はHigh(1)でデータはクロックの立ち下がりエッジでラッチされます。 -立ち下がりエッジ（遅れあり）： クロックは通常はHigh(1)でデータは遅延付きのクロックの立ち下がりエッジでラッチされます。
コマンドワード 長	SPI通信コマンドのワード長、または有効ビットの長さ。1～16ビットにすることができます。
同期信号タイプ	SPIデバイスの同期信号のトリガモード。立ち上がりエッジまたは立ち下がりエッジのいずれかです。
SPI 初期コマン ド	SPIデバイスの初期コマンドです。
ハードウェア割 り込みモード	SPIデバイスが生成する割り込み信号の種類を指定します。SPIデバイスの割り込み出力ノードがデジタル出力要素の入力に接続されている場合にのみ有効です。次の

	いずれかになります。 - ハードウェア割り込みなし - 立ち上がりエッジ - 立ち下がりエッジ
割り込みタイミング	変換完了時のSPIデバイスの割り込み生成方法を指定します。次のいずれかになります。 - 割り込みなし 割り込みは生成されません。この場合、DSPはSPI入力デバイスにコマンドを送信します。デバイスは変換を開始して同じコマンドで結果を返します - 連続多重割り込み： 多重割り込みは、各変換の後に連続で生成されます。 1つのA/D変換ユニットと複数の入力チャンネルを持つSPIデバイス用です。この場合、DSPが最初に変換のコマンドを送信し、SPIデバイスの変換が開始されます。変換が完了すると、SPIデバイスが割り込みを生成します。割り込みサービスルーチンでは、DSPが変換結果をフェッチするためにコマンドを送信し、同じSPI入力デバイスの別のチャンネルの新しい変換を開始します。 - フントタイム割り込み： 変換終了後に1回だけ割り込みが生成されます。1つのリクエストで複数のチャンネルの変換を実行可能なSPIデバイス用です。この場合、DSPはSPI入力デバイスにコマンドを送信し、SPIデバイスは複数の入力チャンネルの変換を完了します。すべての変換が完了すると、SPIデバイスが割り込みを生成します。
コマンド間ギャップ (ns)	2つのSPIコマンドの間隔で単位[nsec]です。
変換シーケンス	変換シーケンスを決定するためのコマンドで区切られたSPIの入力要素の名前を定義します。このパラメータは、SPIデバイスが連続で複数の割り込みを生成する場合にのみ有効であることに注意してください。

回路図では、チップセレクトロジックが実装される方法を定義せずに、すべての SPI デバイスのチップセレクトピンは SPI 設定ブロックのチップセレクト端子に接続されています。しかし、実際のハードウェアでは、対応するチップセレクトロジックに合わせて実装する必要があります。

SPI のコマンドは、カンマで区切られた 16 ビットの連続の数字で構成されています。16 ビットの数値では、下位ビットだけがコマンドによって使用される有効ビットです。例えば、コマンドワード長が 8 の場合、ビット 0～7 はコマンドであり、ビット 8～15 は使用されません。

SPI デバイスは、入力デバイスまたは出力デバイスのいずれかです。
例えば、外部 A/D コンバータは、入力デバイスです。通常、DSP はデバイスに 1 つまたは複数の A/D 変換のコマンドを送信し、変換を開始するために同期信号を設定します。同期信号は、同じデバイスの次のコマンドでリセットされます。

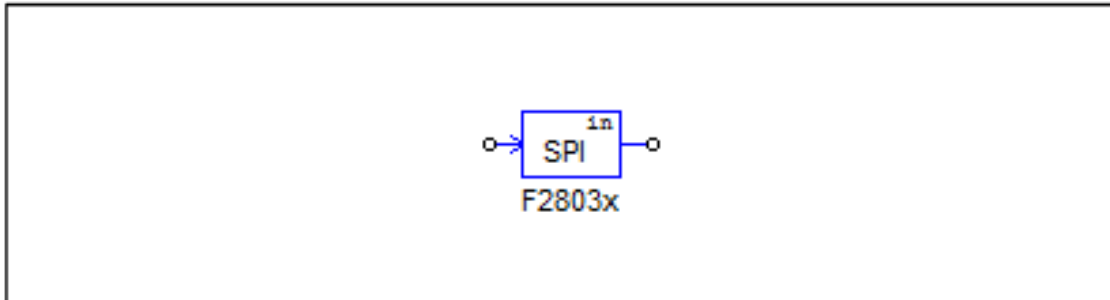
同期信号を使用した SPI の入力デバイスは、通常、割り込みサービスルーチンを入力して DSP をトリガする割り込みピンを必要とします。

一方、外部 D/A コンバータは出力デバイスです。通常、DSP はデバイスに 1 つまたは複数の D/A 変換のコマンドを送信し、変換を開始するために同期信号を設定します。同期信号は、同じデバイスの次のコマンドでリセットされます。

8.15.3 SPI 入力

SPI 入力デバイスは、複数の入力チャンネルを持つことができます。SPI 入力ブロックは、SPI 通信用の入力チャンネル、および 1 つの入力チャンネルに対応する 1 つの SPI 入力ブロックのプロパティを定義するために使用されます。

シンボル:



仕様:

パラメータ	機能
デバイス名	SPI入力デバイスの名前です。
変換開始コマンド	コンマで区切られた16進の数字で、変換を開始するコマンドです。 (例 ; 0x23、0x43、0x00)
データ受信コマンド	コンマで区切られた16進の数字で、受信を開始するコマンドです。 (例 ; 0x23、0x43、0x00)
データビット位置	データビットが受信データの文字列のどこにいるか定義します。形式は次のとおりです。 <i>ElementName</i> = { <i>Xn</i> [<i>MSB</i> .. <i>LSB</i>]} ここで、 - “ <i>ElementName</i> ”はSPIの入力デバイスの名前です。もし現在のSPI入力デバイスの場合は、代わりに <i>y</i> を使用してください。 - {}は、括弧内の項目が複数回繰り返されることを意味します。 - <i>Xn</i> はSPIの入力装置から受信した <i>n</i> 番目の単語であり、 <i>n</i> は0から始まります。 - <i>MSB</i> .. <i>LSB</i> は、ワードの有効ビットの位置を定義します。
入力範囲	入力範囲を定義するパラメータ <i>Vmax</i> を指定します。このパラメータは、SPIデバイスがA/Dコンバータの場合にのみ有効です。デバイスの変換モードがDCの場合、入力範囲は0～ <i>Vmax</i> 。デバイスの変換モードがACの場合、入力範囲は- <i>Vmax</i> /2～ <i>Vmax</i> /2。
スケール係数	出力スケールファクタの <i>Kscale</i> 。スケールファクタが0の場合、SPIデバイスは、A/Dコンバータではなく、結果はDSPがSPI通信で受信したものとまったく同じになります。そうでない場合は、SPIデバイスはA/Dコンバータであり、結果はこの係数とA/D変換モードに基づいてスケールリングされます。
AD変換モード	デバイスのA/D変換モード。変換モードはDCまたはACのどちらでもかまいません。このパラメータはデバイスがA/Dコンバータの場合にのみ有効であることに注意してください。
初期値	入力の初期値です。

データのビット位置の式は、SPI 入力デバイスのデータ長を定義します。例えば、
 $y = x1[3..0]x2[7..0]$ は、データ長が 12 であることを意味し、結果は 2 ワード目の下位 4 ビットと 3 番目のワードの下位 8 ビットです。受信データの文字列は 0x12、0x78、0xAF の場合、結果は 0x8AF です。

スケールファクタが0でない場合、出力は以下に基づいて拡大縮小されます。

DC 変換モードの時：

-シミュレーションの場合 $Output = Input \cdot K_{scale}$

-実際のハードの場合 $Output = \frac{Result \cdot V_{max} \cdot K_{scale}}{2^{Data_Length}}$

AC 変換モードの時：

-シミュレーションの場合 $Output = Input \cdot K_{scale}$

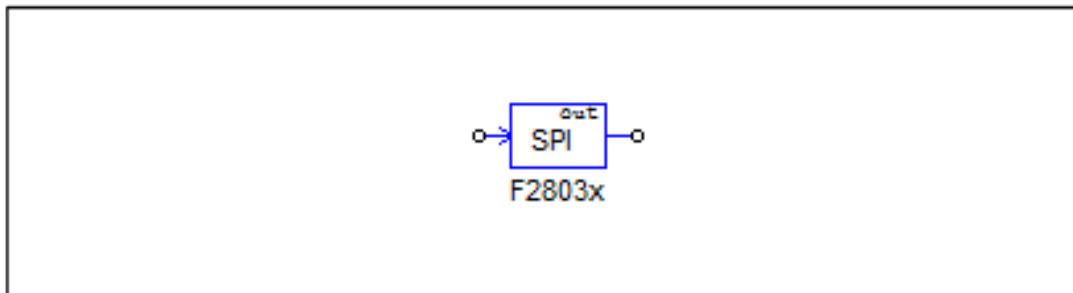
-実際のハードの場合 $Output = \frac{(Result - 2^{Data_Length-1}) \cdot V_{max} \cdot K_{scale}}{2^{Data_Length-1}}$

パラメータ Data_length はデータのビット位置の計算式で計算されます。

8.15.4 SPI 出力

SPI 出力デバイスは、複数の出力チャンネルを持つものもあります。SPI 出力ブロックは、SPI 通信用の出力チャンネル、1 つの出力チャンネルに対応する 1 つの SPI 出力ブロックのプロパティを定義するために使われます。

シンボル:



仕様:

パラメータ	機能
デバイス名	SPI出力デバイスの名前です。
スケール係数	出力スケール係数Kscale。スケールファクタが0の場合、SPIデバイスは、D/Aコンバータではなく、結果はDSPがSPI通信で受信したものとまったく同じになります。そうでない場合は、SPIデバイスはD/Aコンバータであり、結果はこの係数とD/A変換モードに基づいてスケールリングされます。
出力範囲	出力範囲を定義するパラメータVmaxを指定します。このパラメータは、SPIデバイ

	スがD/Aコンバータの場合にのみ有効です。デバイスの変換モードがDCの場合、入力範囲は0～Vmax、デバイスの変換モードがACの場合、入力範囲は-Vmax/2～Vmax/2。
DAC モード	デバイスのD/A変換モード。変換モードはDCまたはACのどちらでもかまいません。このパラメータはデバイスがD/Aコンバータの場合にのみ有効であることに注意してください。
データ送信コマンド	カンマで区切られた16進の数字で、出力データを送信するコマンドです。 (例：0x23、0x43、0x00)
データビット位置	データビットが送信データの文字列のどこにいるか定義します。形式は次のとおりです。 素子名(ElementName) = {Xn[MSB..LSB]} ここで、 - “素子名”はSPI出力デバイスの名前です。もし現在のSPI出力デバイスの場合は、代わりにyを使用してください。 - {}は、括弧内の項目が複数回繰り返されることを意味します。 - XnはSPI出力デバイスに送信したn番目の単語であり、nは0から始まります。 - MSB..LSBは、ワードの有効ビットの位置を定義します。
同期コマンド	カンマで区切られた16進の数字で、SPI出力デバイスの出力チャネルと同期させるコマンドです。(例：0x23、0x43、0x00) このコマンドは、SPI出力デバイスが同期信号を持っていない場合に使用されます。

データのビット位置の式は、SPI 出力デバイスのデータ長を定義します。例えば、y= x1[3..0]x2[7..0]は、データ長が 12 であることを意味し、結果は 2 ワード目の下位 4 ビットと 3 番目のワードの下位 8 ビットです。送信データの文字列が 0x12、0x78、0xAF の場合、データは 0x8AF です。

スケールファクタが 0 でない場合、出力は以下に基づいて拡大縮小されます。

DC 変換モードの時：

$$\begin{aligned} \text{--シミュレーションの場合} \quad & Output = Input \cdot K_{scale} \\ \text{--実際のハードの場合} \quad & Output = \frac{Result \cdot K_{scale} \cdot 2^{Data_Length}}{V_{max}} \end{aligned}$$

AC 変換モードの時：

$$\begin{aligned} \text{--シミュレーションの場合} \quad & Output = Input \cdot K_{scale} \\ \text{--実際のハードの場合} \quad & Output = 2^{Data_Length} + \frac{Result \cdot K_{scale} \cdot 2^{Data_Length-1}}{V_{max}} \end{aligned}$$

パラメータ Data_length はデータのビット位置の計算式で計算されます。

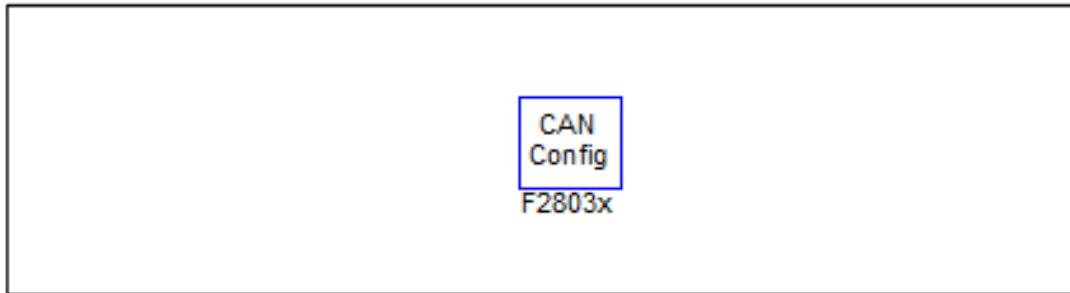
8.16 Controller Area Network (CAN) Bus

F2803x は CAN バスコミュニケーションの機能をもっています。PSIM では CAN バスに必要な機能を備えています。SimCoder には CAN Configuration、CAN Input、CAN output の 3 つの機能があります。

8.16.1 CAN Configuration

CAN Configuration ブロックは CAN バスソースでデータバイト順や他の設定を定義します。F2803xDSP には 1 つの CAN ソース CAN A があり、送信用の GPIO31 と受信用の GPIO30 を持っています。

シンボル:



仕様:

パラメータ	機能
CAN 速度 (MHz)	Communication Speed (単位はMHz)。 次の周波数、125kHz、250kHz、500kHz、1MHzのどれか1つを選択するかもしくは手入力で設定できます。手入力の場合は1MHzを超えないようご注意ください。
データバイト順	データバイトの順番になります。次の1つとなります。 - Least significant byte 1st - Most significant byte 1st “Least significant byte 1st” は最下位バイトが最初に転送され、最上位バイトが最後に転送されます。一方 “Most significant byte 1st”は最上位バイトが最初に転送され、最下位バイトが最後に転送されます。
入力メールボックス数	入力mailboxの数
受信メッセージ紛失検査	もしEnable(有効)に設定されると受信メールが無くなった場合、エラーレポート関数の “_ProcCanAErrReport(nErr)” (CAN Aに対して) が呼ばれます。 この関数はエラー状態nErrの数値を返します。 もし、Disable(無効)に設定されるとエラーレポート関数は呼ばれません。
バスオフ検査	もしEnable(有効)に設定されると受信メールが無くなった場合、エラーレポート関数の “_ProcCanAErrReport(nErr)” (CAN Aに対して) が呼ばれます。 この関数はエラー状態nErrの数値を返します。 もし、Disable(無効)に設定されるとエラーレポート関数は呼ばれません。
エラー検査モード	エラーチェックモードを受動か能動かに設定できます。受動の場合、エラーカウン트가128になったときに割り込みが発生します。能動の場合割り込みが毎回発生します。

関数 “_ProcCanAErrReport(nErr)” (CAN A に対して) の戻り状態 nErr は 32 ビットの整数です。

エラーとレジスターCANES 状態からの値を含んでいます。関数 “_ProcCanAErrReport(nErr)”からの値が

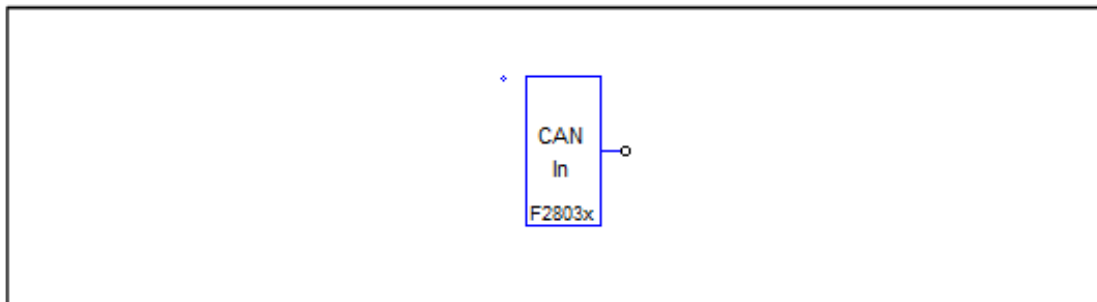
戻るとレジスターCANES はクリアされます。

もし、特定のエラーに対応したい場合は “__ProcCanAErrReport(nErr)”関数に独自のコードを追加することができます。

8.16.2 CAN Input

CAN 入力ブロックは CAN バスから CAN メッセージを受け取ります。

シンボル:



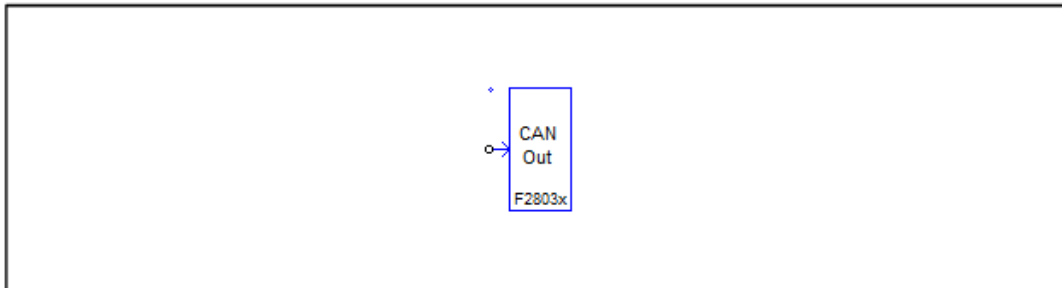
仕様:

パラメータ	機能
入力信号数	CAN入力数です。8入力まで可能です。
拡張ID使用	Yesに設定された場合、メッセージのIDは29-bitの整数となります。 Noに設定された場合、メッセージのIDは11-bitの整数となります。
メッセージID	メッセージのIDです。整数で例えば0x23のようになります。
ローカルマスク	メッセージに対するMASKとなります。整数です。もしメッセージIDのビットがMASKのビットと一致するとメッセージを受信します。そうでない場合は無視されます。例えばmaskが0x380でメッセージIDが0x389の場合、このメッセージが受信されます。もし、メッセージIDが0x480の場合はメッセージは無視されます。
上書き保護フラグ	Allow overwrite(上書き可)かDo not allow overwrite(上書き不可)に設定できます。例えばメッセージ受信用にメールボックスが設定され、新しいメッセージがCANバス経由で受信されたが古いメッセージがまだメールボックスにあり、何も処理されていない場合、上書き保護フラグが"Allow overwrite"に設定されると新しいメッセージが受信され古いメッセージは上書きされます。Flagが"Do not allow overwrite"に設定されていると新しいメッセージは受信されずに古いものはそのまま残ります。
受信頻度	特定期間に入力ブロックで受信したメッセージの数。例えば、最初の入力ブロックで1秒間に20のメッセージを、2番目の入力ブロックで1秒間に30のメッセージを受信する2つの入力ブロックがあるとします。その場合"Receive Message Rate"は最初のブロックに20、2番目のブロックに30と設定されます。
入力 i ゲイン	i 番目の入力に対するゲインでiは1から8で設定できます。出力は入力のゲインの値をかけた値になります。
入力 i データ開始位置	メッセージは8データポイントまで可能です。データポイントは1bitから32bitsまで可能です。これはメッセージの現在のデータポイントの開始位置を指定します。
入力 i データ終了位置	これはメッセージの現在のデータポイントの終了位置を指定します。
入力 i データタイプ	データタイプは、浮動、整数、またはIQ1からIQ30までとなります。
入力 i デフォルトデータ	SCI入力変数の初期値となります。

8.16.3 CAN Output

CAN 出力ブロックは CAN メッセージを CAN バスへ送ります。

シンボル:



仕様:

パラメータ	機能
出力信号数	CAN出力の数です。8出力まで可能です。
拡張ID使用	Yesに設定された場合、メッセージのIDは29-bitの整数となります。 Noに設定された場合、メッセージのIDは11-bitの整数となります。
メッセージ ID	メッセージのIDです。整数で例えば0x23のようになります。
トリガタイプ	トリガのタイプは次のようになります。 - No trigger : トリガなし。 - Rising edge : トリガソースの立ち上がりエッジでトリガがかかります。 - Falling edge : トリガソースの立下りエッジでトリガがかかります。 - Rising/Falling edge: トリガソースの立ち上がりと立下りのエッジでトリガがかかります。 立ち上がり／立下りエッジはトリガソースの現在の値と、最終のトリガの値が1以上で差がある場合に適用されます。
トリガソース	トリガソースはトリガタイプによるグローバル変数の定数にもあります。 トリガタイプが “No Trigger”に設定されるとトリガソースはカウンターリミットを定義します。例えばトリガソースが定数かグローバル変数で値が5の時、5回に 1 回トリガがかかることを意味しています。すなわちデータが5サイクル毎に送られます。 もしトリガタイプがエッジトリガに設定された場合、トリガソースはグローバル変数のみです。グローバル変数がトリガタイプに依存して立ち上がりか立下りもしくは両方のエッジとなる時にトリガがかかります。
出力 <i>i</i> ゲイン	<i>i</i> 番目の入力に対するゲインで <i>i</i> は1から8で設定できます。出力は入力のゲインの値をかけた値になります。
出力 <i>i</i> 開始位置	メッセージは8データポイントまで可能です。データポイントは1bitから32bitsまで可能です。これはメッセージの現在のデータポイントの開始位置を定義します。
出力 <i>i</i> データ終了位置	これはメッセージの現在のデータポイントの終了位置を定義します。
出力 <i>i</i> データタイプ	データタイプは、浮動、整数、またはIQ1からIQ30までとなります。

8.17 割込み時間

割込み時間ブロックは割込みサービスルーチンの時間間隔を測定するのに使われます。

仕様:

パラメータ	機能
Time Output Method	割込み時間測定の設定を行います。 -SCI(使用時間): SCIを使用します。DSPクロックカウンタで割込みサービスルーチンとして使われる時間はSCI出力を経て送られ測定されます。 -SCI(残り時間): SCIを使用します。DSPクロックカウンタの割込みサービスルーチンの残り時間はSCI出力を経て送られ測定されます。残り時間は現在の最後の割込みから次の割り込みの最初までの時間として定義されます。 -GPIO0 to GPIO44又はAIP2 からAIO14: GPIOポートを使います。パルスは特定されたGPIOポートで生成されます。パルスは割込みに入った時high(1)にセットされます。割込みが終了した時にはlow(0)にセットされます。オシロスコープでパルス幅を測定できます。
サンプリング周波数	割込みサービスルーチンのサンプリング周波数 (単位はHz)

SCI が使われる場合、値は DSP クロックでカウントされます。例えば値が 6000 の時 60MHz の DSP クロックの割込み時間は $6000/60M=100\text{ us}$ となります。

8.18 プロジェクト設定とメモリ配置

TI F2803x Hardware Target でコード生成をしようとしたとき、SimCoder は開発環境 TI Code Composer Studio(CCS)のプロジェクトファイルも作成できますので、コンパイル、リンキング、DSP へのアップロードが容易に行えます。

現在、Code Composer Studio のバージョン 3.3 に対応しています。PSIM の回路ファイル名が「test.sch」である場合、コード生成後、「test(C code)」というフォルダが回路ファイルと同じ場所に作成されます。作成されたフォルダには下記のファイルが生成されています。

- test.c	: 生成された C コード
- PS_bios.h	: SimCoder F2803x のヘッダファイル
- passwords.asm	: DSP コードパスワードファイル
- test.pjt	: CCS プロジェクトファイル
- DSP2803x_Headers_nonBIOS.cmd	: ペリフェラルレジスタリンカコマンドファイル
- F28035_FLASH_Lnk.cmd	: フラッシュメモリリンカコマンドファイル
- F28035_FLASH_RAM_Lnk.cmd	: フラッシュ RAM リンカコマンドファイル
- F28035_RAM_Lnk.cmd	: RAM メモリリンカコマンドファイル

注): リンカコマンドファイルの名前は F28035 でない場合はターゲットハードウェアによりアサインされます。例えばもしターゲットハードウェアが F28034 の場合、ファイル名はそれに応じて F28034_FLASH_Lnk.cmd, F28034_FLASH_RAMLnk.cmd, F28034RAM Lnk.cmd となります。

更に下記のファイルが必要となります。

- PsBiosRamF03xFixpt.lib : PSIM フォルダ内の SimCoder F2803x ライブラリ
- PsBiosRomF03xFixpt.lib : PSIM フォルダ内の SimCoder F2803x ライブラリ
- IQmath.lib : PSIM の¥lib フォルダ内の TI IQmath.lib
- 2803x_IQmath_BootROMsymbols.lib : PSIM の¥lib フォルダ内の IQmath のシンボルライブラリ

これらのラブラリファイルはコード生成時に自動的にプロジェクトフォルダにコピーされます。

.c ファイルと .pj1 ファイル(例で言うところの "test.c" と "test.proj") はコード生成実行時に毎回生成及び書き換えが行われます。もしこれら 2 つのファイルを自分で別途変更を加えたい場合は、まずこれらのファイルを別のフォルダにコピーしてから行ってください。次回コード生成実行時に、強制的に上書きが行われ、手動での変更が失われる可能性があります。

プロジェクト設定 :

CCS プロジェクトファイルには 4 つの設定があります。

- RAM Debug : デバッグモードでコンパイル RAM メモリへ書き込みの設定
- RAM Release : リリースモードでコンパイル RAM メモリへ書き込みの設定
- Flash Release : リリースモードでコンパイルフラッシュメモリへ書き込みの設定
- Flash RAM Release : デバッグモードでコンパイル RAM メモリへ書き込みの設定

RAM Debug または RAM Release 設定が選択されたとき、CCS はリンカコマンドファイル F2803x_RAM_Lnk.cmd でプログラムとデータ空間の配置を行います。

Flash Release 設定が選択されたとき、CCS はリンカコマンドファイル F2803x_FLASH_Lnk.cmd でプログラムとデータ空間の配置を行います。

Flash RAM Release 設定が選択されたとき、CCS はリンカコマンドファイル F2803x_FLASH_RAM_Lnk.cmd でプログラムとデータ空間の配置を行います。メモリ配置は RAM Release と同じです。

リリースモードでのコンパイルはデバッグモードに比べると早く済みます。また RAM Release または Flash RAM Release のコードが最も早いです。RAM Debug は遅く、Flash Release が最も遅いですが、開発段階では一般的にデバッグのし易さから RAM Debug から始めて、RAM Release へ、そして Flash Release や Flash RAM Release へ移行していきます。

メモリ配置 :

生成されたリンカファイルではメモリ配置は下記のように定義されます。

RAM Debug、RAM Release、Flash RAM Release の設定の場合 :

RAM Memory
0x0000_0x07FF(2K)
Interrupt vectors
stack
0x8000-0x9FFF(8K*)
Program and data space

Flash Release の設定の場合 :

RAM Memory
0x0000_0x07FF(2K)
Interrupt vectors
stack
0x8000-0x9FFF(8K*)
data space

Flash Memory
0x3E8000_0x3F7FFF(64K**)
Program
Password
etc.

注意 :

- * RAM メモリでは SimCoder はプログラム空間やデータ空間を定義します。
- F28035, F28034, F28032 : 0x8000 から 0x9FFF(8K)
- F28031 : 0x8000 から 0x97FF(6K)
- F28030 : 0x8000 から 0x8FFF(4K)

もしプログラムとデータ空間を合わせたものが RAM 空間を越えた場合は Flash Release 設定を選択する必要があります。

- * * SimCoder によりあらかじめ定義されているプログラム空間に対する flash memory は
- F28035 と F28034 : 0x3E8000 から 0x3F7FFF(64K)
- F28033, F28032, F28031 : 0x3F0000 から 0x3F7FFF(32K)
- F28030 : 0x3F4000 から 0x3F7FFF(16K)

となります。

9 TI F2837x Hardware Target

9.1 概要

オプションモジュールTI F2837x Hardware TargetとSimCoderの組み合わせによって、PSIMの制御回路図からTexas Instruments社製固定小数点型DSP F2837xを搭載した基板で汎用的に使用できるCコードを生成することができます。

TI F2837xハードウェアターゲットはF2837x全てのパッケージに対応しています。

TI F2837x Hardware Targetには下記の機能があります。

- PWM生成機能：3-phase、2-phase、1-phaseとAPWM生成機能
- 可変周波数PWM
- PWM生成のStart、Stop機能
- トリップゾーン、トリップゾーンステート機能
- A/D変換機能
- ADC電圧基準
- コンパレータ入力、出力とDAC
- DACコンバータ機能
- デジタル入力、出力、サンプルタイム機能
- 入力X-BAR、出力X-BAR、PWM X-BAR機能
- エンコーダ、エンコーダステート、エンコーダインデックス／ストローブ位置機能
- Up/Downカウンタ機能
- キャプチャ、キャプチャステート機能
- SCI基板設定機能、入力出力
- SPI基板設定機能、デバイス、入力出力
- CAN設定機能、入力出力、リモートリクエスト
- 割り込み時間機能
- DSP クロック機能
- ハードウェア基板設定機能

複数のサンプリング周波数でコードを生成する回路の場合、SimCoder は優先的に PWM 割り込みの周波数に使います。他のサンプリング周波数では、タイマ 1 割り込み、タイマ 2 割り込みの順に使われます。3 つ以上のサンプリング周期がある場合は対応する割り込み関数をメイン関数内で実行する形にします。

TI F2837x では PWM 生成部からハードウェア割り込みを発生させることができますので、SimCoder は PWM 生成ブロックに繋がっていて、PWM 生成ブロックと同じ周波数を持つ全ての素子を検索してグループ化します。各素子ブロックは自動的に配置され、出力コード内の割り込み関数として実行されます。

また、ハードウェア割り込みはデジタル入力、エンコーダ、キャプチャ、そしてトリップゾーンにもあります。各ハードウェア割り込みは割り込みブロックから生成され、(5.4 章参照) 割り込みブロックは割り込み関数と繋げる必要があります。(割り込み関数はサブ回路で作ります。) 二つ以上の割り込みを使うときはそれぞれの機能に割り込みブロックと割り込み関数を設定します。

本章では TI F2837x Hardware Target ライブラリ内の各素子の使い方について解説します。次の図は F2837x 100-pin PZP ポートアサイン解説します。GPIO 端子には GPIO の機能のみが表示されます。その他のポートの機能については、次のハードウェア構成の章で説明します。

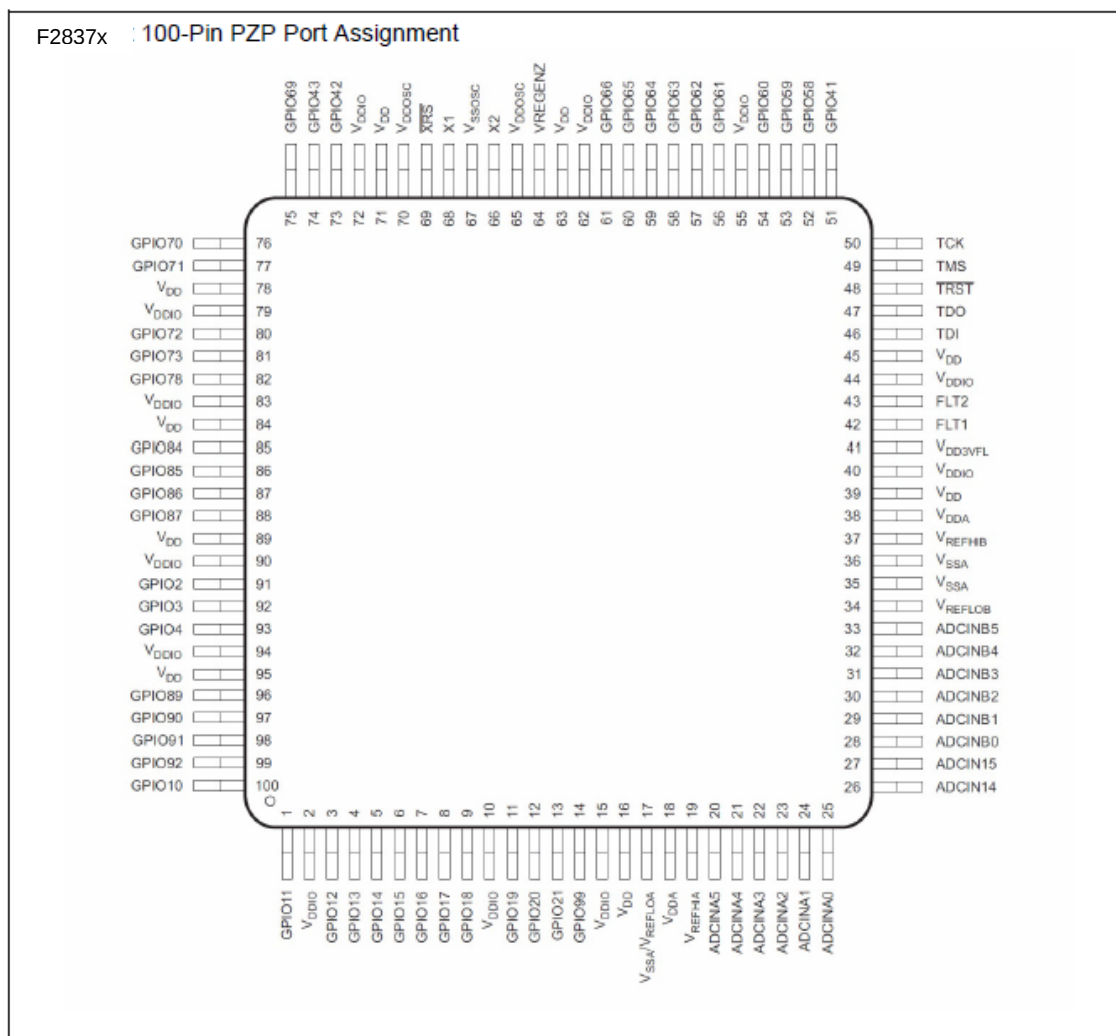


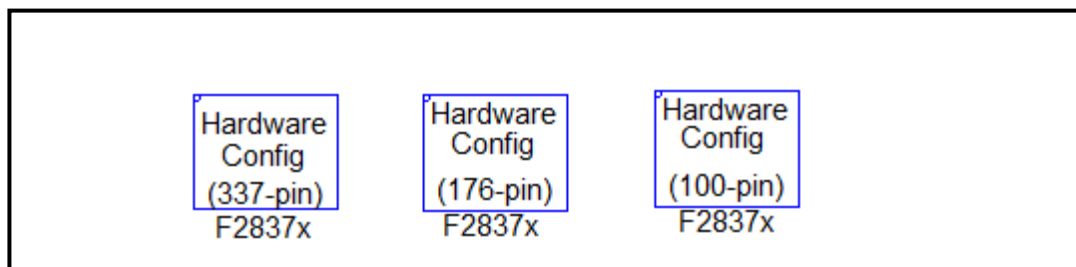
図 9-3 F2837x 100-pin PZP ポートアサイン図

9.2 ハードウェア構成

F2837x にはアナログ入力と独立したプログラムマルチプレクサの GPIO ポートがあります。GPIO ポートの大部分はいくつかの機能の 1 つに割り当てられます。例えばポート GPIO0 はデジタル入力、デジタル出力、または PWM 出力として使用できます。A/D チャンネルのいくつかはコンパレータ入力

として定義できます。それゆえ PSIM 回路図の中で正しく GPIO ポートの機能を定義しないといけません。

シンボル:



ブロックのダイアログウィンドウは次のようになります。

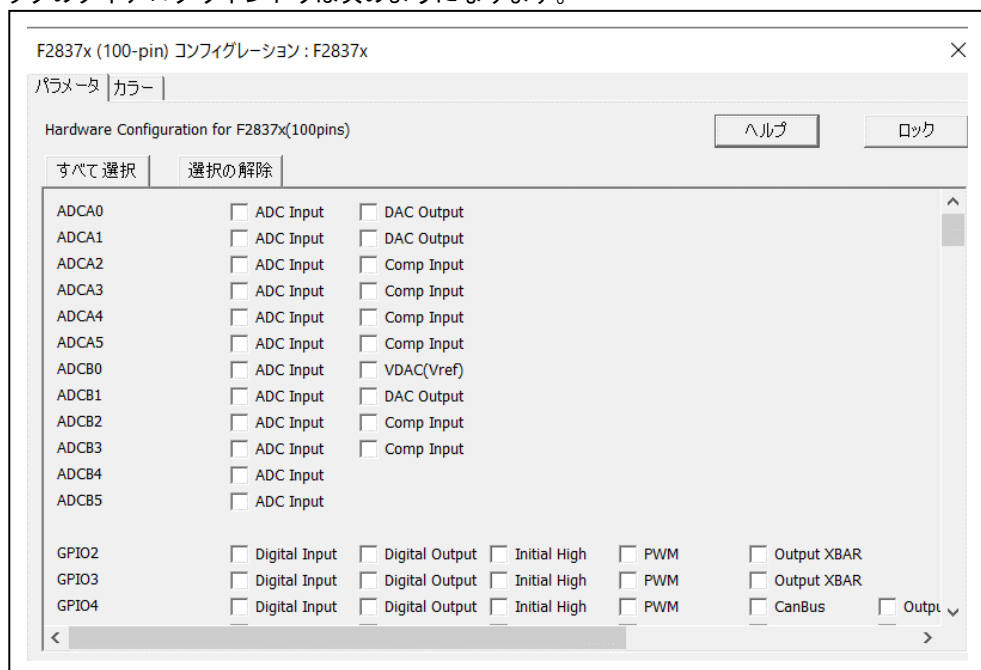


図 9- 4 ブロックのダイアログウィンドウ

ハードウェア構成ブロックは F2837x ハードウェアの I/O ポートを定義します。ポートは正しくアサインしてください。使用しないポートはチェックを入れないようにしてください。

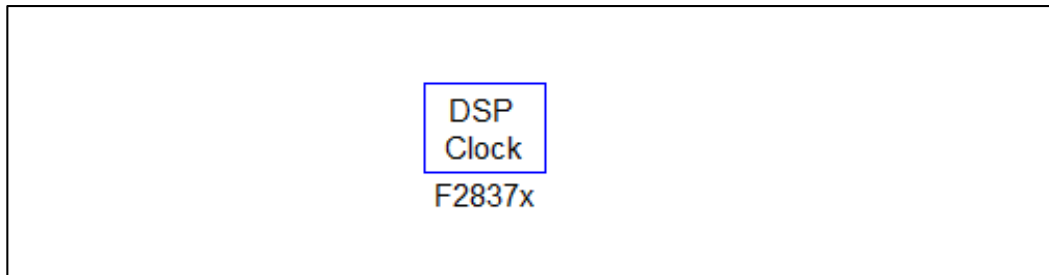
右上の「ロック」／「アンロック」をクリックして設定のロックと変更ができます。変更する場合は「アンロック」ボタンをクリックします。変更後は「ロック」をクリックします。各々の GPIO ポートのチェックボックスにある機能が使用可能です。チェックを入れるとその機能にあわせて GPIO ポートが設定されます。各 GPIO ポートは 1 つのボックスのみチェックを入れることができます。 SimCoder ではこのボックスがチェックされている機能だけが使用できます。例えばポート GPIO1 で “Digital Input” がチェックされるとこのポートは “Digital Input”

となります。他の機能としては使用できません。PSIM 回路中で PWM 出力として使用されるとエラーメッセージが出ます。

9.3 DSP クロック

DSP クロックブロックは外部クロック周波数と F2837x DSP の周波数の設定ができます。

シンボル:



仕様:

パラメータ	機能
DSP クロック ソース	F2837xは次の3つの方法でシステムクロックを提供します。 ・ Internal Oscillator1(内部発信器1) ・ Internal Oscillator2(内部発信器2) ・ External Oscillator(外部発信器)
外部クロック (MHz)	DSPボード上の外部クロックの周波数(単位はMHz) 周波数は整数値である必要があり最大値は20MHzです。このパラメータはDSPクロックソースが内部オシレータ1か2と選択された場合は無視されます。
DSP速度 (MHz)	DSP速度 (単位はMHz)。 入力するDSP速度は整数値でなければなりません。外部クロック周波数の整数倍でなければなりません。内部発信器使用時は194MHz、外部発信器使用時は200MHzが最大値です。

DSP クロックブロックが回路中で使用されていない場合は、DSP コンフィギュレーションブロックはデフォルト値が使われその設定は以下のようになります。

- ・ 外部発信器
- ・ 20MHz
- ・ 200MHz

9.4 PWM 生成器

F2837x には、PWM モジュールが 100pin パッケージ用に 7 セット、176pin パッケージ用に 12 セット、330pin パッケージ用に 12 セット含まれています。各 PWM モジュールには 2 つの出力ポートがあります。

SimCoder では PWM は次のように使われます。

- ・ 1-phase PWM 生成器 : 2 つの出力があり、互いに相補的になるようになっています。
- ・ 外部位相シフト付き 1-phase PWM 生成器 : 2 つの出力があり、互いに相補的になるようになっています。

- ・ 2-phase PWM 生成器：相補的ではない特別なモードでの 2 つの出力があり
- ・ 3-phase PWM 生成器：PWM123（PWM1、2、3 で構成）、PWM456（PWM4、5、6 で構成）のように、連続した順序の 3 つの単相 PWM 発生器で構成されています。

これらの PWM 生成器は A/D コンバータにトリガをかけたり、トリップゾーン信号を使用することができます。上記の PWM 生成器のほかに、キャプチャと同じリソースを使用する 6 つの単出力 APWM 生成器があります。

これらの APWM 生成器は、PWM 生成器に比べて機能が制限されています。A/D コンバータのトリガーや、トリップゾーン信号を使用することはできません。また、リソースを共有しているため、特定のポートをキャプチャに使用すると、そのポートを PWM 生成器に使用することができません。

SimCoder のすべての PWM 生成器は、内部に 1 つのスイッチング期間の遅延を含んでいることに注意してください。つまり、PWM 生成器の入力値は、PWM 出力の更新に使用される前に、1 周期分だけ遅延されます。この遅延は、DSP ハードウェアの実装に固有の遅延をシミュレートするために必要となります。

PWM 生成器には、"PWM Freq. Scaling Factor"というパラメータがあります。

1~15 に設定することができます。スケーリングファクタが 3 より大きい場合、PWM は未使用の PWM を使用してサンプリング周波数で割り込みを発生させます。

この未使用の PWM は、定期的な割り込みを発生させるためだけに使用され、その出力はまだ他の機能に使用することができます。

システム内に未使用の PWM が存在しない場合は、タイマーが使用されます。

PWM 生成器は 2 つの方法で割り込みを発生させることができます。

- ・ 定期的な割り込み。割り込み周波数は、サンプリング周波数と同じです。

以下の方法で発生させることができます。

- PWM が割り込みソースとして選択されている場合、PWM の搬送波の開始時に PWM 自身によって割り込みが発生します。
- A/D コンバータが割り込みソースとして選択されている場合は、PWM が A/D コンバータをトリガして変換を開始します。A/D 変換が完了すると、割り込みが発生します。
- ・ トリップゾーンの割り込み。入力 X バーからの 3 つのトリップゾーンシングル（TZ1, TZ2, TZ3）と 4 つの DC トリップイベント（またはその組み合わせ）を割り込みソースとして使用することができます。PWM のトリップゾーン信号がアクティブになると、トリップゾーン割り込みが発生します。トリップゾーン割り込みに入る前に、すべての PWM 出力はトリップアクションの定義に従って設定されます。DC トリップイベントは、対応する PWM トリップゾーンエレメントに設定されます。

9.4.1 3-phase PWM、1-phasePWM,1-phasePWM(phase shift) :

3 相 PWM 発生器は、連続した順序の 3 つの単相 PWM ブロックで構成されてい

ます。そのため、3 相 PWM ブロック、1 相 PWM ブロック、1 相 PWM（位相シフト）ブロックの属性はほとんど同じです。

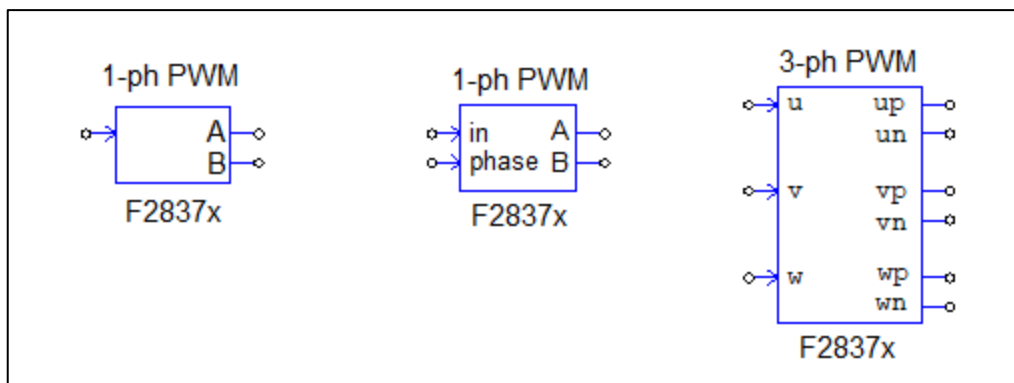
1-Phase PWM ブロックと 1-Phase-PWM（フェーズシフト）ブロックの違いは、フェーズシフトの定義方法です。1-Phase PWM ブロックは、定数であるパラメータを介して位相シフトを定義するのに対し、1-Phase PWM(位相シフト)ブロックは、外部入力（画像では "phase"と表示されている）から位相シフトを読み取ります。位相シフトの単位は degree(°)です。

3 相 PWM 発電機のイメージでは、「u」「v」「w」が 3 つの相を意味します（別の言い方をすれば、相「a」「b」「c」となります）。

p "は正の出力を、"n "は負の出力を意味します。

例えば、3 相 PWM 123 の場合、"up "は PWM1A、"un "は PWM1B です。

シンボル:



仕様:

パラメータ	機能
PWM ソース	PWM生成器のソースです。各1相PWMブロックは、2つのGPIOピンに割り当てられた2つの出力（PWM AおよびPWM B）を有しています。各3相PWMモジュールは、3つの1相PWMブロックで構成されます。例えば、3相PWM 123 (GPIO0)は、PWM1、PWM2、PWM3で構成されています。したがって、各3相PWMには6本のGPIO端子があります。
出力 Mode (3-phaseでは使用しません。)	PWM成生器の出力モード。以下のいずれかとなります。 - PWM A&Bを使用：PWM出力AとBの両方が使用され、それらは相補的です。 - PWM Aの使用：PWM出力Aのみを使用します。 - PWM Bを使用：PWM出力Bのみを使用します。 - No PWM output: PWM出力なし：PWM出力を使用しません（タイマーとして使用します）。
デッドバンド タイプ	PWMのデッドバンドの種類。 - アクティブ・ハイ・コンプリメンタリー - アクティブ・ロー・コンプリメンタリー - アクティブ・ハイ - アクティブ・ロー
立上りエッジのデッドタイム(us)	PWM生成器の立ち上がりエッジのデッドタイム（単位：秒）。
立下がりエッジのデッドタイム(us)	PWM生成器の立ち下がりエッジのデッドタイム（単位：秒）。

サンプリング周波数	PWM生成器のサンプリング周波数（単位:Hz）。計算され、この周波数に基づいてPWM信号のデューティサイクルが更新されます。
PWM周波数倍率	PWMスイッチング周波数とサンプリング周波数間の比で、1~15の値となります。例えばサンプリング周波数が50kHzでスケーリングファクタ3の場合はPWMのスイッチング周波数は150kHzとなります。つまり、スイッチは150kHzで動作します。しかし、ゲーティング信号は50kHz、つまり3回のスイッチングサイクルに1回の割合で更新されます
キャリア波型	キャリア波形タイプの設定。 - Triangular(start low) : キャリアを三角波、PWM出力の初期値はlow。 - Triangular(start high) : キャリアを三角波、PWM出力の初期値はhigh。 - Sawtooth(start low) : キャリアをのこぎり波、PWM出力の初期値はlow。 - Sawtooth(start high) : キャリアをのこぎり波、PWM出力の初期値はhigh。
トリガADC	A/DコンバータをトリガするPWMジェネレータの有無を設定します。以下のいずれかになります。 - Do not trigger ADC : PWMはADCをトリガしません。 - Trigger ADC : PWMはADCをトリガします。
ADCトリガ位置	A/Dのトリガ位置は0から1未満の範囲で、0の場合はPWM周期の最初にA/Dコンバータをトリガし、0.5の場合はPWM周期の180°の位置でA/Dコンバータをトリガします。
トリップゾーン 1, 2, 3使用	PWMジェネレーターがトリップゾーン信号を使用するかどうかを定義します。以下のいずれかを指定します。 - Do not use : i 番目のトリップゾーン信号を無効にします。 - One shot : PWM生成器は、ワンショットモードではトリップゾーン信号を使用します。一度トリガーがかかると、PWMは手動で開始しなければなりません - Cycle by cycle : PWM生成器は、トリップゾーン信号をサイクルごとに使用します。トリップゾーン信号は、現在のサイクル内で有効であり、PWMは次のサイクルで自動的に再起動します。
トリップイベント 選択	電流PWMに適用されるDCトリップイベントの組み合わせを指定します DCAEVT1, DCAEVT2, DCBEVT1, DCBEVT2の1つまたは任意の組み合わせとなります。
トリップ アクション	トリップ動作に対するPWM生成器の応答方法を定義します。次のいずれかとなります。 - 「High Impedance」 : PWMの両出力端子はハイインピーダンスに設定されます。 - 「PWM A high & B low」 : PWMのAはハイレベル、Bはローレベルに設定されます。 - 「PWM A low & B high」 : PWMのAはローレベル、Bはハイレベルに設定されます。 - 「PWM A/B high」 : PWMのA,Bともにハイレベルに設定されます。 - 「PWM A/B low」 : PWMのA,Bともにローレベルに設定されます。 - 「No action」 : 何も動作を設定しません。
ピーク間電圧	キャリア波のピーク間電圧Vppの設定。

オフセット電圧	キャリア波のオフセット電圧Voffsetの設定。
位相シフト (1相のみ)	基準となるPWM生成器出力に対する出力の位相シフト量（単位：deg）。 例えば、位相シフトが-30の場合、出力は基準PWM出力に対して30deg.だけ右にシフトする（遅れている）ことになります。
初期入力値	u、v、w三相入力ノードの初期値の設定。三相PWMブロックの3つの入力u,v,wは、3つの初期値が設定されています。
HRPWM使用（3相では使用しません）	高分解能PWMを定義します。次のいずれかとなります。 - Do not use HRPWM: 高分解能PWMを使用しません。 - Use HRPWM without calibration: キャリブレーションを行わずに高分解能PWMを使用します。 - Use HRPWM with calibration（校正付きHRPWMの使用）：校正に高分解能PWMを使用します。
最初からPWM 信号出力	- Startに設定されている場合、PWMは最初から開始されます。 - Do not startに設定されている場合は、Start PWMブロックを使ってPWMを開始する必要があります。
シミュレーション出力モード	シミュレーションの出力モードは、次のいずれかに設定できます。 - switching mode：PWMブロックの出力はPWM信号となります。 - average mode：PWMブロックの出力は平均モードの信号となります。 average modeでは、搬送波が負から正に向かっており、負のピークと正のピークの絶対値が等しい場合（例えば、搬送波が-1から+1、または-5から+5の場合）、その変調はAC信号の変調とみなされます。 それ以外の場合、変調は直流信号の変調とみなされます。例えば、三相インバータや単相インバータの変調は交流変調であり、降圧コンバータの変調は直流変調となります。 交流信号変調では、PWMブロックの入力uをVuとすると、平均モードの出力upとunは $V_{up} = V_u / (V_{pp} + V_{offset})$ $V_{un} = -V_{up}$ この場合、Vuは-(Vpp + Voffset)と(Vpp + Voffset)の間、Vupは-1と+1の間となります。 直流信号の変調では、アベレージモードでの出力upとunが $V_{up} = (V_u - V_{offset}) / V_{pp}$ $V_{un} = 1 - V_{up}$ となります。 この場合、VuはVoffsetと(Vpp + Voffset)の間、Vupは0と+1の間となります。 例えばPWMブロックの入力をVuとすると出力のupとunは $V_{up} = V_u / (V_{pk_pk} + V_{offset})$ $V_{un} = -V_u / (V_{pk_pk} + V_{offset})$ となります。 Average modeに設定するとPWMブロックの出力をaverage mode modelのコンバータ/インバータへ接続できます。

Phase Shift

1 相 PWM 発生器は、他の PWM 信号に対して位相をシフトさせた PWM 信号を生成することができます。位相シフトのための PWM ブロックの定義方法については、8.4.5 項で説明します。

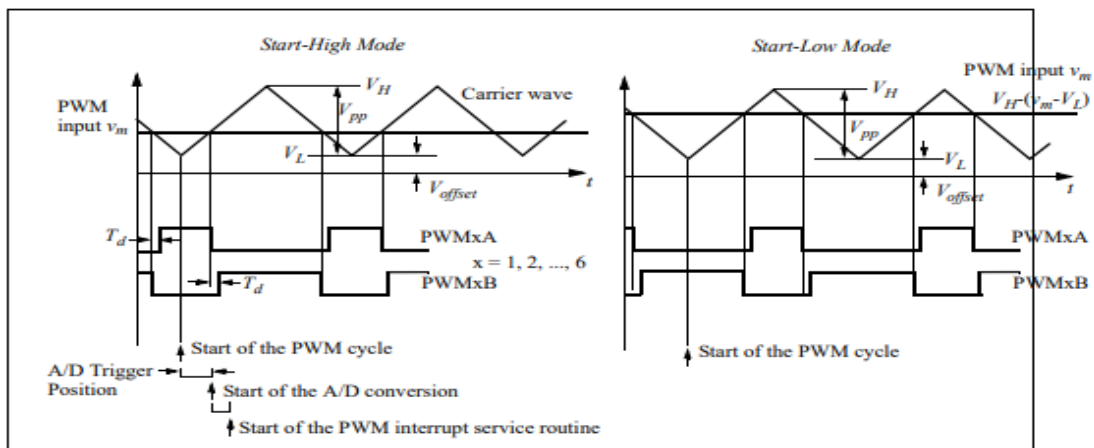
フェーズシフトの値は“°”で表します。値が-30°の場合、出力は基準となる PWM ジェネレータの出力に対し、スイッチング周期の 30° だけ右にシフト（遅れ）します。

これは、PWM 搬送波を 30° 右にシフトさせることに相当します。位相値が 30° の場合、出力は左に

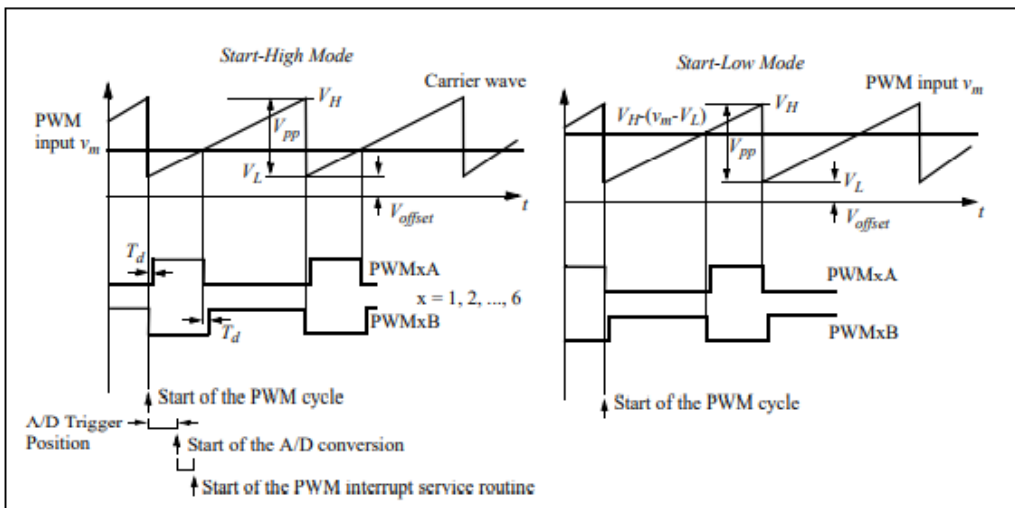
30° シフトします。(先行) 30°

Carrier Wave

キャリア波形には、三角波（立ち上がりと立ち下りの間隔が等しい）とノコギリ波の 2 種類があります。また、動作モードには、以下のようにスタート-low モードとスタート-high モードの 2 種類があります。三角波搬送波を用いた PWM 発生器の入力波形と出力波形は次のようになります。



のこぎり波搬送波を用いた PWM 発生器の入力波形と出力波形は次のようになります。



前図は、デッドタイムの定義と、PWM 発生器が A/D コンバータをトリガするときのタイムシーケンスを示しています。A/D コンバータのトリガが選択された場合、PWM サイクルの開始から、A/D トリガの位置で定義される一定の遅延の後、A/D 変換が開始されます。A/D 変換が完了した後、PWM 割り込みサービスルーチンが開始されます。

PWM 発生器が A/D コンバータをトリガしない場合、PWM 割り込みサービスルーチンは PWM サイク

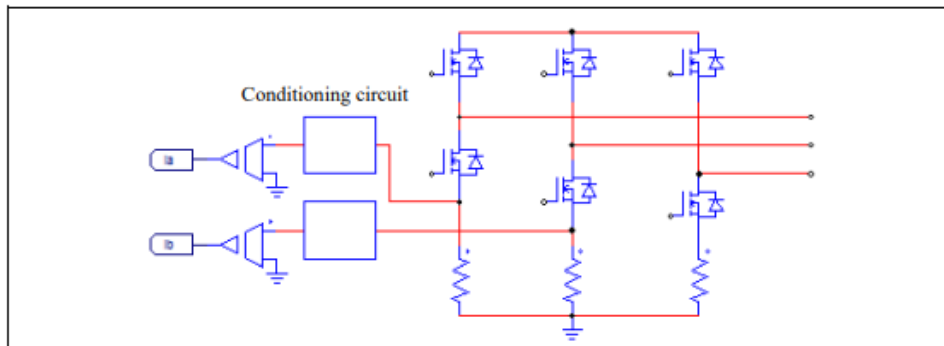
ルの開始時に開始されます。

前図は、スタート-high およびスタート-low モードの動作も示しています。PWM 入力を v_m とし、搬送波の最小値を V_L 、最大値を V_H とします。スタート-high モードでは、PWM 正出力 PWMA は、スイッチングサイクルの開始時に high レベルとなり、入力 v_m が搬送波より大きい間、high レベルを維持します。
例えば、0 から 1 の搬送波の場合、 $V_L=0$ 、 $V_H=1$ となります。 $v_m=0.2$ の場合は PWM 出力 PWMA は、搬送波が 0.2 以下である限り、high を維持します。

一方スタート-low モードでは、PWM の正出力 PWMA はスイッチングサイクル開始時に low となり、キャリア波が $V_H-(v_m-V_L)$ より大きい場合は high となります。例えば、搬送波が 0 から 1 の場合、 $V_L=0$ 、 $V_H=1$ となります。 $v_m=0.2$ の場合、搬送波が 0.8 以上であれば、PWM 出力 PWMA は High となります。

キャリアスタートモードは、スイッチ電流の測定方法によって異なります。例えば、3 相インバータの場合、スイッチ電流がトップ スイッチ電流を測定する場合は、スタート-high モードを選択する必要があります。これにより、サイクルの開始時に、トップスイッチのゲーティング信号が高く、電流が導通していることを確認します。

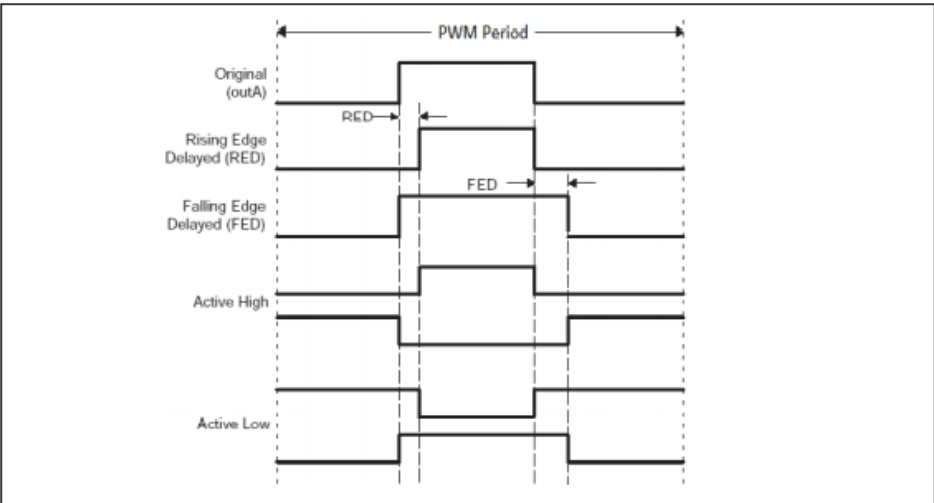
逆に、ボトムスイッチ 電流が測定された場合は、スタート-low モードを選択すべきである。これにより、サイクルの開始時には トップスイッチのゲーティング信号は低く、ボトムスイッチのゲーティング信号は高く、電流は流れています。例えば、次の回路では、A 相と B 相のボトムスイッチ電流を測定しています。この場合、キャリア・スタート・low・モードを選択する必要があります。



注)：スタート-low モードでは、PWM 信号を生成するために搬送波と比較される前に、PWM 入力 v_m

内部で $V_H-(v_m-V_L)$ に変換されます。この変換により、スタート-low モードとスタート-high モードは同じデューティサイクル表現になります。例えば、 $V_L=0$ 、 $V_H=1$ の鋸歯状波や、 $V_L=-V_H$ の三角波の場合、スタート-low、スタート-high の両モードにおける PWMA 出力のデューティサイクル D は、次のようになります。 $D = v_m/V_H$ となります。

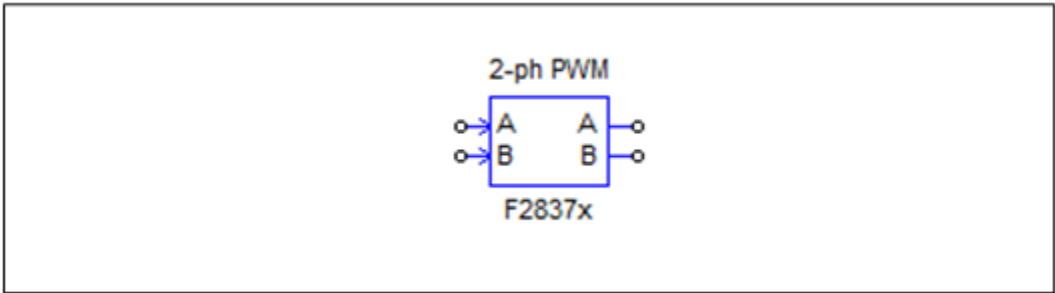
PWM のデッドバンドは次のようになります。



9.4.2 2-phase PWM 生成器：

2 相 PWM 生成器は、2 つの入力と 2 つの出力を持ち、あらかじめ設定された 6 つの動作モードのうちの 1 つで動作します。

シンボル:



仕様:

パラメータ	機能
PWM ソース	PWM生成器の出力ピン設定。 PWM1~8までを選択できます。
サンプリング 周波数	PWM生成器のサンプリング周波数の設定。単位は[Hz] PWM信号のデューティサイクルはここで設定した周波数でアップデートされます。
PWM 周波数倍率	PWM周波数とサンプリング周波数の倍率の設定。 1~100倍に設定ができます。例えばサンプリング周波数が50kHzでスケーリグ倍率が3の場合、PWMスイッチング周波数は150kHzとなります。つまりスイッチは150kHzで動作します。がゲーティング信号は50kHzで、または3スイッチングサイクル毎に1回更新されます。

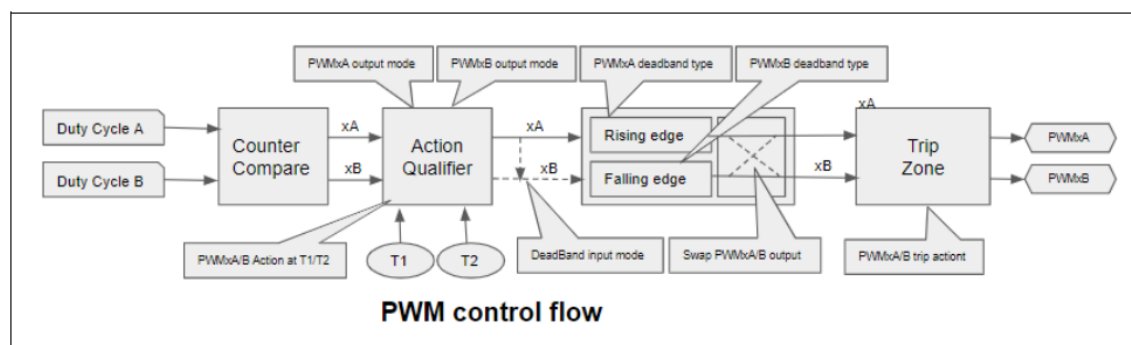
キャリア波型	搬送波の種類と、PWM出力の初期状態。次のいずれかとなります。 - Sawtooth のこぎり波 - triangular 三角波
PWMxA出力モード PWMxB出力モード	PWMxAおよびPWMxBの出力モードは、次のいずれかとなります。 - <i>Do nothing</i> : PWM出力はPWMデューティサイクルの値に影響されません。 - <i>Toggle at valley</i> : PWM期間の初めにPWM出力がトグルします。 - <i>Toggle at peak</i> : PWM搬送波のピークでPWM出力がトグルします。 - <i>Toggle at peak/valley</i> : PWM搬送波のピーク／バレーの両方でPWM出力がトグルします。 - <i>Set/Reset at valley/peak</i> : 搬送波の谷でhigh、ピーク部でlowになるPWM出力です。 - <i>Reset/Set at valley/peak</i> : 搬送波の谷でlowにリセットされ、ピークでhighになるPWM出力です。 - <i>Set at valley</i> : PWM周期の開始時にPWM出力がhighになります。 - <i>Reset at valley</i> : PWM周期の開始時にPWM出力がlowにリセットされます。 - <i>Set at peak</i> : 搬送波のピーク時にPWM出力をhighにします。 - <i>Reset at peak</i> : 搬送波のピーク時にPWM出力をlowにリセットします。 次はPWMxAのみの設定です。 - <i>Set high at CMPA</i> : PWM出力は、PWM期間の初めにlowで始まり、CMPAが設定されるとhighになります。 - <i>Set low at CMPA</i> : PWM出力は、PWM期間の初めにhighになり、CMPAが設定されるとlowになります。 - <i>Set high/low at CMPA/B</i> : PWM出力は、PWM期間の初めにlowから始まり、CMPAが設定されるとhighになり、CMPBが設定されるとlowになります。 - <i>Set low/ high at CMPA/B</i> : PWM出力は、PWM期間の初めにhighから始まり、CMPAが設定されるとlowになり、CMPBが設定されるとhighになります。 次はPWMxBのみの設定です。 - <i>Complementary to PWMxA</i> : PWM出力はPWMxAと相補的な関係にあります。 - <i>Set high at CMPB</i> : PWM出力は、PWM期間の初めにlowで始まり、CMPAが設定されるとhighになります。 - <i>Set low at CMPB</i> : PWM出力は、PWM期間の初めにhighで始まり、CMPAが設定されるとlowになります。
AD変換トリガ	A/D変換へのトリガの設定。次のいずれかになります。 - 「Do not trigger ADC」 : PWMはA/D変換をトリガしません。 - 「Trigger ADC」 : PWMはA/D変換をトリガします。
A/D変換トリガ 位置	A/Dのトリガ位置は0から1未満の範囲です。0の場合はPWM周期の最初にA/Dコンバータをトリガし、0.5の場合はPWM周期の180° の位置でA/Dコンバータをトリガします。
T1ソース T2ソース	PWMトリップアクションのT1およびT2イベントソースは、次のいずれかとなります。 - <i>Do not us,</i> - <i>DCAEVT1,</i> - <i>DCAEVT2,</i> - <i>DCBEVT1,</i> - <i>DCBEVT2,</i> - <i>TZ1,</i> - <i>TZ2,</i> - <i>TZ3,</i>

PWMxA Action at T1(Up) PWMxA Action at T1(Down) PWMxB Action at T1(Up) PWMxB Action at T1(Down) PWMxA Action at T2(Up) PWMxA Action at T2(Down) PWMxB Action at T2(Up) PWMxB Action at T2(Down)	PWMxA (PWMxB) 出力は、PWM波形のアップ (ダウン) カウント期間にT1 (T2) イベントが発生した場合、次の動作で強制的に出力することができます。 - <i>Do nothing</i> , - <i>Clear</i> , - <i>Set</i> , - <i>Toggle</i> ,
デッドバンド入力モード	デッドバンドの入力モードを指定します。 - 両方の入力を使用する。PWMデッドバンドサブモジュールの入力をPWMxAとPWMxBの両方から行う - PWMxAの入力のみを使用。PWMデッドバンドサブモジュールの入力がPWMxAのみの場合
PWMxA Dead Band Type PWMxB Dead Band Type	PWMデッドバンドのタイプは、次のいずれかになります。 - No Delay: 遅延をかけない - Rising/Falling edge delay: PWMxA/Bに立上り/立下りエッジディレイを適用 - 立ち上がり/立ち下がりエッジディレイ (反転)。立ち上がり/立ち下がりエッジディレイをPWMxA/Bに適用した後、反転させます。
PWMxA/B出力交換	PWM出力AとBを入れ替える場合の指定
立ち上がりエッジのデッドタイム	PWM生成器の立上り時間 単位us
立ち下がりエッジのデッドタイム	PWM生成器の立下り時間 単位us
トリップゾーン1, 2, 3の使用	PWM生成器がトリップゾーン信号を使用するかどうかを定義します。以下のいずれかを指定します。 - Do not use: i番目のトリップゾーン信号を無効にします。 - One Shot: PWMジェネレーターは、ワンショットモードでトリップゾーン信号を使用します。一度トリガーされると、PWMは手動で開始する必要があります。 - Cycle by cycle: PWMジェネレーターは、トリップゾーン信号をサイクルバイサイクルで使用します。トリップゾーン信号は現在のサイクル内で有効であり、PWMは次のサイクルで自動的に再起動します。
トリップイベント選択	電流PWMに適用されるDCトリップイベントの組み合わせを指定します。DCAEVT1, DCAEVT2, DCBEVT1, DCBEVT2の1つまたは任意の組み合わせとなります。
CBCロック解除タイミング	サイクルバイサイクルモード時のロッククリアのタイミングを指定します。 - PWMバレー時 - PWMバレー/ピーク時
PWMxA Trip Action(Up) PWMxA Trip Action(Down) PWMxB Trip Action(Up) PWMxB Trip Action(Down)	トリップ動作に対するPWMジェネレーターの応答方法を定義します。以下のいずれかとなります。 - <i>High impedance</i> : PWM出力をハイインピーダンスにします。 - <i>Force high</i> : PWM AをHigh, BをLowに設定します。 - <i>Force low</i> : PWM AをLow, BをHighに設定します。 - <i>Toggle</i> : 出力をトグルします。 - <i>No action</i> : アクションしません。

ピーク値	搬送波のピーク値Vpk。下限は0で、-Vpkではありません。
位相シフト	基準となるPWM発電機の出力に対する出力の位相シフト量（単位：°）。 例えば、位相シフトが-30の場合、出力は基準PWM出力に対して30° だけ右にシフトする（遅れている）ことになります。
初期値A 初期値B	入力A,Bの初期値
HRPWM使用	高分解能PWMを定義します。次のいずれかとなります。 - Do not use HRPWM: 高分解能PWMを使用しない - Use HRPWM without calibration: キャリブレーションを行わずに高分解能PWMを使用します。 - HRPWM with calibration: キャリブレーション付きの高分解能のPWMを使用します。
最初からPWM信号出力	"Start "に設定されている場合：PWMは最初から始まります。 "Do not start "に設定されている場合："Start PWM "ファンクションブロックを使用してPWMを開始する必要があります。

2 相 PWM 生成器は、パワーコンバータの典型的なアプリケーション用に特定の構成の 2 つの PWM 信号を生成します。PWM の搬送波はノコギリ波または三角波で、その範囲は 0 からピーク値 Vpk までです。

PWM モジュールの制御フローは以下の通りです。



T1/T2 イベントは、F2837x Trip Zone ブロックで定義されます。

SimCoder では、次の PwmxA/B 出力モードの組み合わせが可能です。

PWMxA output mode	PWMxB output mode	Sawtooth	Triangular
Toggle	Toggle		
Toggle	Set high at duty cycle 2	PwmxB set high at duty cycle 2	PwmxB set high/low at count up/ down period at duty cycle 2.
Toggle	Set low at duty cycle 2	PwmxB sets high at duty cycle 2	PwmxB set low/high at count up/ down period at duty cycle 2.
Set high at duty cycle 1	Toggle	PwmxA sets high at duty cycle 1	PwmxA sets high/low at count up/ down period at duty cycle 1
Set high at duty cycle 1	Set high at duty cycle 2	PwmxA/B set high at duty cycle 1/2	PwmxA set high/low at count up/ down period at duty cycle 1; PwmxB set high/low at count up/ down period at duty cycle 2.
Set high at duty cycle 1	Set low at duty cycle 2	PwmxA/B set high/ low at duty cycle 1/2	PwmxA set high/low at count up/ down period at duty cycle 1; PwmxB set low/high at count up/ down period at duty cycle 2.
Set low at duty cycle 1	Set high at duty cycle 2	PwmxA/B sets low/ high at duty cycle 1/2	PwmxA sets low/high at count up/ down period at duty cycle 1; PwmxB set high/low at count up/ down period at duty cycle 2.
Set low at duty cycle 1	Set low at duty cycle 2	PwmxA/B sets low at duty cycle 1/2	PwmxA sets low/high at count up/ down period at duty cycle 1; PwmxB set low/high at count up/ down period at duty cycle 2.
Set high/low at duty cycle 1/2	Toggle	PwmxA sets high/low at duty cycle 1/2	PwmxA sets high/low at count up/ down period at duty cycle 1/2.
Set low/high at duty cycle 1/2	Toggle	PwmxA sets low/high at duty cycle 1/2	PwmxA sets high/low at count up/ down period at duty cycle 1/2.

表の“Toggle”は、以下のモードのいずれかです。何もしない、谷でトグル、ピークでトグル、谷/ピークでトグル、谷/ピークでセット/リセット、谷/ピークでリセット/セット、谷でセット、谷でリセット、ピークでセット、ピークでリセット。

デッドバンドは、PWMxB の出力モードが PWMxA と相補的な場合にのみ許可されます。

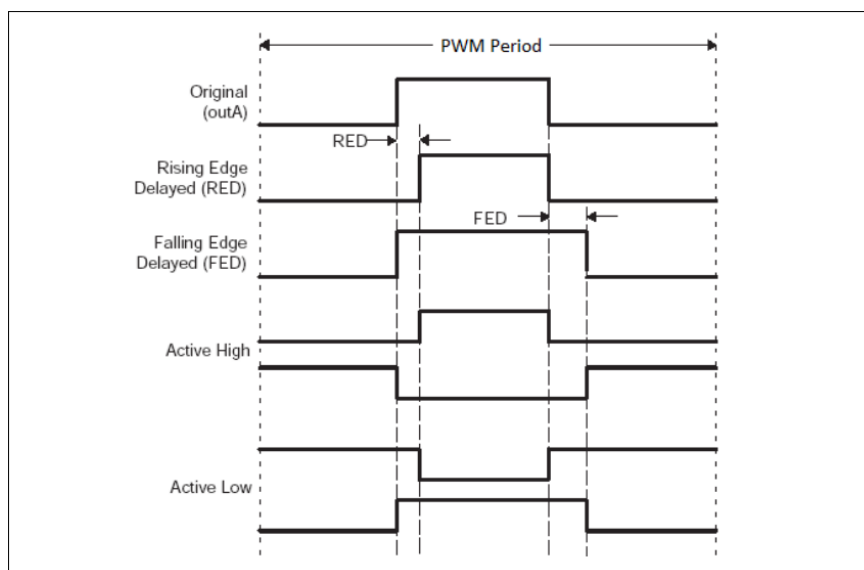
通常、立ち上がりエッジの遅延は PWMxA に、立ち下がりエッジの遅延は PWMxB に設定されますが、立ち上がりエッジの遅延を PWMxB に設定する場合、または立ち下がりエッジを PWMxA に設定する場合は、パラメータ「Swap PWMxA/B Output」を「Yes」に設定してください。

デッドバンドは、次のパラメータで指定された PWM 動作にのみ適用されます。

PWMxA/B 出力モード、T1/2 (Up/Down) における PWMxA/B 動作。

デッドバンドは、PWM トリップ動作には適用されません。

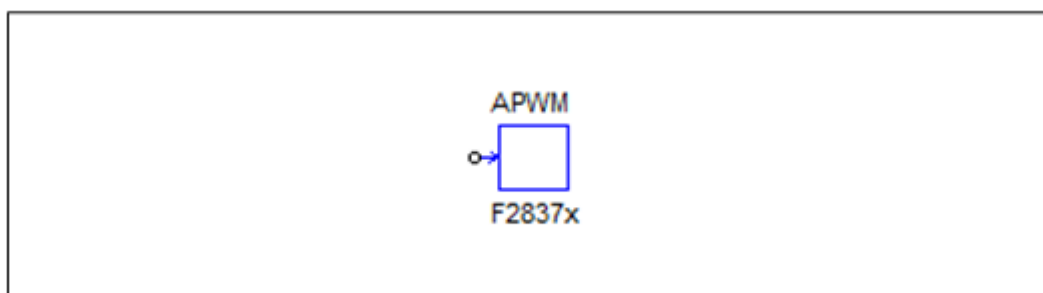
デッドバンドの設定は次のグラフのようになります。



9.4.3 Single PWM(キャプチャと共用) :

単相 PWM (キャプチャと共用) 生成器は、APWM と呼ばれ、キャプチャと同じリソースを共有します。他の PWM ジェネレーターに比べて機能が制限されています。A/D コンバータをトリガすることができず、トリップゾーン信号も使用できません。また、リソースを共有しているため、共有の GPIO ポートをキャプチャに使用した場合、PWM 生成器には使用できません。

シンボル:



仕様:

パラメータ	機能
PWM ソース	PWM生成器のソースです。APWM1～APWM6のいずれかとなります。
PWM 周波数	PWM生成器の周波数です。単位は[Hz]
キャリア波形タイプ	キャリア波形種類とPWM出力の初期状態を指定します。 次のいずれかとなります。: - のこぎり波 (lowでスタート): キャリアをのこぎり波、PWM出力の初期値はlow。 - のこぎり波 (highでスタート): キャリアをのこぎり波、PWM出力の初期値はhigh。

ストップアクション	PWM生成器が停止している時の出力設定。 次のようになります。 - 出力low: PWM出力はlowに設定されます。 - 出力high: PWM出力はhighに設定されます。
ピーク間電圧	キャリア波のピーク間電圧の設定。
オフセット値	キャリア波のDCオフセット電圧の設定。
位相シフト	PWM生成器の出力に対しての位相シフト値（単位 °）
初期入力値	入力の初期値の設定。
最初からPWM信号	開始からPWM信号出力をするかどうかの設定。 - 「Start」：開始からPWM出力を許可します。 - 「Do not start」：「Start PWM」関数を実行するまでPWM出力をしません。

APWM 方式は、単相 PWM 方式と同様に、他の PWM 方式との間で位相をずらした PWM 信号を生成することができます。位相シフトのための PWM ブロックの定義方法についてはセクション 9.4.4 で説明します。

9.4.4 PWM ブロック間の同期 Synchronization Between PWMBlocks

3 種類の PWM ブロックを同期させることができ、相互に位相のずれを定義することができます：単相 PWM、単相 PWM（位相シフト）、単相 PWM（キャプチャと共用）（APWM）の 3 種類があります。

単相 PWM ブロックは、他の PWM 信号に対して位相をずらした PWM 信号を生成することができます。通常の PWM ブロックには 1 つのシリーズがあります：PWM 1, 2, 3, 4, 5, 6, 7, 8.です。

同様に、位相シフトのための APWM ブロックには 1 つのシリーズがあります：PWM 1、APWM 1、2、3 です。

位相シフトのための PWM ブロックの定義は次のようになります。

- ー 基準となる PWM と位相をシフトさせる PWM は、同じシリーズでなければなりません。つまり、PWM1 を基準とし、PWM2、3 などを PWM1 に対して位相シフトさせることができます。または、PWM 2 を基準とし、PWM 3、4 などを PWM 2 に対して位相をシフトさせることができます。

これは、APWM ブロックにも当てはまります。例えば、PWM 1 を基準にして、APWM 1、2 などを位相シフトさせることができます。あるいは、APWM 1 を基準とし、APWM 2、3 などを位相シフトさせることができます。

- ー 基準となる PWM とシフトされる PWM は、スキップされる PWM が使用されない場合を除き、直列に連続していなければなりません。例えば、PWM 1、2、3 をすべて使用するが、PWM 2 が PWM 1 と 3 に同期していない場合、これは許されません。

例えば、PWM1、2、3 をすべて使用しているが、PWM2 が PWM1、3 と同期していない場合は不可です。ただし、PWM 2 が回路内で使用されていない場合は、

PWM 1 を基準にして、PWM 3 を位相シフトさせても問題ありません。

例えば、最初の PWM を基準とし、後続の PWM ブロックの位相をずらした場合、以下の定義が正しくなります。

PWM 1 (reference), PWM 2, PWM 3, PWM 4, PWM 5, PWM 6, PWM 7

PWM 2 (reference), PWM 3

PWM 4 (reference), PWM 5

PWM 5 (reference), PWM 6

PWM 6 (reference), PWM 7

PWM 1 (reference), PWM 2, PWM 3, PWM 4, PWM 5, PWM 6, APWM 1,

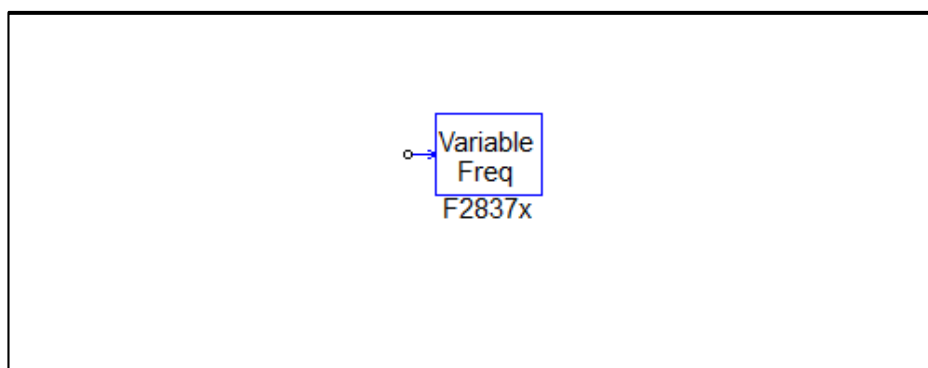
APWM 2, APWM 3

PWM 1 (reference), APWM 1, APWM 2, APWM 3

9.5 可変周波数 PWM

可変周波数 PWM ブロックでは PWM 生成器のサンプリング周波数の変更ができます。

シンボル :



仕様

パラメータ	機能
PWM ソース	PWM 生成器のソース。プルダウンメニューでリストが表示されます。
割り込み位置調整	割り込み位置を周波数で調整するかどうかを指定します。次のいずれかになります。 - <i>Do not adjust(調整しない)</i> : 割り込み位置はベース周波数で計算された初期位置を維持します。 - <i>Adjust(調整する)</i> : 割り込み位置は、新しい周波数に基づき再計算されます、次の PWM 周期で適用されます。
ランプ補正調整	コンパレータ DAC ブロックのランプ補償を調整するかどうかを指定します。次のいずれかになります。 - <i>Do not adjust(調整しない)</i> : ランプ補正はベース周波数で計算された初期補正値を維持します。 - <i>Adjust(調整する)</i> : ランプ補正は新しい周波数に基づき再計算されます。次の PWM 周期で適用されます。

PWM ブロックと一致するサンプリング周波数は次のような PWM 周期の最初に変更されます。

$$PWM_Frequency = PWM_Base \text{ Frequency} / Input_Value$$

ここで *PWM_Base_Frequency* は PWM_block に一致したサンプリング周波数です。*Input_Value* はこのブロックの入力値です。

割り込み位置を調整する場合、各サイクルで割り込み位置が再計算されます。割り込み位置調整は時間が

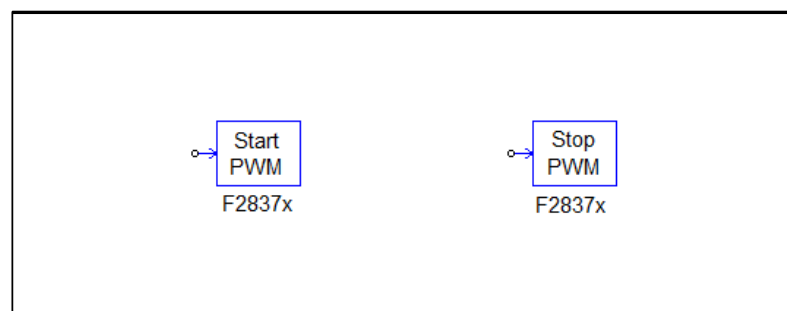
かかるので周波数変化が小さい場合はこの割り込み位置の調整は行わないでください。

同様にランプ補正を調整する場合、各サイクルで割り込み位置が再計算されます。ランプ補正の調整には時間がかかるので周波数変化が小さい場合はこのランプ補正の調整は行わないでください。

9.6 Start PWM と Stop PWM

Start PWM と Stop PWM ブロックは、PWM 生成器の動作を開始する、または停止する機能があります。ブロック図とパラメータは下記に示します。

シンボル:



仕様:

パラメータ	機能
PWM ソース	PWM生成器のソース。 ブルダウナリストのPWMブロックまたはキャプチャのいずれかになります。

9.7 トリップゾーンとトリップゾーンステート

F2837xのPWM DCトリップゾーン信号は、すべてPWM Xバーから出ています。それらは、TRIP1(TZ1)、TRIP2(TZ2)、TRIP3(TZ3)、TRIP4、TRIP5、TRIP7、TRIP8、TRIP9、TRIP10、TRIP11、TRIP12 です。TRIP6はありません。

トリップゾーンは、外部からの故障やトリップ状態を処理するために使用されます。対応する PWM 出力は、それに応じて応答するようにプログラムすることができます。

Trip-Zone State 要素は、トリップゾーン信号の種類、ワンショットかサイクルバイサイクルかを示しています。

9.7.1 PWM トリップゾーン

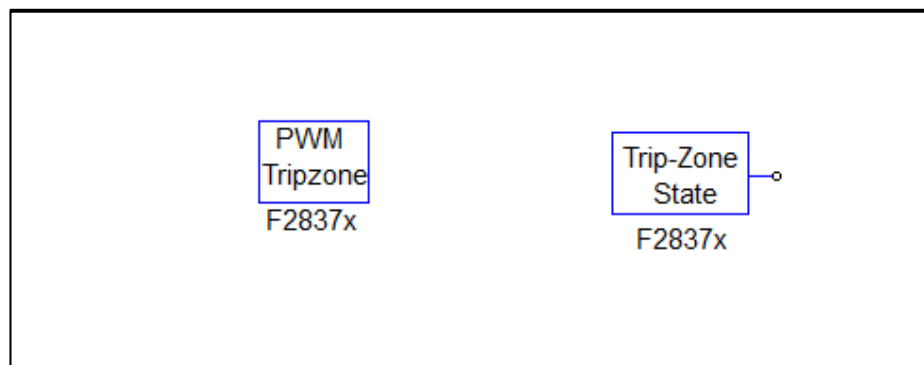
任意の PWM X-Bar 信号を 1 つまたは複数の PWM で使用することができます。トリップゾーン信号で発生する割り込みは、割り込みブロックで処理されます。

デジタル・コンペアによるトリップゾーンの信号は、指定されたアクティブ・レベル (high または low) でトリップ動作を引き起こします。

トリップゾーン割り込みは、PWM ジェネレータのパラメータ入力で定義されているように、ワンショットモードまたはサイクルバイサイクルモードのいずれかで生成することができます。サイクルバイサイクルモードでは、割り込みは現在の PWM サイクル内の PWM 出力にのみ影響を与えます。一方、ワンショットモードでは、割り込みによって PWM 出力が永久的に設定され、動作を再開するためには PWM ジェネレータを再起動する必要があります。

バレースイッチに対応しています。バレースイッチング回路をシミュレーションするには、指定されたトリップゾーン信号に応じて適切なシミュレーションステップを設定する必要があり、通常はより小さな時間ステップを推奨します。1 つのタイムステップでトリップゾーン信号が複数回変化した場合、SimCoder は回路を正しくシミュレーションすることができません。

シンボル:



仕様（トリップゾーン）：

パラメータ	機能
PWMソース	PWM生成器のソースです。PWMブロックのプルダウンリストから選択できます。
DC Trip Src1(DCAH)	PWMxAのデジタルコンペア（DC）トリップソース DCAH。PWMチャンネル は、最大2つのDCトリップソースを持つことができます。DCAHおよびDCALです。トリップソースは次のいずれかになります。 - Do not use : PWMはこの信号を使用しません。 - TZi(Ti) : PWMはトリップゾーンiをデジタルコンペア入力信号として使用します。ここでi=1, 2, 3です。 - TRIPI(Ti) : PWMはTRIPiをデジタルコンペア入力信号として使用します。i=4, 5, 7 - 12. - Other Combinations : PWMはトリップ信号の組み合わせの1つを使用します。 PWMソースが3相PWM生成器の場合、PWMAは"up", "vp", "wp "の3つのトップスイッチの出力を参照します。
PWMA DC Trip Src2(DCAL)	PWMxAのデジタルコンペア（DC）トリップソース DCAL。トリップソースは次のいずれかです。 - Do not use: PWMはこの信号を使用しません。 - TZi(Ti): PWMはトリップゾーンiをデジタルコンペア入力信号として使用します。ここでiは1, 2, 3です。

	- TRIPI(Ti): PWMはTRIPiをデジタルコンペア入力信号として使用します。ここでi=4, 5, 7 - 12. です。 - Other combinations: PWMはトリップ信号の組み合わせの1つを使用します。
PWMA 1-shot Evt (DCAEVT1)	ワンショット信号がPWMAのDCトリップ信号にどのように使われるかを定義します。 DCトリップ信号のアクティブレベルは以下の中から選択できます。 - Do not use: PWMはこの信号を使用しません。 - Trip source 1 is low: ース信号1がHighの場合、PWMがトリップします。 - Trip source 1 is high: ソース信号1がHighの場合、PWMがトリップします。 - Trip source 2 is low: ソース信号2がLowの場合、PWMがトリップします。 - Trip source 2 is high: ソース信号2がハイの場合、PWMがトリップします。 - Trip source 1 low & source 2 high: ソース1の信号がlowで、同時にソース2の信号がhighの場合、PWMがトリップします。ソース1の信号がlowで、同時にソース2の信号がhighの場合、PWMがトリップします。
PWMA CBC Evt (DCAEVT2)	CBC信号がPWMAのDCトリップ信号にどのように使われるかを定義します。PWMA DCトリップ信号のアクティブレベルを以下の中から選択できます。次の通りです。 - Do not use: PWMはこの信号を使用しません。 - Trip source 1 is low: ース信号1がHighの場合、PWMがトリップします。 - Trip source 1 is high: ソース信号1がHighの場合、PWMがトリップします。 - Trip source 2 is low: ソース信号2がLowの場合、PWMがトリップします。 - Trip source 2 is high: ソース信号2がハイの場合、PWMがトリップします。 - Trip source 1 low & source 2 high: ソース1の信号がlowで、同時にソース2の信号がhighの場合、PWMがトリップします。ソース1の信号がlowで、同時にソース2の信号がhighの場合、PWMがトリップします。
DC Trip Src1(DCBH)	PWMxBのデジタルコンペア(DC)トリップソースDCBH。PWMチャンネルは最大2つのDCトリップソースを持つことができます。DCBHとDCBLです。トリップソースは次のいずれかになります。 - Do not use: PWMはこの信号を使用しません。 - TZi(Ti): PWMはトリップゾーンiをデジタルコンペア入力信号として使用します。ここでiは1, 2, 3です。 - TRIPI(Ti): PWMはTRIPiをデジタルコンペア入力信号として使用します。ここでi=4, 5, 7 - 12. です。 - Other combinations: PWMはトリップ信号の組み合わせの1つを使用します PWMソースが3相PWMジェネレータの場合、PWMxBは3つのボトムスイッチの出力 "un"、"vn"、"wn" を参照します。
PWMB DC Trip	PWMxBのデジタルコンペア (DC) トリップソース

Src2(DCBL)	DCBL.トリップソースは次のいずれかになります。 - Do not use: PWMはこの信号を使用しません。 - Tzi(Ti): PWMはトリップゾーンiをデジタルコンペア入力信号として使用します。ここでiは1, 2, 3です。 - TRIPi(Ti): PWMはTRIPiをデジタルコンペア入力信号として使用します。ここでi=4, 5, 7 - 12. です。 - Other combinations: PWMはトリップ信号の組み合わせの1つを使用します
PWMB 1-shot Evt (DCBEVT1)	PWMBのDCトリップ信号にワンショット信号を使用する方法を定義します。DCトリップ信号のアクティブレベルを次の中から選択できます。 - Do not use: PWMはこの信号を使用しません。 - Trip source 1 is low: ソース信号1がHighの場合、PWMがトリップします。 - Trip source 1 is high: ソース信号1がHighの場合、PWMがトリップします。 - Trip source 2 is low: ソース信号2がLowの場合、PWMがトリップします。 - Trip source 2 is high: ソース信号2がHighの場合、PWMがトリップします。 - Trip source 1 low & source 2 high: ソース1の信号がlowで、同時にソース2の信号がhighの場合、PWMがトリップします。ソース1の信号がlowで、同時にソース2の信号がhighの場合、PWMがトリップします。
PWMA CBC Evt (DCBEVT2)	PWMBのDCトリップ信号に、サイクルバイサイクル信号をどのように使用するかを定義します。DCトリップ信号のアクティブレベルは次の中から選択できます。 - Do not use: PWMはこの信号を使用しません。 - Trip source 1 is low: ソース信号1がHighの場合、PWMがトリップします。 - Trip source 1 is high: ソース信号1がHighの場合、PWMがトリップします。 - Trip source 2 is low: ソース信号2がLowの場合、PWMがトリップします。 - Trip source 2 is high: ソース信号2がHighの場合、PWMがトリップします。 - Trip source 1 low & source 2 high: ソース1の信号がlowで、同時にソース2の信号がhighの場合、PWMがトリップします。ソース1の信号がlowで、同時にソース2の信号がhighの場合、PWMがトリップします。
DC Event Filter Source	デジタルコンペアイベントフィルターのソース。次のいずれかを指定します。 - Do not use: DCイベントフィルターを使用しない。 - DCAEVT1 : DCAEVT1 をフィルターソースとして使用します。 - DCAEVT2 : DCAEVT2 をフィルターソースとして使用します。 - DCBEVT1 : DCBEVT1 をフィルターソースとして使用します。 - DCBEVT2 : DCBEVT2 をフィルターソースとして使用します。
Blanking Window Pos (us)	ブランキングウィンドウの位置(us)。
Blanking Window Width (us)	ブランキングウィンドウの幅(us)。

	幅はハードウェアによって制限されており、次のように計算できます。 255 / CPU周波数 例えば、CPU速度が90MHzの場合、幅は0~2.83usとなります。
Blanking Window Range	ブランキングウィンドウの適用方法を指定します。 次のいずれかを指定します。 - ウィンドウの中 ウィンドウを定義した状態でブランキング処理を行います。 (ウィンドウ幅が定義されている場合は開始位置から) - ウィンドウの外 ウィンドウの外側でブランキング処理を行います。
Applying Event Filtering	デジタルコンペアのイベントに適用されるイベントフィルタリングを設定します。 イベントフィルタリングは、以下のデジタルコンペアイベントの任意の組み合わせに適用できます。 DCAEVT1, DCAEVT2, DCBEVT1, DCBEVT2
Enable Valley Switching	パレースイッチ信号に使用するDCイベントを指定します。 以下のDCイベントが選択できます。DCAEVT1, DCAEVT2, DCBEVT1, および DCBEVT2
Toggle Event Selection	トリガーイベントを指定します。このトリガーイベントが "Trigger Event Edge Type "で指定されたエッジを満たすと、PWMはカウントアップを開始します。
Toggle Event Type	トリガーイベントのエッジタイプを指定します。上昇エッジ、下降エッジ、上昇/下降エッジのいずれかを指定します。
Toggle Event Count	トリガーイベントのエッジカウントを指定します。このトリガーイベントカウンタがこの値に等しいとき、PWMはフィルターイベントエッジのカウントを開始します。
Filter Event Start Edge	指定したフィルターイベントの開始エッジカウントを指定します。フィルターイベントは「DCイベントフィルターソース」で選択します。
Filter Event Stop Edge	指定されたフィルターイベントの停止エッジ数を指定します。イベント開始エッジ <= イベント停止エッジ。
Software DelayTime (us)	ソフトウェアの遅延時間 (us)
Delay Calculate Mode	遅延時間の計算モードを指定します。パラメータ「トリガーイベントエッジカウント」で指定された条件を満たすとパラメータ「トリガーイベントエッジカウント」で指定された条件を満たすと、このパラメータで決定された遅延時間後に指定されたフィルターイベントが通過します。 以下のパレーディレイは、フィルターイベント開始エッジとフィルターイベント停止エッジの間隔です。 遅延時間の算出方法は次のいずれかとなります。 - No delay time: 遅延時間は0です。 - Use software delay: 遅延時間はソフトウェアの遅延時間です。 - Use Software delay + valley delay: 遅延時間はソフトウェアの遅延時間+valley遅延です。 - Use Software delay + valley delay/2: 遅延時間はソフトウェアの遅延時間+valley/2遅延です。 - Use Software delay + valley delay/4: 遅延時間はソフトウェアの遅延時間+valley/4遅延です。 - Use Software delay + valley delay/16: 遅延時間はソフトウェアの遅延時間+valley/16遅延です。

--	--

注意：トリップゾーンに関連する割り込みブロックを定義する際、割り込みブロックの「デバイス名」パラメータには、トリップゾーンのブロック名ではなく、PWM生成器の名前を指定する必要があります。例えば、“PWM_G1”というPWM生成器が、トリップゾーンブロック“TZ1”のトリップゾーン1を使用する場合。対応する割り込みブロックの“Device Name”は“TZ1”ではなく“PWM_G1”としてください。この場合、割り込みブロックの「チャンネル番号」パラメータは使用しません。

トリップゾーン割り込みは、PWMジェネレータのパラメータ入力で定義されているように、ワンショットモードまたはサイクルバイサイクルモードのいずれかで生成することができます。サイクルバイサイクルモードでは、割り込みは現在のPWMサイクル内のPWM出力にのみ影響を与えます。一方、ワンショットモードでは、入力信号がL (0) のときにトリップ動作を起こします。このとき、PWM出力は永久的に設定され、動作を再開するにはPWMジェネレータを再起動する必要があります。

9.7.2 トリップゾーンステート

Trip-Zone State 要素は、トリップゾーン信号が PWM ジェネレータをトリガして割り込みを発生させる際に、ワンショットモードかサイクルバイサイクルモードかを示します。

仕様（トリップゾーンステート）：

パラメータ	機能
PWM ソース	PWM生成器のソースの指定。 プルダウンメニューで選択できます。

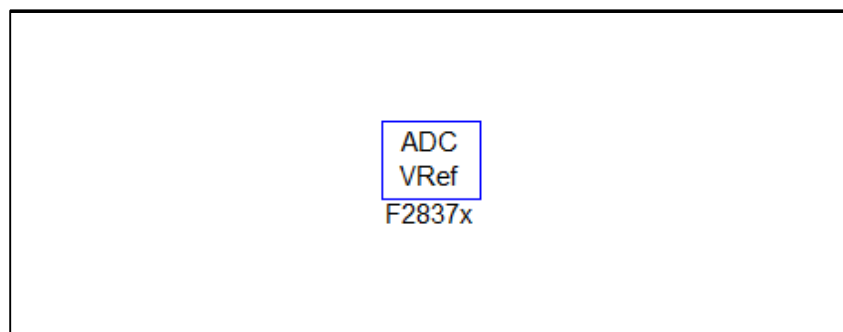
トリップゾーン状態エレメントは、通常、トリップゾーン信号の動作モードを示すために、トリップゾーン割り込みルーチンで使用されます。このエレメントの出力が1の場合、トリップゾーン信号がワンショットモードであることを意味します。

出力が0の場合、トリップゾーン信号はCBCモードになっています。

9.8 ADC 電圧リファレンス

ADC,DAC,コンパレータの電圧基準です。

シンボル：



仕様:

パラメータ	機能
ADC-x High	ADCブロックxの高電圧リファレンスを指定します（xはA、B、C、D）。

Voltage Ref	
ADC-x Low Voltage Ref	ADCブロックx (xはA、B、C、D) の低電圧リファレンスを指定します。この値は必ず0にしてください。
Use ADC-B0 as VDAC	ADC-B0をVDACとして使用する場合の指定
DAC Voltage Ref	VDACの電圧基準値を指定します。

すべての高電圧リファレンス（VDAC を含む）は、2.4V～3.3V の範囲で設定できます。すべての低電圧リファレンスは0に設定する必要があります。

ADC-B0 が VDAC として使用されている場合、DAC と Comparator DAC は VDAC を電圧リファレンスとして使用することができますが、そうでない場合、DAC と Comparator DAC のパラメータダイアログで VDAC を選択することはできません。

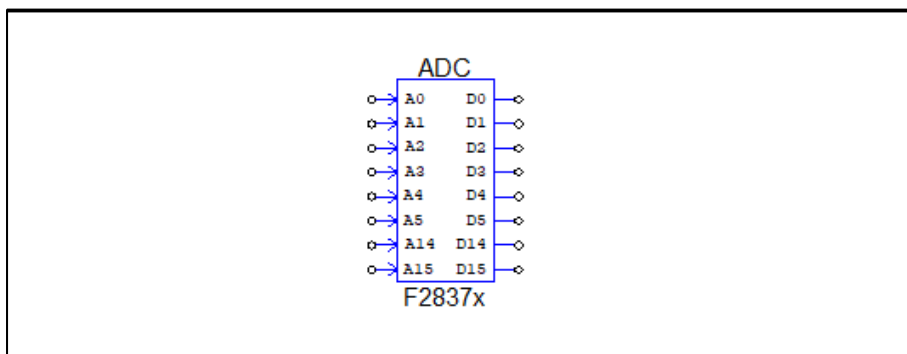
PSIM では ADCB0 に電圧リファレンスを接続する必要はありませんが、ターゲットシステムでは ADC-B0 を電圧リファレンスに接続する必要があります。

9.9 A/D 変換器

F2837x ターゲットには、8 チャンネルのシングルエンド 12 ビット A/D コンバータが 4 つ搭載されています。ADC-A、ADC-B、ADC-C、ADC-D です。

F2837x ターゲット用の SimCoder ライブラリに含まれる A/D コンバータのイメージとパラメータを次に示します。

シンボル:



仕様:

パラメータ	機能
ADC Source	ADCソースを指定してください。ADC-A、ADC-B、ADC-C、ADC-D。
Chn i Gain	i番目のA/Dコンバータチャンネルのゲインです。（iは0から5、14、15の範囲）。
Chn i Sample Time (us)	A/Dコンバータのチャンネルiのサンプルタイムを指定します。iは0～5、14、15の範囲で指定します。
Conversion Order	A/D変換の順序。フィールドを空白（未定義）にした場合、A/Dチャンネルの連番に基づいて変換が行われます。このパラメータで使用するADCチャンネル名は以下の通りです。A0、A1、A2、A3、A4、A5、A14、A15です。例えば、A0,A2,A4,A14,A15を使用した場合、A0,A2,A4,A14,A15の順に変換されます。

	特定のチャンネルを最初に実行したい場合は、ここで順序を定義することができます。例えば、変換順序が次のように定義されているとします。 A4,A0,A2,A14,A15 変換は定義された順序で行われます。つまり、A4を最初に、次にA0、次にA2、というように。
High Priority PIE Selection	最優先の割り込みでPIEグループ1を使用するか、PIEグループ10を使用するかを指定します。

4つのADCコンバータはすべてChn14とChn15を持っていますが、この2つのチャンネルは同じ入力です。

A/Dコンバータの出力は、以下に基づいてスケールリングされます。 $V_{out} = Gain * V_{in}$, V_{in} は A/D コンバータの入力ポートの値です。

各ADCコンバータは独自の電圧リファレンスを持っており、これらの電圧リファレンスはADC Voltage Reference ブロックで指定されます。高電圧リファレンスは2.4V～3.3Vの範囲で設定でき、低電圧リファレンスは0Vに設定する必要があります。

A/Dコンバータは、1つ以上のPWMジェネレータ、タイマ1、またはタイマ2によってトリガされます。回路図において、A/DチャンネルがPWMに関連していない場合、A/Dチャンネルの出力にZOHブロックを接続し、SimCoderが選択したタイマーをトリガースソースとして使用するようになります。

各ADCコンバータは最大4つの割り込みを持ち、1つは割り込み優先度が最も高いPIEグループ1に、他の3つは割り込み優先度が低いPIEグループ10に入っています。

いずれかのサンプリングレートがいずれかのADCチャンネルを使用する場合、SimCoderはそのサンプリングレートにADC割り込みを割り当てます。

パラメータ「High Priority PIE Selection」をPIEグループ1に設定した場合、SimCoderはPIE1ADC割り込みに最高のサンプリングレートを割り当て、その他の低いサンプリングレートはPIEグループ10を使用します。パラメータ「High Priority PIE Selection」をPIEグループ10に設定した場合、SimCoderはこのADCコンバータに関連するすべてのサンプリングレートをPIEグループ10を使用するように割り当てます。

同じサンプリングレートを複数のADCコンバーターで使用する場合、SimCoderはすべてのADCコンバーターのこのサンプリングレートでの合計変換時間をチェックし、変換時間が最も長いADCコンバーターを割り込みを起こすように設定し、他のADCコンバーターは割り込みを起こす必要はありません。このようにして、関連するすべての変換が行われます。

9.10 コンパレータ

F2837xには8つのコンパレータモジュールがあります。ADCの入力チャネルと同じポートを共有しています。PSIMはポートがA/Dコンバータ入力ではなく、コンパレータ入力として定義されたらエラーを出します。

9.10.1 コンパレータ入力

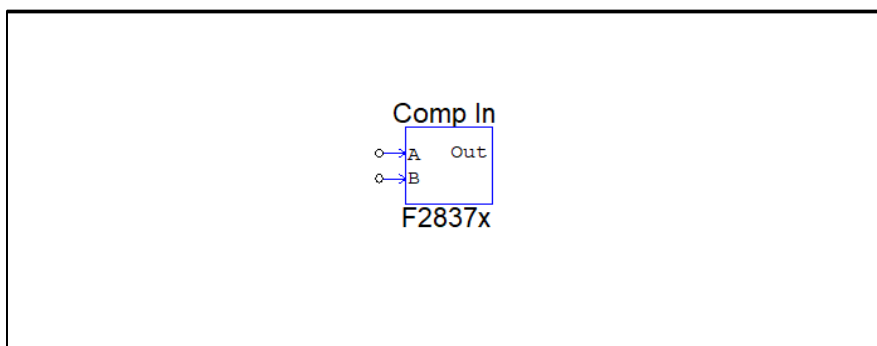
F2837x には 8 つのコンパレータがあります。入力は次のように ADC 入力チャネルをもつ ADC ポートを共用しています。

Comparator Number and Input Name	Port Assigned
Comparator 1 input A and B	ADCA2,ADCA3
Comparator 2 input A and B	ADCA4,ADCA5
Comparator 3 input A and B	ADCB2,ADCAB3
Comparator 4 input A and B	ADCC2,ADCC3
Comparator 5 input A and B	ADCC4,ADCC5
Comparator 6 input A and B	ADCD0,ADCD1
Comparator 7 input A and B	ADCD2,ADCD3
Comparator 8 input A and B	ADC14,ADC15

各ポートに指定できる機能は 1 つだけです。同じ PSIM 回路図において、ポートがコンパレータ入力として定義されていても、A/D コンバータチャネルとしても使用されている場合、Simcoder はエラーを報告します。

各コンパレータには、ハイサイドとローサイドの 2 つのサイドコンパレータがあります。各サイド・コンパレータの入力 A は、常に外部のアナログ入力、サイド・コンパレータの別の入力、または内部の DAC の出力である。入力 B が現在の素子で使用されていない場合は、ADC の入力として使用することができます。

シンボル:



仕様:

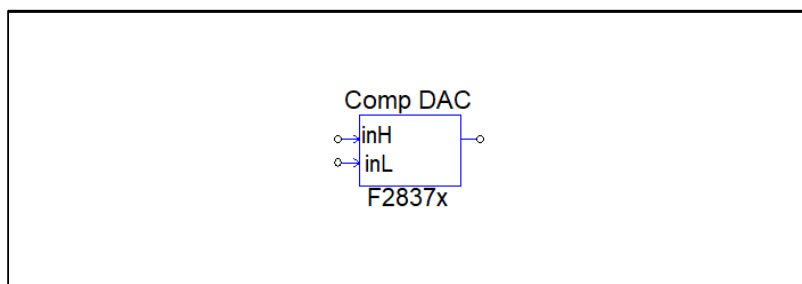
パラメータ	機能
Comparator Source	コンパレータのソースを指定してします。 次のいずれかになります。Comparator1, Comparator2, ..., Compator8.
PWM Sync. Source	このコンパレータの同期PWMを指定します。PWM同期は使用しないか、プルダウンリストのPWM生成器のいずれかを使用します。

PWM's Sync. Pos (us)	このコンパレータのPWM同期位置をus単位で指定します。 コンパレータは、同期信号を受信すると、コンパレータDACの最新の入力値を使用します。
Comparator Usage	どの信号を使用するかを指定します。 - Use TRIPH signal: TRIPH (TRIPOUTH) のみ使用しています。 - Use TRIPL signal: TRIPL (TRIPOUTL) のみ使用しています。 - Use TRIPH/TRIPL signal: TRIPH (TRIPOUTH) と TRIPL (TRIPOUTL) の両方を使用しています。
Input Hysteresis Selection	入力信号に適用するヒステリシスの種類を指定します。 この情報はシミュレーションでは無視されます。次のいずれかを指定します。 - None: ヒステリシスを適用しません。 - Typical hysteresis: 典型的なヒステリシスを適用します。 - 2x hysteresis: 2倍のヒステリシスを適用します。 - 3x hysteresis: 3倍のヒステリシスを適用します。 - 4x hysteresis: 4倍のヒステリシスを適用します。
High Output Invert	ハイサイド・コンパレータの出力を反転させる場合に指定します。
High Output Sync.	ハイサイド・コンパレータの出力の同期モードを指定します。 次のいずれかを指定します。 - Asynchronous: 同期をとりません。 - Sync. by SysClk: 入力信号がシステムクロックに同期します。 - Digital filter result: 入力信号がフィルタリングされます。 - Latched result: 入力信号はフィルタリングされ、アクティブになるとラッチされます。
High Filter Width (us)	ハイサイド・コンパレータのフィルタ・ウィンドウの幅を、us単位で指定します。 この情報は、「High Output Sync.」が「Digital Filter result」または「Latched result」に選択されている場合のみ有効です。
High Threshold (%)	ハイサイドコンパレータのフィルタ閾値を%で指定します。この情報は、「High Output Sync.」が「Digital Filter result」または「Latched result」に選択されている場合のみ有効です。
Low Output Invert	ローサイドコンパレータの出力を反転させる場合に指定します。
Low Output Sync.	ローサイドコンパレータの出力の同期モードを指定します。 以下のいずれかを指定します。 - Asynchronous: 同期をとりません。 - Sync. by SysClk: 入力信号はシステムクロックに同期しています。 - Digital filter result: 入力信号はフィルタリングされます。 - Latched result: 入力信号はフィルタリングされ、アクティブになるとラッチされます。
Low Filter Width (us)	ローサイド・コンパレータのフィルタ・ウィンドウ幅を指定します。(単位: us) この情報は、「Low Output Sync.」が「Digital Filter result」または「Latched result」に選択されている場合のみ有効です。
Low Threshold (%)	Low側コンパレータのフィルタ閾値を%で指定します。この情報は、「Low Output Sync.」が「Digital Filter result」または「Latched result」に選択されている場合のみ有効です。

9.10.2 コンパレータ DAC

各コンパレータ DAC には、2 つの 12 ビットサイド D/A コンバータが搭載されています。サイドコンパレータ DAC の出力は、対応するサイドコンパレータの反転入力（入力 B）に連動しています。また、コンパレータ DAC ブロックは単独では使用できず、コンパレータ入力ブロックと一緒に使用する必要があり、それらは同じコンパレータソースをもっています。

シンボル:



仕様:

パラメータ	機能
コンパレータソース	コンパレータソースは、8つのコンパレータのいずれかになります。
DAC 基準電圧	内側のDACの電圧基準で、「Use VDDA(3.3V)」または「Use VDAC」のいずれかを指定します。
DACH使用	ハイサイドコンパレータのもう一方の信号です。 - Do not use DACH: ハイサイドインナーDACは使用せず、ハイサイドコンパレータを使用する場合はハイサイドコンパレータのもう一方の信号は入力Bとなります。 - Use constant DAC value: ハイサイドインナーDACを使用しますが、DAC の出力は定数となります。生成されたプログラムでは、初期化時にのみこの定数を設定します。 - Use variable DAC value: ハイサイドインナーDAC が使用され、生成プログラムは サンプル周波数に応じて内部DACの値を設定します。 - Use ramp(Constant): 内部DACのハイサイドはランプジェネレータで設定され、ランプジェネレータの最大値は定数となります。生成されたプログラムでは、この定数値は初期化時にのみ設定されます。 - Use ramp(Variable): インナーDACのハイサイドはランプジェネレータで設定され、生成されたプログラムはそのサンプル周波数に応じてランプジェネレータの最大値を設定します。
DACHレンジ	ハイサイドインナーDACの入力範囲です。
DACH初期値	ハイサイドインナーDACの初期値です。 この値は、DACH/Rampが定数値のみを使用する場合、DACH/Rampへの初期化に設定されます。
全ランプ補正	PWM1周期におけるランプジェネレータの総補正量。 これは、1周期におけるランプの総減少量を表します。

ランプ遅延(us)	PWM周期の最初に遅延時間を指定します。遅延時間が終了すると、ランプは減少します。単位はusです。
DACL使用	ローサイドコンパレータのDACの使用方法を指定します。 次のいずれかになります。 - Do not use DACL: ローサイドインナーDACを使用しません。ハイサイドコンパレータのもう一方の信号は、ローサイドコンパレータを使用する場合は入力Bとなります。 - Use constant DAC value: ローサイドインナーDACが使用されていますが、DACの出力は定数となっています。 生成されたプログラムは、初期化でこの定数値を設定するだけです。 - Use variable DAC value: ローサイドのインナーDACを使用し、生成されたプログラムは、そのサンプリング周波数に応じてインナーDACの値を設定します。
DACLレンジ	ローサイドインナーDACの入力範囲です。
DACL初期値	ローサイドインナーDACの初期値です。この値は、DACLが定数値のみを使用する場合、DACLへの初期化時に設定されます。

ランプジェネレータはハイサイドインナーDACにのみ存在します。

ランプジェネレータを使用する場合は、対応するコンパレータ入力ブロックに同期 PWM を対応するコンパレータ入力ブロックに指定する必要があります。

入力値はランプジェネレータの最大値に適用され、入力値はランプジェネレータの最大値に適用され、同期開始からの遅延時間後にランプジェネレータは減少します。PWM 期間が終了するとランプジェネレータは減少します。

ハイサイド・コンパレータの結果がハイになると、ランプ・ジェネレータはランプ値を最大値に戻して減少を止めるため、アクティブな結果は短時間しか残りません。

コンパレータ入力ブロックのパラメータ「High Output Sync.」は、「Latched result」を選択すると良いでしょう。

ラッチされた結果は、次の PWM 周期の初めにリセットされます。

インナーDACを使用した場合、同期がないときは入力値がすぐに DAC 出力に反映されます。

PWM がコンパレータ入力ブロックで指定されている場合、そうでない場合は同期開始時に入力値が DAC 出力に印加されます。PWM の次の周期。

内部 DAC の基準電圧は、VDDA(3.3V)またはVDAC となります。

VDAC を選択した場合、ADC-B0 はVDAC として使用されます。

Psim の回路図では、これは Voltage Reference ブロックで指定するだけで、ADC-B0 に電圧源をリンクする必要はありません。Psim の回路図では、Voltage Reference ブロックで指定するだけで、ADC-B0 に電圧源を接続する必要はありません。

出力は以下のように計算できます。

Use VDDA: $\text{DAC_Output} = \text{DAC_Input} * 3.3 / \text{DAC_Range}$

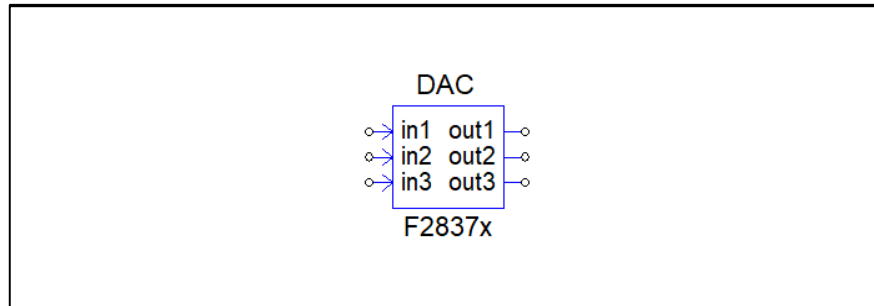
Use VDAC (the voltage reference is specified in Voltage Reference block):

$\text{DAC_Output} = \text{DAC_Input} * \text{VDAC} / \text{DAC_Range}$

9.11 D/A コンバータ

F2837x は、3 つの 12 ビットバッファ付き D/A コンバータ出力をサポートしています。

シンボル:



仕様:

パラメータ	機能
DAC-x 使用	DAC-xで使用される電圧リファレンスで、xはA、B、またはCです。 次のいずれかになります。 - Do not use DAC-A: DAC-xは使用しません。 - ADC Voltage Reference: DAC-xはシステムで使用され、ADC-xの電圧リファレンスはDAC-xによって選択されます。 電圧リファレンスはDAC-xで選択されます。 - VDAC Voltage Reference: DAC-xはシステムで使用されており、VDAC電圧リファレンスはDAC-xで選択されます。
DAC-x 入力レンジ	DAC-xの入力範囲で、xはA、B、Cのいずれかです。
DAC-xPWM同期信号指定	DAC-xがPWMブロックと同期するかどうかを指定します。xはA、B、またはCです。 選択肢は、「使用しない」またはプルダウンリストのPWMブロックのいずれかです。
DAC-x初期値	DAC-xの初期値です。xはA、B、Cのいずれかとなります。

VDACを基準電圧として使用する場合、DACとコンパレータDACはVDACを基準電圧として使用することができます。それ以外の場合、DACとコンパレータDACはVDACを選択できません。PsimではADC-B0に電圧リファレンスを接続する必要はありませんが、ターゲットシステムではADC-B0に電圧リファレンスを接続する必要があります。

DACの基準電圧は、VDACまたはADC-A/Bの基準電圧となります。VDACを選択した場合、ADCBOをVDACに設定する必要があります。PSIMの回路図では、これはVoltage Referenceブロックで指定するだけで、ADC-B0に電圧源をリンクする必要はありません。しかし、ターゲットハードウェアで

は、ADC-B0に電圧リファレンスをリンクする必要があります。

出力は次のように計算できます。

VDAC (the voltage reference is specified in Voltage Reference block)を使う場合:

$$\text{DAC_Output} = \text{DAC_Input} * \text{VDAC} / \text{DAC_Range}$$

ADC-A voltage reference (for DAC-A and DAC-B only)を使う場合:

$$\text{DAC_Output} = \text{DAC_Input} * \text{Vref_ADC-A} / \text{DAC_Range}$$

ADC-B voltage reference (for DAC-C only)を使う場合:

$$\text{DAC_Output} = \text{DAC_Input} * \text{Vref_ADC-B} / \text{DAC_Range}$$

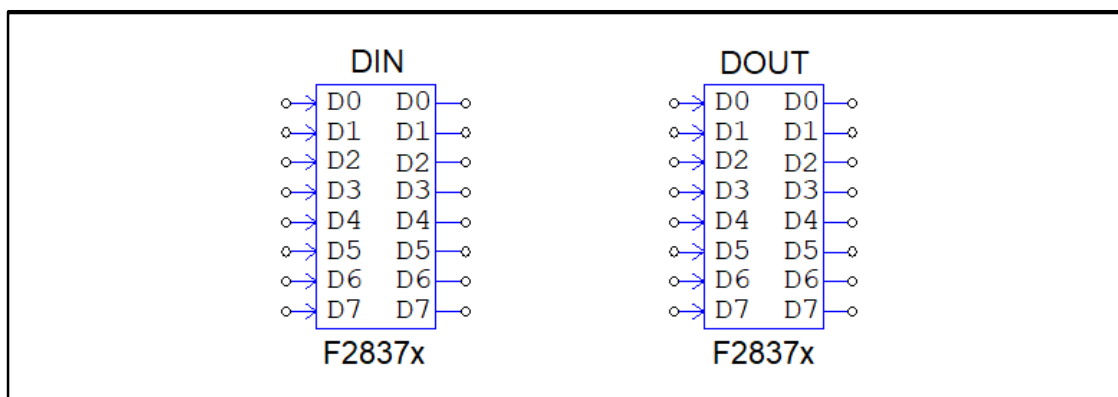
9.12 デジタル入力とデジタル出力

F2837x は、最大 169 個の個別にプログラム可能な多重化された GPIO (General-Purpose Input/Output) ポートをサポートし、入力フィルタリング機能を備えています。

SimCoder では 8 チャンネルのデジタル入出力持つブロックを使うことができます。複数の 8 チャンネルのデジタル入出力ブロックは同じ回路上で使用できます

未使用の入力はすべてグラウンドに接続してください。

シンボル:



仕様 (デジタル入力) :

パラメータ	機能
入力ポート <i>i</i> の位置	入力ポート <i>i</i> の位置で、 <i>i</i> は0~7です。 GPIOポートのいずれかを選択します。
入力ポート <i>i</i> の反転	入力ポート <i>i</i> の信号を反転するか否かを指定します。
入力ポート <i>i</i> にプルアップ抵抗を使用	入力ポート <i>i</i> を内部プルアップ抵抗に接続するか否かを指定します。
入力ポート <i>i</i> の	次の状態認定オプションがあります。

状態認定	・ CPUクロックに同期 ・ 3サンプリング周期 ・ 6サンプリング周期 ・ 同期又は状態認定なし
------	--

仕様（デジタル出力）：

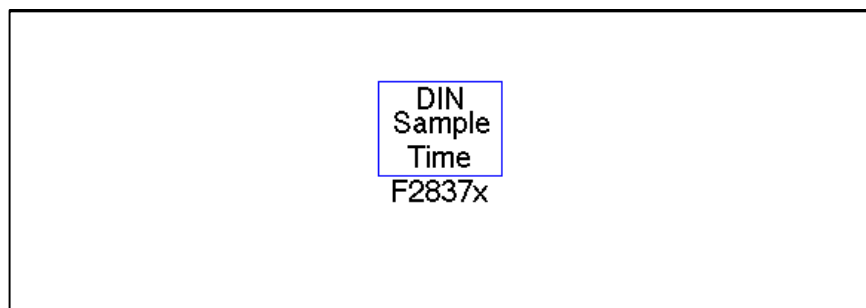
パラメータ	機能
出力ポート <i>i</i> の位置	出力ポート <i>i</i> のポート位置で <i>i</i> は0~7です。GPIOポートのいずれかを選択します。
出力ポート <i>i</i> のモード	出力ポート <i>i</i> のポートモードです。ここで <i>i</i> は0~7です。 次のいずれかになります。 ・ 通常出力 ・ プルアップ抵抗付き出力 ・ オープンドレイン出力

GPIO ポートをデジタル入力ポートとした場合、そのポートは他のペリフェラルのポートとしては使えない点にご注意ください。もし IO ポートがデジタル入力として使用される場合、他のデジタル出力やリフェラルのポートとしては使用できません。例えば GPIO1 はデジタル入力ポートと PWM1 出力の両方の設定ができますが、同時に設定した場合、エラーが出力されます。

9.13 デジタル入力サンプル時間

このブロックはデジタル入力ポートのサンプルタイムを設定します。シミュレーションではなく、コード生成の時に使用されます。

シンボル：



仕様：

パラメータ	機能
Gpio8 <i>i</i> - 8 <i>i</i> +7 サンプル時間設定	- GPIOグループに設定サンプルタイムを使用するかどうかを指定します（範囲は0~21）。
Gpio8 <i>i</i> - 8 <i>i</i> +7 サンプル時間 (us)	- GPIOグループのサンプルタイム（<i>i</i>は0から21まで）。サンプリング時間の単位はusです。

9.14 入力 X-BAR,出力 X-BAR,PWM X-BAR

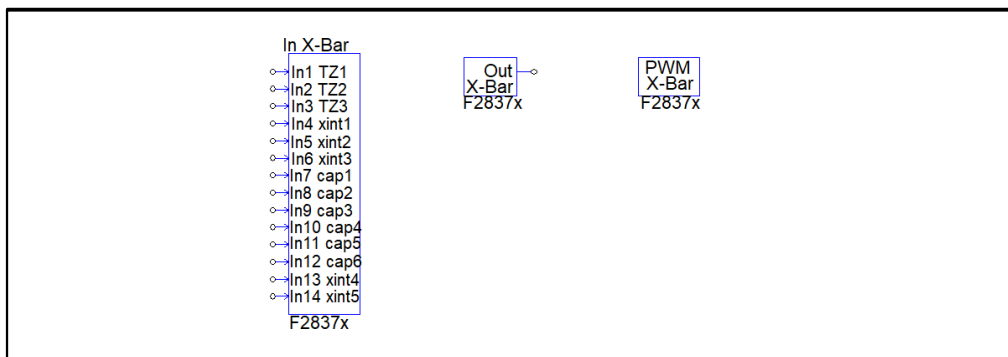
クロスバー（本書では X-BAR と表記）は、デバイスの入力、出力、内部リソースを様々な構成で柔軟に接続することができます。F2837x デバイスには 3 つの X-Bar があります。それぞれの X-BAR は、信号を受ける場所によって名前が付けられています。

入力 X-BAR: 外部信号を GPIO からデバイスに入力します。

出力 X-BAR: デバイスの内部信号を GPIO に出力します。

PWM X-BAR: 信号を受け取り、PWM モジュールに受け渡します。

シンボル:



9.14.1 入力 X-BAR

各々の F2837x システムでは 1 つの X-BAR 入力が可能です。

仕様:

パラメータ	機能
Gpio8*i - 8*i+7 サンプル時間設定	- GPIOグループに設定サンプルタイムを使用するかどうかを指定します（範囲は0～21）。

X バーの入力信号として使用した GPIO ポートは、他の機能としても使用できます。例えば GPIO0 は PWM1A として使用されていますが、入力 X-Bar 信号としても使用されます。

14 の X-Bar 信号 ,INPUT1~INPUT14 があります。

- ・ INPUT1~3 はトリップゾーン信号として使用できます。: TZ1,TZ2,TZ3
- ・ INPUT4~6,13,14 は外部割込みソースとして使用できます。: XINT1,XINT2,XINT3,XINT4,XINT5.
- ・ INPUT7~12 はキャプチャ入力信号として使用されます。対応する入力キャプチャは Capture1 ~6 になります。
- ・ INPUT1~6 は PWMX-Bar と出力 X-Bar でも使用可能で、INPUTXBAR1-INPUTXBAR6 の名前で使用できます。
- ・ INPUT7~12 は、PWM X-Bar や Output X-Bar でも、次の名称で使うことができます。

9.14.2 出力 X-BAR

X バーの出力信号は、OUTPUT1～OUTPUT8 の 8 種類あります。これらの信号は、内部の信号を GPIO ポートに出力するために使用されます。

仕様:

パラメータ	機能
出力X-Bar No.	- GPIOグループに設定サンプルタイムを使用するかどうかを指定します（範囲は0～21）。
出力反転	- 出力を反転させるかどうかを指定します。
出力タイプ	- GPIOの出力タイプを指定します。次のいずれかを指定します。 -Normal Output: プルアップ抵抗無しの通常出力 -Output with pull-up: GPIOポートは内部プルアップ抵抗にリンクします。 -Open drain output: GPIOポートはオープンドレイン出力です（SimCoder はシミュレーションでこの機能をサポートしていません）。
MUXi信号選択	- 次に説明する出力Xバー信号を指定します。

SimCoder は、出力 X-Bar のソースとして以下の信号のみをサポートします。

複数の入力ソースが可能です。

出力は、選択されたすべてのソースの OR です。

MUX 番号	機能 0	機能 1	機能 2	機能 3
0	CMPSS1.CTRIPOUTH	CMPSS1.CTRIPOUTH_OR_CTRIPOUTL	N/A	ECAP1OUT
1	CMPSS1.CTRIPOUTL	INPUTXBAR1	N/A	N/A
2	CMPSS2.CTRIPOUTH	CMPSS2.CTRIPOUTH_OR_CTRIPOUTL	N/A	ECAP2OUT
3	CMPSS2.CTRIPOUTL	INPUTXBAR2	N/A	N/A
4	CMPSS3.CTRIPOUTH	CMPSS3.CTRIPOUTH_OR_CTRIPOUTL	N/A	ECAP3OUT
5	CMPSS3.CTRIPOUTL	INPUTXBAR3	N/A	N/A
6	CMPSS4.CTRIPOUTH	CMPSS4.CTRIPOUTH_OR_CTRIPOUTL	N/A	ECAP4OUT
7	CMPSS4.CTRIPOUTL	INPUTXBAR4	N/A	N/A
8	CMPSS5.CTRIPOUTH	CMPSS5.CTRIPOUTH_OR_CTRIPOUTL	N/A	ECAP5OUT
9	CMPSS5.CTRIPOUTL	INPUTXBAR5	N/A	N/A
10	CMPSS6.CTRIPOUTH	CMPSS6.CTRIPOUTH_OR_CTRIPOUTL	N/A	ECAP6OUT
11	CMPSS6.CTRIPOUTL	INPUTXBAR6	N/A	N/A
12	CMPSS7.CTRIPOUTH	CMPSS7.CTRIPOUTH_OR_CTRIPOUTL	N/A	N/A
13	CMPSS7.CTRIPOUTL	N/A	N/A	N/A
14	CMPSS8.CTRIPOUTH	CMPSS8.CTRIPOUTH_OR_CTRIPOUTL	N/A	N/A
15	CMPSS8.CTRIPOUTL	N/A	N/A	N/A

9.14.3 PWM X-BAR

F2837x の PWM DC トリップゾーン信号はすべて PWM X-Bar から出ています。

仕様:

パラメータ	機能
PWM X-Bar No.	- PWM X-Bar信号を指定します。 - 次のいずれかを指定します。: TRIP4, TRIP5, TRIP7~TRIP12 TRIP6はありません。
出力反転	- 出力を反転させるかどうかを指定します。
MUXi信号選択	- 次に説明する出力Xバー信号を指定します。

PWM X-Bar 信号は、TRIP1~TRIP5、TRIP7~TRIP12 の 11 種類です。TRIP6 の信号はありません。これらの信号は、PWM の DC トリップゾーン信号として使用されます。

TRIP1~TRIP3 は、Input X-Bar の TZ1~TZ3 に直接配線されています。その他の信号については、PWM X-Bar エlementで設定する必要があります。SimCoder は、PWM X-Bar の入力ソースとして以下の信号のみをサポートしています。複数の入力ソースが可能です。出力は、選択されたすべての入力ソースの論理 OR となります。

MUX 番号	機能 0	機能 1	機能 2	機能 3
0	CMPSS1.CTRIPH	CMPSS1.CTRIPH_OR_CTRIPL	N/A	ECAP1OUT
1	CMPSS1.CTRIPL	INPUTXBAR1	N/A	N/A
2	CMPSS2.CTRIPH	CMPSS2.CTRIPH_OR_CTRIPL	N/A	ECAP2OUT
3	CMPSS2.CTRIPL	INPUTXBAR2	N/A	N/A
4	CMPSS3.CTRIPH	CMPSS3.CTRIPH_OR_CTRIPL	N/A	ECAP3OUT
5	CMPSS3.CTRIPL	INPUTXBAR3	N/A	N/A
6	CMPSS4.CTRIPH	CMPSS4.CTRIPH_OR_CTRIPL	N/A	ECAP4OUT
7	CMPSS4.CTRIPL	INPUTXBAR4	N/A	N/A
8	CMPSS5.CTRIPH	CMPSS5.CTRIPH_OR_CTRIPL	N/A	ECAP5OUT
9	CMPSS5.CTRIPL	INPUTXBAR5	N/A	N/A
10	CMPSS6.CTRIPH	CMPSS6.CTRIPH_OR_CTRIPL	N/A	ECAP6OUT
11	CMPSS6.CTRIPL	INPUTXBAR6	N/A	N/A
12	CMPSS7.CTRIPH	CMPSS7.CTRIPH_OR_CTRIPL	N/A	N/A
13	CMPSS7.CTRIPL	N/A	N/A	N/A
14	CMPSS8.CTRIPH	CMPSS8.CTRIPH_OR_CTRIPL	N/A	N/A
15	CMPSS8.CTRIPL	N/A	N/A	N/A

9.15 エンコーダとエンコーダステート、エンコーダインデックス/ストロブポジション

PSIM には、エンコーダ機能を実行するための 3 つのブロックが用意されています。Encoder ブロック、Encoder State ブロック、Encoder Index/Strobe Position ブロックです。

F2837x は 3 つのエンコーダモジュールをサポートしています。

各エンコーダモジュールには、複数の GPIO ポートオプションがあります。

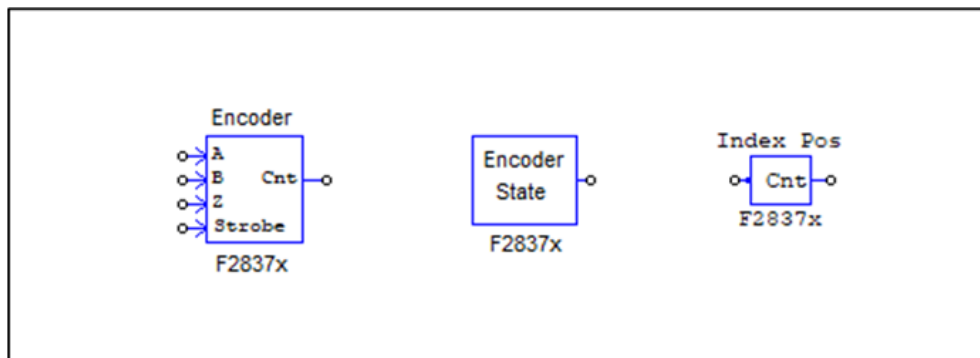
各 GPIO オプションについて。

- －最初の GPIO クロック入力
- －2 番目の GPIO 方向入力
- －3 番目の GPIO Z(もしくは基準信号)入力
- －4 番目の GPIO ストローブ信号入力

エンコーダステートブロックはどの入力信号（基準信号とストローブ信号）が割り込みを発生させる設定なのかを見ることに使います。ハードウェアの割り込みは Z 信号（基準信号）とストローブ信号から生成され、エンコーダステートの出力はどの信号が割り込み発生させているかを示しています。エンコーダステートブロックの出力が 0 のときは、Z 信号（基準信号）から割り込みが発生する設定になっていることを示し、1 のときはストローブ信号から割り込みが発生する設定になっていることを示します。

エンコーダインデックス/ストロボ位置ブロックはエンコーダの初期位置をラッチするのに使用されます。この素子の入力が 0 の時、エンコーダカウンタは 0 に設定されます。入力が 1 になるとエンコーダは、インデックス/ストロボイベントの発生を待ち、発生するとエンコーダのカウント値をラッチします。

シンボル:



仕様（エンコーダ）：

パラメータ	機能
エンコーダソース	エンコーダのソースです。 3つのエンコーダのうち、どのエンコーダでも選択でき、またどのGPIOポートオプションでも選択できます。
Z信号使用	Z信号（基準信号）の有効無効設定。
ストローブ信号使用	ストローブ信号の有効無効設定。
カウンタ方向	カウンタの方向の設定。 - Forward : カウンタは増加方向に進みます。 - Reverse : カウンタは減少方向に進みます。
Z 信号極性	Z信号のトリガの極性 - Active High : 正論理 - Active Low : 負論理
ストローブ信号	ストローブのトリガ極性

極性	- Active High : 正論理 - Active Low : 負論理
エンコーダ分解能	接続するエンコーダハードウェアの分解能の設定です。 ・ 0に設定した場合、エンコーダカウンタはカウントを続けリセットされません。 ・ 4096に設定した場合、カウンタ値が4095になった次のタイミングで0にリセットされます。

仕様（エンコーダ状態）：

パラメータ	機能
エンコーダソース	エンコーダのソースです。 3つのエンコーダのうち、どのエンコーダでも選択でき、またどのGPIOポートオプションでも選択できます。

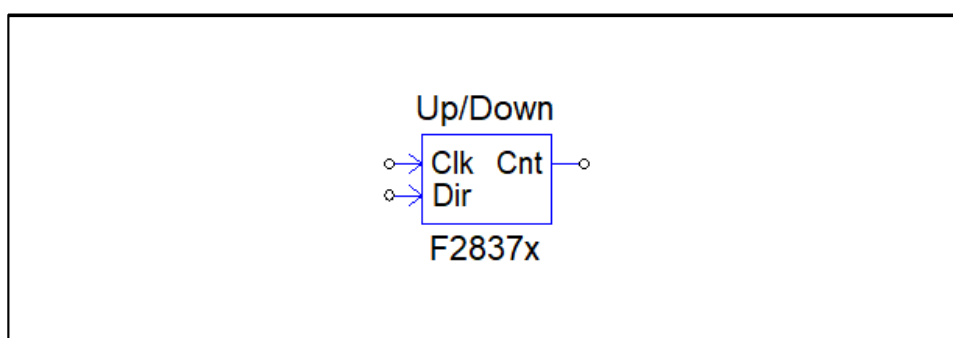
仕様（エンコーダインデックス/ストロブ位置）：

パラメータ	機能
エンコーダソース	エンコーダのソースです。 3つのエンコーダのうち、どのエンコーダでも選択でき、またどのGPIOポートオプションでも選択できます。
ラッチ位置	キープカウンタの種類を次の中から指定します。 - Index位置 エンコーダの「Z信号の使用」が「No」以外の場合 - Strobe位置 エンコーダの「ストロブ信号の使用」が「No」以外の場合
位置タイプ	ラッチ位置の選択です。 - 最初のラッチ位置 - 現在のラッチ位置

9.16 UP/DOWN カウンタ

F2837x は、3つのエンコーダモジュール各々に1つのアップ/ダウンカウンタをサポートしています。

シンボル:



仕様:

パラメータ	機能
カウンタソース	カウンタのソースです。 3つのエンコーダのいずれかで、プルダウンメニューに表示されているGPIOポートオプションのいずれかを使用できます。

Up/Downカウンタには2つの入力があります。

- ・入力"Clk"は入力クロック信号を定義します。
- ・入力"Dir"はカウント方向を定義します。

"Dir"入力が1のとき、カウンタは前方（アップ）にカウントし、0のとき、カウンタは後方（ダウン）にカウントします。

アップ/ダウンカウンタの出力は、カウンタの値です。

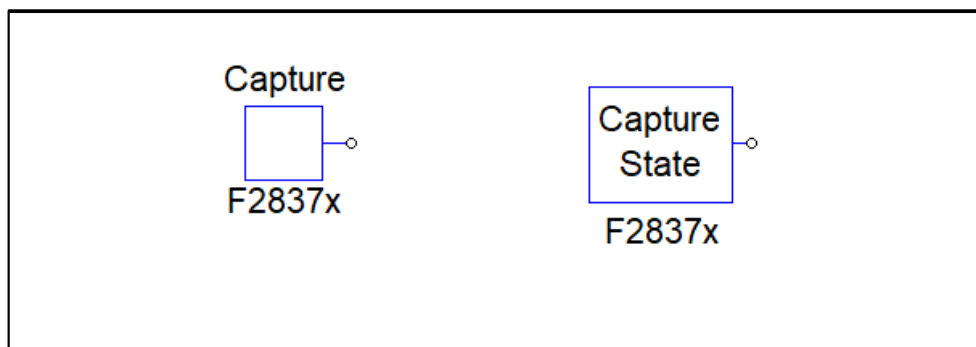
なお、アップ/ダウンカウンタは、エンコーダと同じ入力GPIOポートを共有しています。

そのため、エンコーダとアップ/ダウンカウンタは同じGPIOポートで使用することはできません。

9.17 キャプチャとキャプチャステート

F2837x は 6 つの拡張キャプチャモジュールを搭載しています。PSIM では、Capture ブロックと Capture State ブロックを使って、キャプチャ機能の設定を行います。

シンボル:



仕様（キャプチャ）:

パラメータ	機能
キャプチャソース	キャプチャのソースです。 6つのキャプチャのうちの1つになります。
イベントフィルタ	イベントフィルタのプリスケールです。 入力信号は、選択されたプリスケールで分割されます。
タイマモード	キャプチャカウンタタイマモード設定です。次のどれかになります。 - Absolute time : 出力はシステムクロックの絶対時間です。 - Time difference (clock) : 現在のイベントと前のイベントの差をシステムクロックで出力します。 - Time Difference (us) : 出力は現在のイベントと前のイベントの間の差で、単位はusです。

キャプチャは割り込みを発生させることができ、割り込みブロックで割り込みトリガモードを定義します。回路図に対応する割り込みブロックがない場合、キャプチャは立ち上がりエッジのみをキャッチします。

キャプチャの入力は、入力 X-Bar から来るので、システムでキャプチャを使用する場合は、入力 X-Bar の対応する入力信号も定義する必要があります。

仕様（キャプチャステート）：

パラメータ	機能
キャプチャソース	キャプチャソースです。 Capture1が利用可能な唯一のソースです。

キャプチャステートブロックは、キャプチャのトリガー状態を示します。出力は 1 か 0 で、1 は立ち上がりエッジ、0 は立ち下がりエッジを意味します。

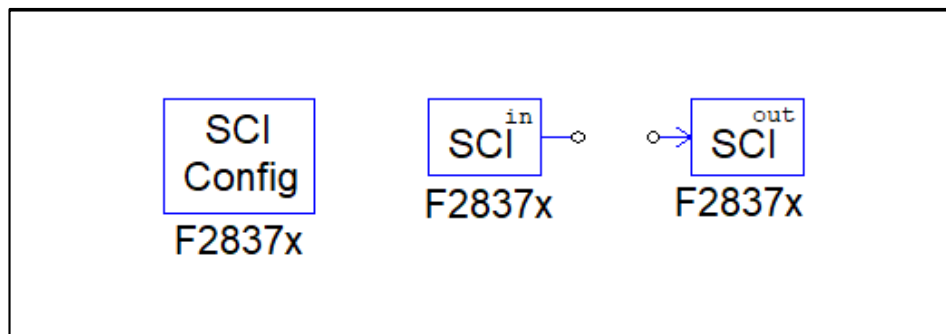
9.18 シリアル通信インターフェース（SCI）

F2837x は、シリアルコミュニケーションインターフェース（SCI）のための関数が用意されています。SCI は、外付けの RS-232 ケーブルを使って DSP とコンピュータの間でデータを転送するものです。PSIM は、DSP 側とコンピュータ側でデータを送受信したり、コンピュータ側でデータを表示したりするために必要な機能をすべて備えています。これにより、DSP のコードをリアルタイムでモニター、デバッグ、調整することができ、非常に便利です。

SCI の詳細な説明とモニタリング機能の使用方法については、「Tutorial - Using SCI for Real-Time Monitoring.pdf」を参照してください。

SimCoder には 3 つの SCI ファンクションブロックが用意されています。以下に説明する SCI Configuration、SCI Input、SCI Output の 3 つです：

シンボル：



9.18.1 SCI 設定

SCI Configuration ブロックは、SCI ポート、通信速度、パリティチェックタイプ、データ、バッファサイズを定義します。

仕様：

パラメータ	機能
ポート選択	F2803xには、SCI機能に使用できる4つのグループのポートがあります。グループ A、B、D、です。各グループには、プルダウンメニューにあるように、異なるポートオプションがあります。
通信速度(bps)	SCIの通信速度をbps（ビット・パー・セカンド）で表しています。

	200000、115200、57600、38400、19200、9600bpsのプリセット速度が用意されています。また、それ以外の速度を手動で指定することもできます。
パリティ チェック	通信時のエラーチェックのためのパリティチェック設定です。 None（なし）、Odd（奇数）、Even（偶数）のいずれかを指定します。
出力バッファ サイズ	DSPでSCI用に割り当てられたデータバッファのサイズです。 バッファはRAMエリアに配置され、各バッファセルには、3つの16ビットワードからなる1つのデータポイントが格納されます（つまり、1つのデータポイントにつき6バイト（48ビット））。

なお、バッファサイズは適切に選択する必要があります。一方で、より多くのデータポイントを収集し、より多くの変数を長期間にわたってモニタするためには、大きなバッファが好ましいです。

一方で、DSP の内蔵メモリには限りがあるため、バッファが大きすぎて DSP の正常な動作に支障をきたすことがないようにする必要があります。

9.18.2 SCI 入力

SCI Input ブロックは、DSP コード内に変更可能な変数を定義します。変数の名前は DSP オシロスコープ（Utilities メニューの下）に表示され、SCI を介してランタイムに値を変更できます。。

仕様:

パラメータ	機能
初期値	SCI入力変数の初期値。

回路図では、SCI 入力は定数として動作します。コードは DSP ターゲット上で実行されている間に短時間で変更することができます。値はシミュレーションの初期値に固定されます。

9.18.3 SCI 出力

SCI 出力ブロックは、表示用の変数を定義するために使用されます。SCI 出力ブロックがノードに接続されている場合、SCI 出力ブロックの名前は、DSP のオシロスコープ（ユーティリティのメニューの下にあります。）に表示され、この変数のデータは、実行時に SCI を介してコンピュータに DSP から送信することができます。波形は DSP オシロスコープで表示することができます。

SCI の出力ブロックは、DSP 波形を観測するための便利な方法を提供します。

仕様:

パラメータ	機能
データポイント ステップ	データの収集頻度を定義します。[Data Point Step]が1の場合、すべてのデータポイントが収集され送信されます。例えば、[Data Point Step]が10の場合、10ポイントのうちの1ポイントのみが収集され送信されます。

[Data Point Step]が小さすぎると、データポイントが非常に多くなる可能性があり、すべてを送信できない場合があることに注意してください。この場合、いくつかのデータポイントがデータの送信時に破棄されます。

また、[Data Point Step]のパラメータは DSP オシロスコープが連続モードの時のみ使用されます。スナップショットノードにある場合、このパラメータは無視され、すべてのポイントを集めて送信されます。

シミュレーションでは、SCI 出力は電圧プローブとして動作します。

9.19 シリアルペリフェラルインターフェース(SPI)

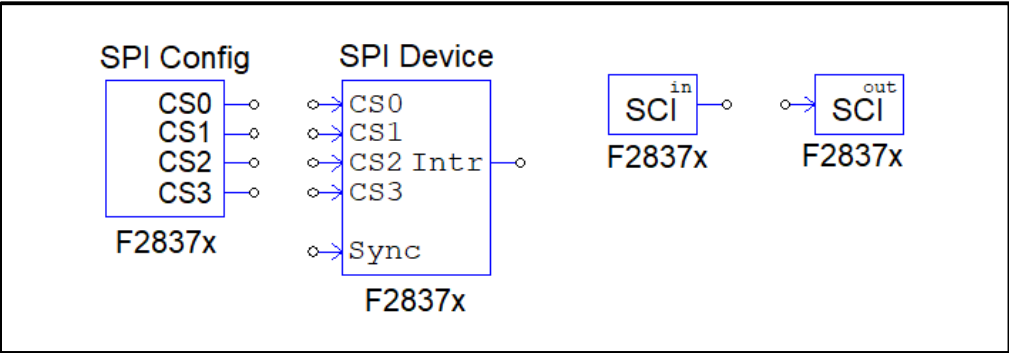
F2837x はシリアルペリフェラルインターフェース（SPI）のための関数が用意されています。TI F2837x ターゲットライブラリの SPI ブロックによって、容易かつ便利に外部の SPI デバイス（外付け A / D および D / A コンバータなど）と通信する機能を実装することができます。

SPI デバイスのために手動でコードを書くことは、しばしば時間がかかり、大変な作業です。SPI をサポートする機能によって PSIM は大幅に簡素化し、コーディングとハードウェアの実装プロセスをスピードアップします。

SPI ブロックを使用する方法についての詳細な説明については、ドキュメントを参照してください”*Tutorial - Using SCI for Real-Time Monitoring.pdf*”。

SimCoder では次に説明する SPI 設定、SPI デバイス、SPI 入力、SPI 出力の 4 つの SCI の機能ブロックが提供されています。

シンボル:



9.19.1 SPI 設定

SPI 設定ブロックは、SPI ポート、チップ選択ピン、および SPI のバッファサイズを定義します。SPI が使用されている回路図に必要で、ブロックはメイン回路図内に配置される必要があります。

仕様:

パラメータ	機能
SPI ポート選択	オプションからSPIポートを定義します。SPIポートは、「SPIA」「SPIB」「SPIC」の3つのポートがあります。 各SPIポートでは、プルダウンメニューでGPIOピンの選択ができます。
チップセレクト Pin0,1,2,3	チップセレクトピンのGPIOポートです。PSIMは、チップセレクトピン0〜3で定義されているように、チップセレクト用の4つのGPIOピンをサポートしています。これらのGPIOポートと、SPIスレーブの送信イネーブル端子SPISTEを使って、チップセレクト信号を生成します。
SPIバッファサイズ	SPI通信バッファサイズ。バッファの各メモリセルは、SPIコマンドのインデックスが保存されます。通常は、すべてのSPIの入力/出力要素の中でSPIコマンド（例え

	ば、Start Conversion Command, Receiving Data Command, Sending Data Command, and Sync. Command) の数に1を足したバッファサイズを指定します。
--	---

9.19.2 SPI デバイス

SPI デバイスのブロックは、対応する SPI ハードウェアデバイスの情報を定義します。回路図の SPI デバイスのブロック数は、SPI ハードウェアデバイスの数と同じでなければなりません。

仕様:

パラメータ	機能
チップセレクトピン	SPIデバイスに対応するチップセレクトピンの状態。チップセレクトピンがこの設定と一致する場合にこのSPIデバイスが選択されます。
通信速度 (MHz)	SPIの通信速度 (MHz)
クロックタイプ	SPIハードウェアデバイスによって決定されるSPIクロックの種類。 -立ち上がりエッジ (遅れなし) : クロックは通常はLOW(0)で、データはクロックの立ち上がりエッジでラッチされます。 -立ち上がりエッジ (遅れあり) : クロックは通常はLOW(0)でデータは遅延付きのクロックの立ち上がりエッジでラッチされます。 -立ち下がりエッジ (遅れなし) : クロックは通常はHigh(1)でデータはクロックの立ち下がりエッジでラッチされます。 -立ち下がりエッジ (遅れあり) : クロックは通常はHigh(1)でデータは遅延付きのクロックの立ち下がりエッジでラッチされます。
コマンドワード長	SPI通信コマンドのワード長、または有効ビットの長さ。1~16ビットにすることができます。
同期信号タイプ	SPIデバイスの同期信号のトリガモード。立ち上がりエッジまたは立ち下がりエッジのいずれかです。
SPI 初期コマンド	SPIデバイスの初期コマンドです。
ハードウェア割り込みモード	SPIデバイスが生成する割り込み信号の種類を指定します。SPIデバイスの割り込み出力ノードがデジタル出力要素の入力に接続されている場合にのみ有効です。次のいずれかになります。 - ハードウェア割り込みなし - 立ち上がりエッジ - 立ち下がりエッジ
割り込みタイミング	変換完了時のSPIデバイスの割り込み生成方法を指定します。次のいずれかになります。 - 割り込みなし 割り込みは生成されません。この場合、DSPIはSPI入力デバイスにコマンドを送信します。デバイスは変換を開始して同じコマンドで結果を返します - 連続多重割り込み : 多重割り込みは、各変換の後に連続で生成されます。 1つのA/D変換ユニットと複数の入力チャンネルを持つSPIデバイス用です。この場合、DSPが最初に変換のコマンドを送信し、SPIデバイスの変換が開始されます。変換が完了すると、SPIデバイスが割り込みを生成します。割り込みサービスルーチンでは、DSPが変換結果をフェッチするためにコマンドを送信し、同じ

	SPI入力デバイスの別のチャンネルの新しい変換を開始します。 - フンタイム割り込み: 変換終了後に1回だけ割り込みが生成されます。1つのリクエストで複数のチャンネルの変換を実行可能なSPIデバイス用です。この場合、DSPはSPI入力デバイスにコマンドを送信し、SPIデバイスは複数の入力チャンネルの変換を完了します。すべての変換が完了すると、SPIデバイスが割り込みを生成します。
コマンド間ギャップ (ns)	2つのSPIコマンドの間隔で単位[nsec]です。
変換シーケンス	変換シーケンスを決定するためのコマンドで区切られたSPIの入力要素の名前を定義します。このパラメータは、SPIデバイスが連続で複数の割り込みを生成する場合にのみ有効であることに注意してください。

回路図では、チップセレクトロジックが実装される方法を定義せずに、すべての SPI デバイスのチップセレクトピンは SPI 設定ブロックのチップセレクト端子に接続されています。しかし、実際のハードウェアでは、対応するチップセレクトロジックに合わせて実装する必要があります。

SPI のコマンドは、カンマで区切られた 16 ビットの一連の数字で構成されています。16 ビットの数値では、下位ビットだけがコマンドによって使用される有効ビットです。例えば、コマンドワード長が 8 の場合、ビット 0~7 はコマンドであり、ビット 8~15 は使用されません。

SPI デバイスは、入力デバイスまたは出力デバイスのいずれかです。例えば、外部 A / D コンバータは、入力デバイスです。通常、DSP はデバイスに 1 つまたは複数の A / D 変換のコマンドを送信し、変換を開始するために同期信号を設定します。同期信号は、同じデバイスの次回のコマンドでリセットされます。

同期信号を使用した SPI の入力デバイスは、通常、割り込みサービスルーチンを入力して DSP をトリガする割り込みピンを必要とします。

一方、外部 D / A コンバータは出力デバイスです。通常、DSP はデバイスに 1 つまたは複数の D / A 変換のコマンドを送信し、変換を開始するために同期信号を設定します。同期信号は、同じデバイスの次回のコマンドでリセットされます。

同期信号を使用する SPI 入力デバイスには、通常、DSP が割り込みサービスルーチンに入るきっかけとなる割り込みピンが必要です。

一方、外付けの D/A コンバータは出力デバイスです。通常、DSP はこのデバイスに 1 つまたは複数の D/A 変換コマンドを送り、同期信号を設定して変換を開始します。同期信号は、同じデバイスの次のコマンドでリセットされます。

9.19.3 SPI 入力

SPI 入力デバイスは、複数の入力チャンネルを持つことができます。SPI 入力ブロックは、SPI 通信用の入力チャンネル、および 1 つの入力チャンネルに対応する 1 つの SPI 入力ブロックのプロパティを定義するために使用されます。

仕様:

パラメータ	機能
デバイス名	SPI入力デバイスの名前です。

変換開始コマンド	コンマで区切られた16進の数字で、変換を開始するコマンドです。 (例 ; 0x23、0x43、0x00)
データ受信コマンド	コンマで区切られた16進の数字で、受信を開始するコマンドです。 (例 ; 0x23、0x43、0x00)
データビット位置	データビットが受信データの文字列のどこにいるか定義します。形式は次のとおりです。 $ElementName = \{Xn[MSB..LSB]\}$ ここで、 - “ElementName”はSPIの入力デバイスの名前です。もし現在のSPI入力デバイスの場合は、代わりにyを使用してください。 - {}は、括弧内の項目が複数回繰り返されることを意味します。 - XnはSPIの入力装置から受信したn番目の単語であり、nは0から始まります。 - MSB..LSBは、ワードの有効ビットの位置を定義します。
入力範囲	入力範囲を定義するパラメータVmaxを指定します。このパラメータは、SPIデバイスがA/Dコンバータの場合にのみ有効です。デバイスの変換モードがDCの場合、入力範囲は0～Vmax。デバイスの変換モードがACの場合、入力範囲は-Vmax/2～Vmax/2。
スケール係数	出力スケールファクタのKscale。スケールファクタが0の場合、SPIデバイスは、A/Dコンバータではなく、結果はDSPがSPI通信で受信したものとまったく同じになります。そうでない場合は、SPIデバイスはA/Dコンバータであり、結果はこの係数とA/D変換モードに基づいてスケールリングされます。
AD変換モード	デバイスのA/D変換モード。変換モードはDCまたはACのどちらでもかまいません。このパラメータはデバイスがA/Dコンバータの場合にのみ有効であることに注意してください。
初期値	入力の初期値です。

データのビット位置の式は、SPI 入力デバイスのデータ長を定義します。例えば、
 $y = x1[3..0]x2[7..0]$ は、データ長が 12 であることを意味し、結果は 2 ワード目の下位 4 ビットと 3 番目のワードの下位 8 ビットです。受信データの文字列は 0x12、0x78、0xAF の場合、結果は 0x8AF です。
 スケールファクタが 0 でない場合、出力は以下に基づいて拡大縮小されます。
 DC 変換モードの時 :

-シミュレーションの場合 $Output = Input \cdot K_{scale}$

-実際のハードの場合 $Output = \frac{Result \cdot V_{max} \cdot K_{scale}}{2^{Data_Length}}$

AC 変換モードの時 :

-シミュレーションの場合 $Output = Input \cdot K_{scale}$

-実際のハードの場合 $Output = \frac{(Result - 2^{Data_Length-1}) \cdot V_{max} \cdot K_{scale}}{2^{Data_Length-1}}$

パラメータ Data_length はデータのビット位置の計算式で計算されます。

9.19.4 SPI 出力

SPI 出力デバイスは、複数の出力チャンネルを持つものもあります。SPI 出力ブロックは、SPI 通信用の出力チャンネル、1 つの出力チャンネルに対応する 1 つの SPI 出力ブロックのプロパティを定義するために使用されます。

仕様:

パラメータ	機能
デバイス名	SPI出力デバイスの名前です。
スケール係数	出力スケール係数Kscale。スケールファクタが0の場合、SPIデバイスは、D/Aコンバータではなく、結果はDSPがSPI通信で受信したものとまったく同じになります。そうでない場合は、SPIデバイスはD/Aコンバータであり、結果はこの係数とD/A変換モードに基づいてスケーリングされます。
出力範囲	出力範囲を定義するパラメータVmaxを指定します。このパラメータは、SPIデバイスがD/Aコンバータの場合にのみ有効です。デバイスの変換モードがDCの場合、入力範囲は0～Vmax、デバイスの変換モードがACの場合、入力範囲は-Vmax/2～Vmax/2。
DAC モード	デバイスのD/A変換モード。変換モードはDCまたはACのどちらでもかまいません。このパラメータはデバイスがD/Aコンバータの場合にのみ有効であることに注意してください。
データ送信コマンド	カンマで区切られた16進の数字で、出力データを送信するコマンドです。 (例； 0x23、0x43、0x00)
データビット位置	データビットが送信データの文字列のどこにいるか定義します。形式は次のとおりです。 素子名(ElementName) = {Xn[MSB..LSB]} ここで、 - “素子名”はSPI出力デバイスの名前です。もし現在のSPI出力デバイスの場合は、代わりにyを使用してください。 - {}は、括弧内の項目が複数回繰り返されることを意味します。 - XnはSPI出力デバイスに送信したn番目の単語であり、nは0から始まります。 - MSB..LSBは、ワードの有効ビットの位置を定義します。
同期コマンド	コンマで区切られた16進の数字で、SPI出力デバイスの出力チャンネルと同期させるコマンドです。(例； 0x23、0x43、0x00) このコマンドは、SPI出力デバイスが同期信号を持っていない場合に使用されます。

データのビット位置の式は、SPI 出力デバイスのデータ長を定義します。例えば、y= x1[3..0]x2[7..0]は、データ長が 12 であることを意味し、結果は 2 ワード目の下位 4 ビットと 3 番目のワードの下位 8 ビットです。送信データの文字列が 0x12、0x78、0xAF の場合、データは 0x8AF です。

スケールファクタが 0 でない場合、出力は以下に基づいて拡大縮小されます。

DC 変換モードの時：

-シミュレーションの場合

$$Output = Input \cdot K_{scale}$$

-実際のハードの場合

$$Output = \frac{Result \cdot K_{scale} \cdot 2^{Data_Length}}{V_{max}}$$

AC 変換モードの時：

-シミュレーションの場合 $Output = Input \cdot K_{scale}$

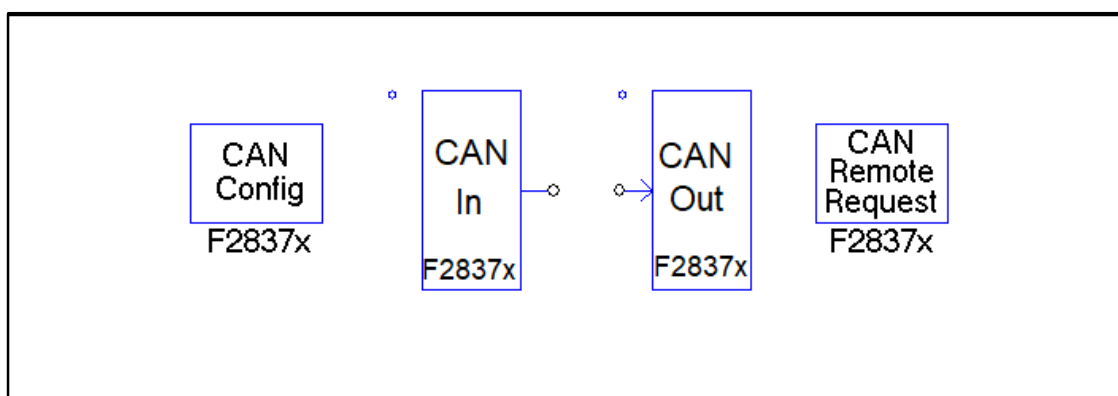
-実際のハードの場合 $Output = 2^{Data_Length} + \frac{Result \cdot K_{scale} \cdot 2^{Data_Length-1}}{V_{max}}$

パラメータ Data_length はデータのビット位置の計算式で計算されます。

9.20 Controller Area Network (CAN) Bus

F2837x は CAN バスコミュニケーションの機能をもっています。PSIM では CAN バスに必要な機能を備えています。SimCoder には *CAN Configuration*、*CAN Input*、*CAN output* *CAN Remote Request* の機能があります。

シンボル:



9.20.1 CAN Configuration

CAN Configuration ブロックは CAN バスソースでデータバイト順や他の設定を定義します。F2837xDSP には 1 つの CAN ソース CAN A があり、送信用の GPIO31 と受信用の GPIO30 を持っています。

仕様:

パラメータ	機能
CAN 速度 (MHz)	Communication Speed (単位はMHz)。 次の周波数、125kHz、250kHz、500kHz、1MHzのどれか1つを選択するかもしくは手入力で設定できます。手入力の場合は1MHzを超えないようご注意ください。
データバイト順	データバイトの順番になります。次の 1 つとなります。

	- Least significant byte 1st - Most significant byte 1st “Least significant byte 1st” は最下位バイトが最初に転送され、最上位バイトが最後に転送されます。一方 “Most significant byte 1st”は最上位バイトが最初に転送され、最下位バイトが最後に転送されます。
入力メールボックス数	入力mailboxの数
受信メッセージ紛失検査	もしEnable(有効)に設定されると受信メールが無くなった場合、エラーレポート関数の “_ProcCanAErrReport(nErr)” (CAN Aに対して) が呼ばれます。 この関数はエラー状態nErrの数値を返します。 もし、Disable(無効)に設定されるとエラーレポート関数は呼ばれません。
バスオフ検査	もしEnable(有効)に設定されると受信メールが無くなった場合、エラーレポート関数の “_ProcCanAErrReport(nErr)” (CAN Aに対して) が呼ばれます。 この関数はエラー状態nErrの数値を返します。 もし、Disable(無効)に設定されるとエラーレポート関数は呼ばれません。
エラー検査モード	エラーチェックモードを受動か能動かに設定できます。受動の場合、エラーカウン트가128になったときに割り込みが発生します。能動の場合割り込みが毎回発生します。

関数 “_ProcCanAErrReport(nErr)” (CAN A に対して) の戻り状態 nErr は 32 ビットの整数です。

エラーとレジスタCANES 状態からの値を含んでいます。関数 “_ProcCanAErrReport(nErr)”からの値が戻るとレジスタCANES はクリアされます。

もし、特定のエラーに対応したい場合は “_ProcCanAErrReport(nErr)”関数に独自のコードを追加することができます。

9.20.2 CAN Input

CAN 入力ブロックは CAN バスから CAN メッセージを受け取ります。

仕様:

パラメータ	機能
入力信号数	CAN入力数です。8入力まで可能です。
拡張ID使用	Yesに設定された場合、メッセージのIDは29-bitの整数となります。 Noに設定された場合、メッセージのIDは11-bitの整数となります。
メッセージID	メッセージのIDです。整数で例えば0x23のようになります。
ローカルマスク	メッセージに対するMASKとなります。整数です。もしメッセージIDのビットがMASKのビットと一致するとメッセージを受信します。そうでない場合は無視されます。例えばmaskが0x380でメッセージIDが0x389の場合、このメッセージが受信されます。もし、メッセージIDが0x480の場合はメッセージは無視されます。
上書き保護フラグ	Allow overwrite(上書き可)かDo not allow overwrite(上書き不可)に設定できます。 例えばメッセージ受信用にメールボックスが設定され、新しいメッセージがCANバス経由で受信されたが古いメッセージがまだメールボックスにあり、何も処理されていない場合、上書き保護フラグが“Allow overwrite”に設定されると新しいメッセージが受信され古いメッセージは上書きされます。Flagが“Do not allow overwrite”に設定されていると新しいメッセージは受信されずに古いものはそのまま残ります。
受信頻度	特定期間に入力ブロックで受信したメッセージの数。例えば、最初の入力ブロックで1秒間に20のメッセージを、2番目の入力ブロックで1秒間に30のメッセージを受

	信する2つの入力ブロックがあるとしします。その場合”Receive Message Rate”は最初のブロックに20、2番目のブロックに30と設定されます。
入力 <i>i</i> ゲイン	<i>i</i> 番目の入力に対するゲインで <i>i</i> は1から8で設定できます。出力は入力のゲインの値をかけた値になります。
入力 <i>i</i> データ開始位置	メッセージは8データポイントまで可能です。データポイントは1bitから32bitsまで可能です。これはメッセージの現在のデータポイントの開始位置を指定します。
入力 <i>i</i> データ終了位置	これはメッセージの現在のデータポイントの終了位置を指定します。
入力 <i>i</i> データタイプ	データタイプは、浮動、整数、またはIQ1からIQ30までとなります。
入力 <i>i</i> デフォルトデータ	SCI入力変数の初期値となります。

9.20.3 CAN Output

CAN 出力ブロックは CAN メッセージを CAN バスへ送ります。

仕様:

パラメータ	機能
出力信号数	CAN出力の数です。8出力まで可能です。
拡張ID使用	Yesに設定された場合、メッセージのIDは29-bitの整数となります。 Noに設定された場合、メッセージのIDは11-bitの整数となります。
メッセージ ID	メッセージのIDです。整数で例えば0x23のようになります。
トリガタイプ	トリガのタイプは次のようになります。 - No trigger : トリガなし。 - Rising edge : トリガソースの立ち上がりエッジでトリガがかかります。 - Falling edge : トリガソースの立下りエッジでトリガがかかります。 - Rising/Falling edge: トリガソースの立ち上がりと立下りのエッジでトリガがかかります。 立ち上がり／立下りエッジはトリガソースの現在の値と、最終のトリガの値が1以上で差がある場合に適用されます。
トリガソース	トリガソースはトリガタイプによるグローバル変数の定数にもあります。 トリガタイプが “No Trigger”に設定されるとトリガソースはカウンターリミットを定義します。例えばトリガソースが定数がグローバル変数で値が5の時、5回に1回トリガがかかることを意味しています。すなわちデータが5サイクル毎に送られます。 もしトリガタイプがエッジトリガに設定された場合、トリガソースはグローバル変数のみです。グローバル変数がトリガタイプに依存して立ち上がりか立下りもしくは両方のエッジとなる時にトリガがかかります。
出力 <i>i</i> ゲイン	<i>i</i> 番目の入力に対するゲインで <i>i</i> は1から8で設定できます。出力は入力のゲインの値をかけた値になります。
出力 <i>i</i> 開始位置	メッセージは8データポイントまで可能です。データポイントは1bitから32bitsまで可能です。これはメッセージの現在のデータポイントの開始位置を定義します。
出力 <i>i</i> データ終了位置	これはメッセージの現在のデータポイントの終了位置を定義します。
出力 <i>i</i> データタイプ	データタイプは、浮動、整数、またはIQ1からIQ30までとなります。
出力 <i>i</i> デフォルトデータ	SCI出力変数の初期値です。

9.20.4 CAN Remote Request

仕様:

パラメータ	機能
CANソース	CANデバイスを選択します。CANAまたはCANBを選択できます。
拡張ID使用	Yesに設定された場合、メッセージのIDは29-bitの整数となります。 Noに設定された場合、メッセージのIDは11-bitの整数となります。
メッセージID	メッセージのIDです。整数で例えば0x23のようになります。
トリガタイプ	トリガタイプです。次のいずれかになります。 - No trigger: トリガがありません。 - Rising edge: トリガソースの立上りエッジでトリガがかかります。 - Falling edge: トリガソースの立下りエッジでトリガがかかります。 - Rising/falling edge: トリガソースの立上り/立下りエッジでトリガがかかります。立ち上り/立ち下りエッジは、次の場合に発生したとみなされます。 トリガソースの現在の値と最後のトリガの瞬間の値の差が等しい場合、立ち上がり/立ち下りエッジが発生したとみなされます。 トリガソースの現在の値と最後のトリガの瞬間の値の差が1以上であれば、立ち上がり/立ち下りエッジが発生したとみなされる。
トリガソース	トリガソースは、トリガタイプに応じて、定数またはグローバル変数になります。 トリガタイプが "No trigger" に設定されている場合は、トリガソースがカウンタリミットを定義します。 例えば、トリガソースが定数またはグローバル値で値が5の場合は 5回に1回の割合でトリガが発生することを意味します。言い換えると、5サイクルに1回、データが送信されることになります。 トリガタイプがエッジトリガに設定されている場合、トリガソースはグローバル変数のみとなります。 トリガは、グローバル変数のエッジの立上りまたは立下りで発生します。 トリガタイプに応じて、グローバル変数の立ち上がりエッジまたは立ち下りエッジ、またはその両方が場合にトリガがかかります。
周波数	サンプリング周波数です。

9.21 割込み時間

割込み時間ブロックは割込みサービスルーチンの時間間隔を測定するのに使われます。

仕様:

パラメータ	機能
Time Output Method	割込み時間測定の設定を行います。 -SCI(使用時間): SCIを使用します。DSPクロックカウンタで割込みサービスルーチンとして使われる時間はSCI出力を経て送られ測定されます。 -SCI(残り時間): SCIを使用します。DSPクロックカウンタの割込みサービスルーチンの残り時間はSCI出力を経て送られ測定されます。残り時間は現在の最後の割込みから次の割り込みの最初までの時間として定義されます。 -GPIO0 to GPIO44又はAIP2 からAIO14: GPIOポートを使います。パルスは特定されたGPIOポートで生成されます。パルスは割込みに入った時high(1)にセットされます。割込みが終了した時にはlow(0)にセットされます。オシロスコ

	ープでパルス幅を測定できます。
サンプリング周波数	割込みサービスルーチンのサンプリング周波数（単位はHz）

SCI が使われる場合、値は DSP クロックでカウントされます。例えば値が 7500 の時 90MHz の DSP クロックの割込み時間は $7500/90M=83.33 \text{ us}$ となります。

9.22 プロジェクト設定とメモリ配置

TI F2837x Hardware Target でコード生成をしようとしたとき、SimCoder は開発環境 TI Code Composer Studio(CCS)のプロジェクトファイルも作成できますので、コンパイル、リンキング、DSP へのアップロードが容易に行えます。

現在、Code Composer Studio のバージョン 3.3 に対応しています。PSIM の回路ファイル名が「test.sch」である場合、コード生成後、「test(C code)」というフォルダが回路ファイルと同じ場所に作成されます。作成されたフォルダには下記のファイルが生成されています。

- test.c : 生成された C コード
- PS_bios.h : SimCoder F2837x のヘッダファイル
- passwords.asm : DSP コードパスワードファイル
- test.pjt : CCS プロジェクトファイル
- DSP2837x_Headers_nonBIOS.cmd : ペリフェラルレジスタリンカコマンドファイル
- F2837_FLASH_Lnk.cmd : フラッシュメモリリンカコマンドファイル
- F2837_FLASH_RAM_Lnk.cmd : フラッシュ RAM リンカコマンドファイル
- F2837_RAM_Lnk.cmd : RAM メモリリンカコマンドファイル

注)：リンカコマンドファイルの名前は F2837 でない場合はターゲットハードウェアによりアサインされます。例えばもしターゲットハードウェアが F28069 の場合、ファイル名はそれに応じて F28069_FLASH_Lnk.cmd, F28069_FLASH_RAMLnk.cmd, F28069_RAM Lnk.cmd となります。

これらのライブラリファイルはコード生成されるたびに自動的にプロジェクトへコピーされます。コード生成を行うたびに、.c ファイルと.pjt ファイル（この例では、「test.c」と「test.pjt」）が作成されます。この 2 つのファイルを手動で変更した場合は、必ず変更したファイルを別の場所にコピーしてください。そうしないと、次のコード生成の際に変更内容が上書きされてしまいます。

プロジェクト設定：

Code Composer Studio のプロジェクトファイルでは、以下の設定を行います。

- RAM Debug：デバッグモードでコードをコンパイルし、RAM メモリーで実行する。
- RAM Release：リリースモードでコードをコンパイルし、RAM メモリーで実行する。
- Flash Release：リリースモードでコードをコンパイルし、フラッシュメモリーで実行する。

-Flasu RAM Release : リリースモードでコードをコンパイルし、RAM メモリーで実行する。

- ・ RAM Debug または RAM Release を選択すると、CCS はリンカー・コマンド・ファイル F2837x_RAM_Lnk.cmd を使用してプログラムとデータの領域を割り当てます。

- ・ FlashRelease を選択した場合、CCS はリンカー・コマンド・ファイル F2837x_FLASH_Lnk.cmd を使ってプログラムとデータの領域を割り当てます。

- ・ Flash RAM Release を選択すると、CCS はリンカ・コマンド・ファイル F2837x_FLASH_RAM_Lnk.cmd を使って、プログラムとデータの領域を割り当てます。メモリの割り当ては RAM Release の場合と同じです。

リリースモードでコンパイルされたコードは、デバッグモードでコンパイルされたコードよりも高速です。また、「RAM Release」または「Flash RAM Release」のコードが最も高速です。RAM Debug のコードは遅く、Flash Release のコードは最も遅くなります。開発時には、通常、デバッグを容易にするために RAM Debug から始めます。その後、RAM Release に切り替え、さらに Flash Release または Flash RAM Release に切り替えます。

10 PE-Expert4 Hardware Target

10.1 概要

SimCoder は Myway プラス製の PE-Expert4 DSP 開発プラットフォームのためのコードを生成できます。

以下のボードに対応しています。

- DSP ボード MWPE4-C6657、コアDSPボード
- PEVボード MWPE4-PEV、PWM生成ボード
- PSPWMボード MWPE4-FPGA24 とMWPE4-FPGA6を含む拡張ボード
- ADボード MWPE4-ADC、アナログ信号入力拡張ボード

複数のサンプルレートをもつシステムに対するコード生成の場合、SimCoderはPWMサンプリングレートではPWM発生器割り込みを使用します。制御システムでは他のサンプリングレートがあるとき、最初にTimer0割り込み、次にTimer1割り込みを使用します。制御システムでは3つを超えるサンプリングレートがある場合、対応する割り込みルーチンはソフトウェアによってメインプログラムで扱われます。

PE-Expert4システムではデジタル入力、エンコーダ、キャプチャがハードウェアの割り込みを生成できます。割り込みはこのマニュアルで説明されている割り込みブロックを使って割り込みサービスルーチン(割り込みサービスルーチンを表示しているサブ回路)に関連づけなければなりません。PE-Expert4ではデジタル入力、エンコーダ、キャプチャ、は同じ割り込みサービスルーチンで扱われるので、すべての割り込みブロックはサブ回路ブロックに接続されていないとなりません。

次に、ハードウェアについて説明します。

10.2 DSP ボード

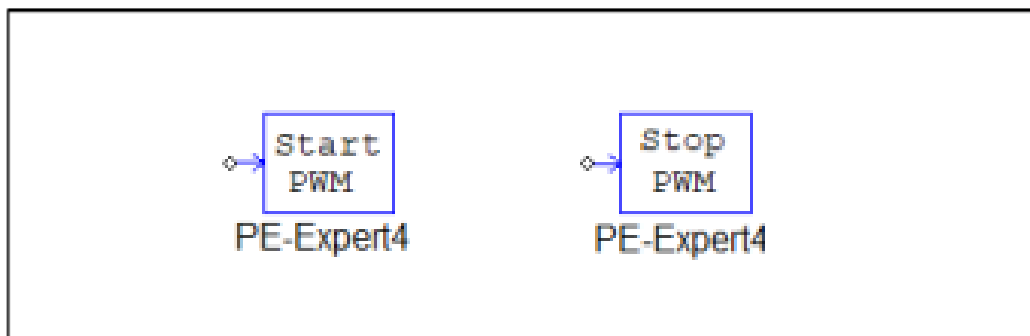
PE-Expert4のDSPボードはコアボードとなります。以下の機能やハードウェア素子が含まれます。

- ースタート・ストップPWM
- ーLED出力
- ータイマー割り込み優先

10.2.1 スタート・ストップ PWM

これらのブロックは、PE-Expert4のPWM機能を開始または停止します。

シンボル:



仕様:

パラメータ	機能
ボードタイプ	Expert4ボードのボードタイプは、PEVボードになります。
ボード番号	起動するPWM生成機を含むPEVボードの番号。

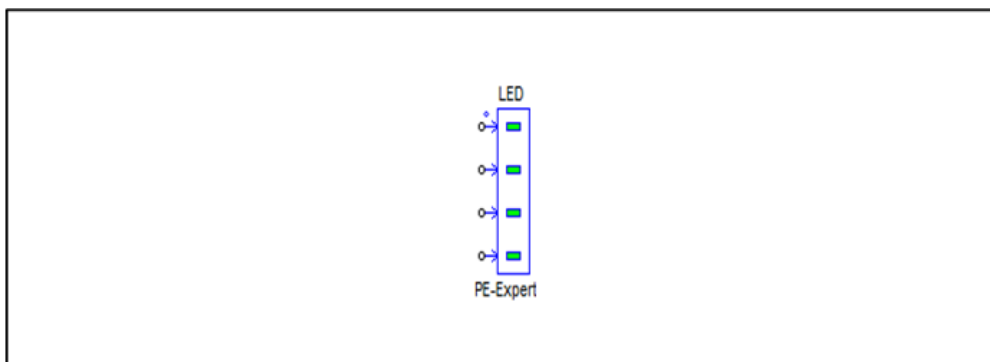
PWMを開始するには、「Start PWM」素子の入力に高論理信号（1）を適用します。PWMを停止するには、「Stop PWM」素子の入力に高論理信号（1）を適用します。

10.2.2 LED 出力

LED 出力素子は DSP ボード(MWPE4-C6657)で提供されています。

DPS ボードには 4 つの LED(LED0,LED1,LED2,LED3)があります。入力が 0.3 以上となった場合に On します。それ以外は Off です。

シンボル:

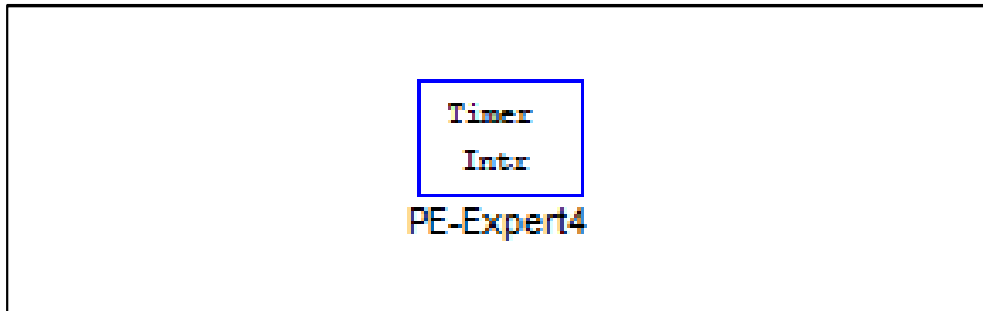


この素子はハードウェア実装のみでシミュレーションでは無視されます。LED 値を表示するためには電圧プローブを入力に接続してください。ドットのある入力が LED0 に対応しています。

10.2.3 タイマー割り込み優先設定

この素子はタイマー割り込み優先順位を設定します。

シンボル:



仕様:

パラメータ	機能
タイマー0の割り込み優先順位	タイマー0の割り込み優先を設定します。
タイマー1の割り込み優先順位	タイマー1の割り込み優先を設定します。

この素子はオプションです。優先順は 4(highest)から 13(lowest)を設定できます。システムにタイマー割り込み優先素子がない場合 SimCoder は自動的にタイマ 0 を割り込み優先順位 10 に、タイマ 1 を割り込み優先順位 11 に設定します。

10.3 PEV ボード

PEVボードには次のハードウェアと機能があります。

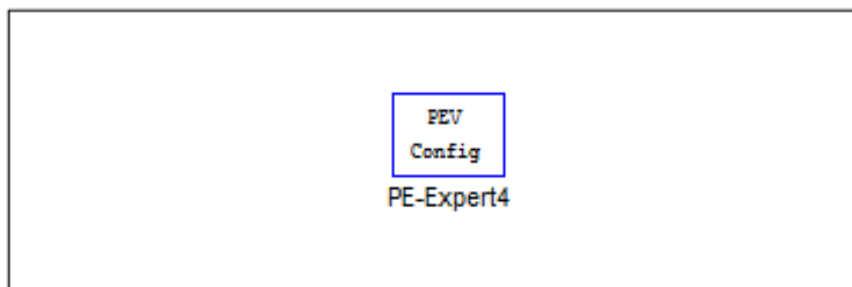
- －PWM生成器
- －A/Dコンバータ
- －デジタル入力／キャプチャ／カウンタ
- －デジタル出力
- －エンコーダ
- －PWMボード構成

PWM発生器、A/D変換器、デジタル入力/出力、アップダウンカウンタ、エンコーダ、キャプチャを含むハードウェア入出力素子はサブ回路内には配置できませんのでご注意ください。主回路のトップレベルにのみ置けます。スタート/ストップPWM関数素子はメイン回路でもサブ回路でも置けます。

10.3.1 PEV ボード設定

この素子は PEV ボードの構成を定義します。PEV ボード素子をのいずれかを PSIM 回路が使用する場合、この設定ブロックを使う必要があります。

シンボル:



仕様:

パラメータ	機能
ボード番号	PEVボードのボード番号(0から4)
intiの割り込みソース	現在のシステムでintiを使用する割り込みソースを特定します。 ADC、PWM、Encoder、Capture、Digital Inputを選択できます。
intiの割り込み優先	現在のシステムでintiを使用する割り込み優先を特定します。 4（最高）から13（最低）まで選択できます。

素子は各割り込み信号の割り込みソースとともに割り込み優先度も定義します。

ADC/PWMの組み合わせの場合、PWM がADCをトリガしない場合 PSIM はPWMを割り込みソースとして設定するので割り込みソースをPWMに設定する必要があります。もしPWMがADCをトリガする場合、PSIMは割り込みソースとしてADCを設定するのでADCを割り込みソースに設定する必要があります。

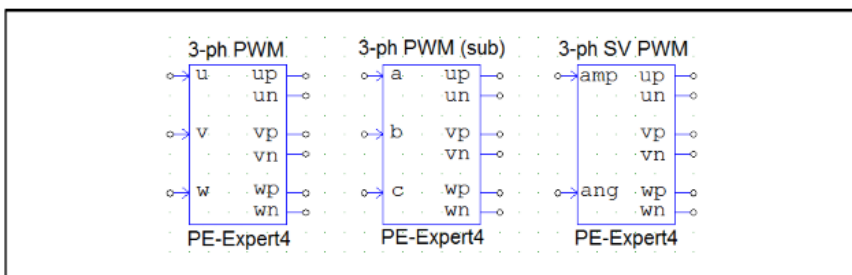
10.3.2 PWM 発生器

PWM信号を発生するために3つの素子が準備されています。

- ・ 3相PWM発生器
- ・ 3相PWM(sub)発生器
- ・ 3相空間ベクトルPWM発生器

です。PWM(sub)発生器はPWM素子に基づいた内蔵モジュールです。

シンボル:



仕様:

パラメータ	機能
ボード番号 スケーリングファクタ	PWM発生器を含んでいるPEVボードのボード番号 スケーリングファクタは、1~1000の範囲の整数で、2つのPWMによる割り込み間のPWM周期の数を指定します。
割り込みモード	割り込みモードは次の設定があります。 <i>Interrupt at peak</i> : 割り込みがPWM搬送波のピーク(山)を基準にします。 <i>Interrupt at valley</i> : 割り込みがPWM搬送波のバレー(谷)を基準にします。 <i>Interrupt at peak/valley</i> : 割り込みがPWM搬送波のピーク(山)/バレー(谷)の両方を基準にします。 割り込みが山/谷の両方にて設定されている場合、搬送波周波数の2倍で割り込みが発生します。搬送波の周波数が10kHzの場合、割り込みレートは20kHzとなります。
割り込み位置	割り込み位置の範囲は-0.5から+0.5です。値が正の場合PWM周期の開始後です。値が負の場合はPWM周期の開始前となります。 A/D変換器が回路に使用されていてその変換モードが“PWMトリガ”に設定されると、このパラメータはPWMをトリガとしたADC変換完了割り込みのタイミング設定となります。PWMがトリガをかけない場合はPWM割り込みの設定となります。 0の場合 A/D変換器(又はPWM割り込み)はPWM周期の最初でトリガされます。 +0.5の場合 A/D変換器(又はPWM割り込み)はPWM半周期でトリガされます。
同期モード	同期モードは次から選択できます。 • Disable • Slave(Sync.iを使用) • Master(Sync.iを使用) iが1~8の場合
デッドタイム	PWM発生器のデッドタイム 単位: 秒
制御周波数	PWM発生器の周波数 単位: Hz
最大振幅値	搬送波のピーク間値。このパラメータはPWM(sub)専用です。
オフセット値	搬送波のDCオフセット値。このパラメータはPWM(sub)専用です。
U相、V相、W相初期値 (またはA相、B相、C相)	PWM生成器の位相u、v、wの初期入力値。またはPWM(sub)発生器の位相a、b、cの初期入力値。
初期振幅	PWM発生器の初期振幅値。空間ベクトルPWM専用です。。
初期角度	PWM発生器の初期角度 単位: ラジアン 空間ベクトルPWMのみ。
PWM信号出力の初期状態	「Start」を設定すると最初からPWM信号を発生します。 「Do not start」を設定すると「PWMゲート信号スタート」素子を使用してPWM信号を開始する必要があります。

PWM生成器は、三相システムのために、正弦波変調PWM信号を生成します。入力「u」、「v」および「w」は、三相入力変調信号で、入力範囲は-1~1です。すなわち、入力が-1である場合、デューティサイクルは0です。また、入力が1である場合、デューティサイクルは1です。入力が0である場合、デューティサイクルは0.5です。搬送波はデューティサイクルは0.5の三角波です。

空間ベクトルPWM発生器は空間ベクトルPWM方式に基づく三相システムのためのPWM信号を生成します。入力の「amp」は空間ベクトルの振幅であり範囲は0~1です。入力「ang」は空間ベクトルの位相角であり範囲は -2π から 4π までです。

PWMジェネレータの入力範囲は-1から1までのため、また、PSIMシミュレーションでは、通常PWM信号は、変調信号を搬送信号と比較することによって生成され、変調信号範囲は-1から1までではないので、スケーリングは変調信号がPWM生成器へ送られる前に必要となります。

PWM(sub)素子はPWM素子と入力スケーリングオフセット回路とから構成されており、次のようになります。コンパレータベースのPWMからPWMハードウェア生成素子への切換えに便利です。

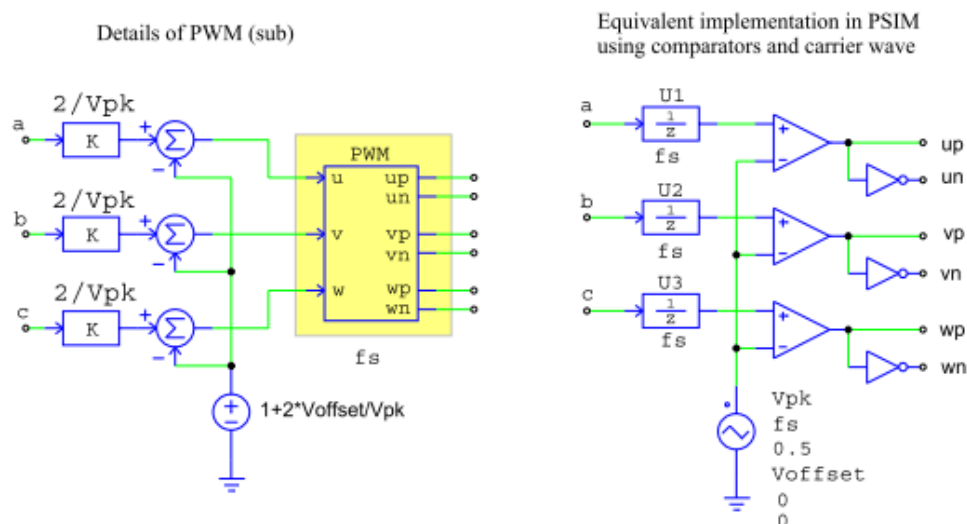


図 10-3-1 PWM(sub)発生器の詳細

PWM(sub)の詳細を図10-3-1の左側に示します。この回路はスケーリングオフセット回路とハードウェアPWM素子から成ります。スケーリングすることによって、入力a、b、およびcの範囲は-1から1までとする必要はありません。キャリア電圧源と同じようにピーク間の値やDCオフセット値を直接定義することができます。

PSIMのコンパレータと三角波キャリア電圧源を利用して作成したPWM(sub)素子の等価回路を図10-3-1の右側に示します。

キャリアソースのパラメータは以下となります。

V _{peak_to_peak} :	Vpk
Frequency:	fs
Duty Cycle:	0.5
DC Offset:	Voffset
Tstart:	0
Phase Delay:	0

3つの単位遅れブロックU1, U2, U3が含まれていることに注意してください。これらは、ハードウェアのPWM素子に存在する1サイクルの遅延効果をモデル化するために使用されます。また、ハードウェアPWM素子の搬送波は三角波であるため、搬送波のデューティサイクルは0.5に固定されています。

スケーリング係数は、PWMによる割り込みが発生する頻度を指定します。

パラメータ「制御周波数」は、サンプリング周波数を指定します。この周波数は、PWMスイッチング周波数と異なる場合があります。

制御周波数 f_{smp} 、スケーリング係数 K_{scale} 、およびPWMキャリア周波数 f_{sw} の関係は、次のように計算されます。

パラメーター「割り込みモード」が「Interrupt at peak」または「Interrupt at valley」として設定されている場合：

$$f_{sw} = K_{scale} * f_{smp}$$

たとえば、 $K_{scale} = 3$ の場合、PWMによる割り込みは3 PWM周期ごとに1回発生します。

パラメーター「割り込みモード」が「Interrupt at peak/valley」として設定されている場合：

$$f_{sw} = K_{scale} * f_{smp} / 2$$

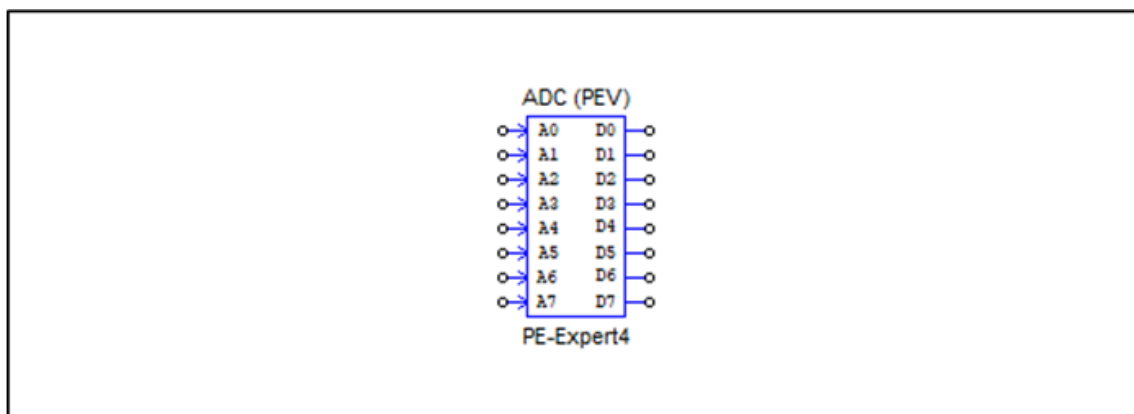
たとえば、 $K_{scale} = 3$ の場合、PWMによる割り込みは $3 \times 2 = 6$ PWM周期ごとに1回発生します。

注) PWMとPWM(sub)生成器がシミュレーション実行される時、デッドタイムは無視されシミュレーションに考慮されません。

10.3.3 A/D 変換器

A/D変換器はアナログ信号をDSPが処理することができるデジタル信号へ変換します。

シンボル:



仕様:

パラメータ	機能
ボード番号	A/D変換器を含んでいるPEVボードのボード番号
AD変換モード	A/D変換器モード(連続かPWMトリガかの選択)

Ai 出力レンジ	i番目のA/Dチャンネルの出力レンジ $V_{range}(i=0から7)$
Ai オフセット	i番目のA/Dチャンネルの出力オフセット $V_{offset}(i=0から7)$

A/D変換器の入力レンジは-5V～+5Vです。A/D変換器の出力レンジは $-V_{range} \sim V_{range}$ です。出力は下記に基づいてスケーリングされます:

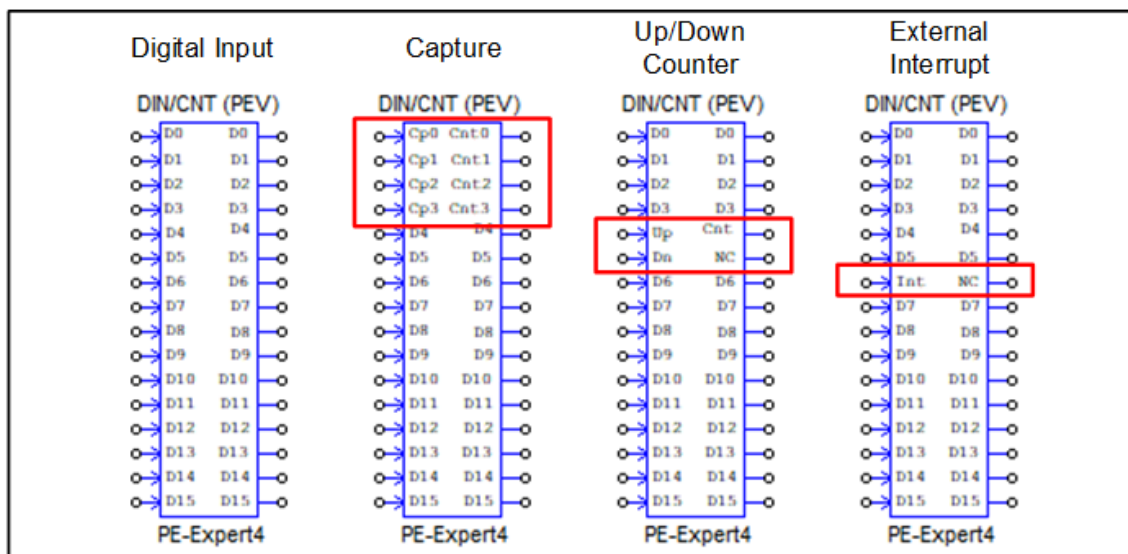
$$V_O = V_i \times V_{range} / 5 - V_{offset}$$

例えば、A/D入力 $V_i = 2V$ 、A/Dレンジ $V_{range} = 20$ 、 $V_{offset} = 0.5$ の場合、出力 V_O は $V_O = 2 \times 20 / 5 - 0.5 = 7.5V$ になります。

10.3.4 デジタル入力・キャプチャ・カウンタ

PEVボードのデジタル入力は16ピンあり、ピンD0～D3はキャプチャ素子と共有され、ピンD4およびD5はアップ/ダウンカウンタと共有されます。そしてピンD6は外部割り込みの入力として使用することができます。従って、入力/出力ピンは定義によって異なった機能に割り当てられます。また素子のシンボルは定義により変化します。各定義毎のシンボルと仕様を次に示します

シンボル:



仕様:

パラメータ	機能
ボード番号	PEVボードのボード番号
入力/キャプチャ i	以下の1つになります: インデックス範囲は0から3、それぞれがD0からD3です。 ■ <i>Digital Input i</i>: 入力ピンD_iはデジタル入力になります。 ■ <i>Capture i</i>: 入力ピンD_iはキャプチャの入力になり、出力ピンD_iはキャプチャiの出力になります。入力・出力ピンのキャプションはCp<i>i</i>とCnt<i>i</i>に変化します。
カウンタソース i	インデックス範囲は0から3、それぞれがD0からD3です。パラメータは

	カウンタソースの名前です。これは、汎用タイマのために「GP_TIMER」あるいはエンコーダの名前のどちらかになります
入力4と5/カウンタ	以下の1つになります： ■ <i>Digital Input4 and 5</i>: 入力ピンD4およびD5はデジタル入力になります。 ■ <i>Counter</i> : 入力ピンD4とD5はアップ・ダウンカウンタの入力になり、出力ピンD4はカウンタの出力になります。カウンタモードでは、出力ピンD5は使用されません。入力ピンはD4の場合Up（アップカウンタ）、D5の場合Dn（ダウンカウンタ）に変化します。出力ピンはD4の場合Cnt（カウンタ出力）、D5の場合NC（接続なし）に変化します。出力ピンD5のキャプションはCnt(カウンタ出力用)に変更され、ピンD6はNC（接続なし）に変更されます。
カウンタモード	入力D4とD5がカウンタ入力として定義されるとき、カウンタモードは「Up/Down」か「Direction/Pulse」のどちらかになります。
入力6/外部割り込み	以下の1つになります： ■ <i>Digital Input 6</i>: 入力ピン6はデジタル入力になります。 ■ <i>External Interrupt</i>: このピンは外部割り込みの入力になります。入力ピンはInt(Interrupt)および出力ピンはNC（接続なし）に変化します。

キャプチャとして:

キャプチャ素子には4つの入力があります。入力がLow(0)からHigh(1)に変わる時、ソースのカウンタ値をキャプチャして出力ポートをから出力します。キャプチャ素子はエンコーダあるいはフリーランタイマ(PEVボードの32ビットフリーランカウンタ)のカウンタ値を保存し出力します。キャプチャすることができます。

カウンタとして:

カウンタには「Up/Down」モードと「Direction/Pulse」モードの2つの操作モードがあります。カウンタが「Up/Down」モードに設定してある場合、「Up」入口にパルスが入力されるとカウンタはカウントアップし、「Dn」入口にパルスが入力されるとカウントダウンします。カウンタが「Direction/Pulse」モードで設定してある場合、「pulse」入口にパルスが入力されたときに「direc」入力が0であればカウントアップし、そして「direc」入力が1ならばカウントダウンします。

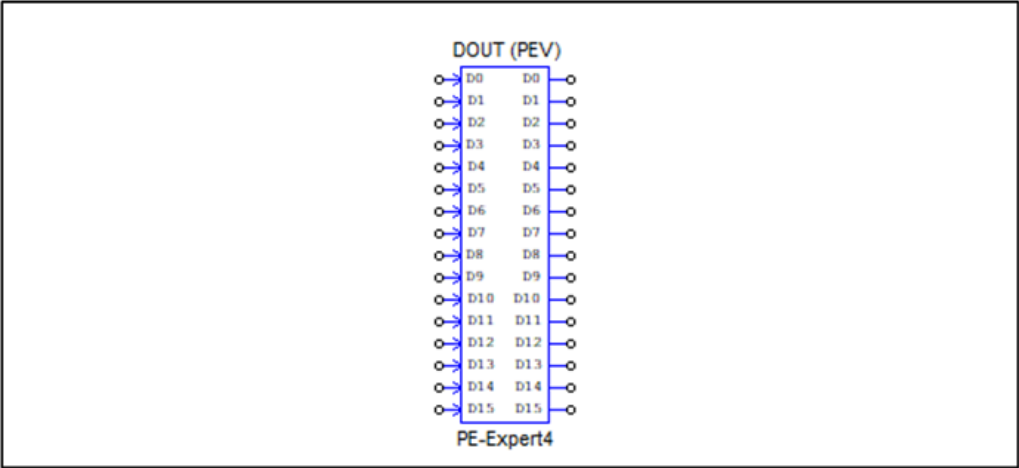
外部割り込みとして:

入力ピンD6が外部割り込みとして定義されている場合、入力が0から1に変化すると割り込みが発生します。

10.3.5 デジタル出力

デジタル出力のシンボルと仕様を以下に示します。

シンボル:



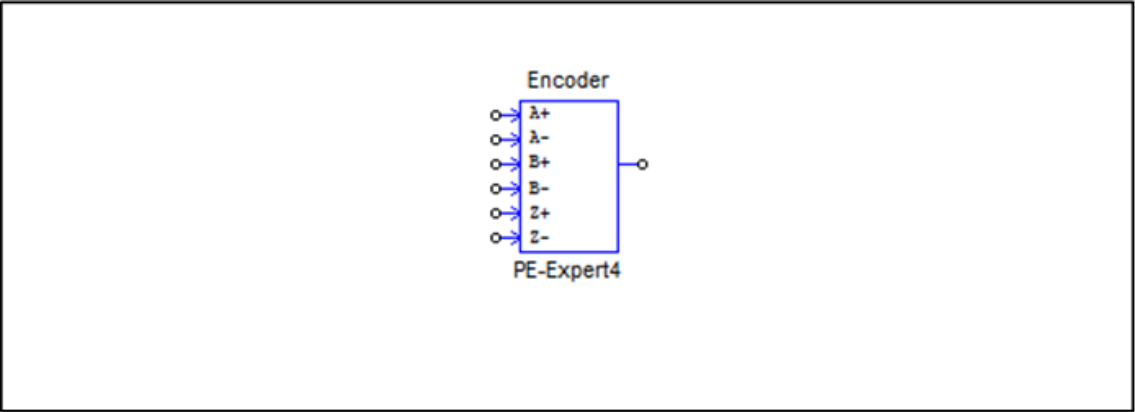
仕様 :

パラメータ	機能
ボード番号	デジタル出力素子を含んでいるPEVボードのボード番号

10.3.6 エンコーダ

エンコーダはモータ駆動システムで位置測定に使用されます。エンコーダは”オープンコレクタ”あるいは”差動入力モード”で動作させることができます。

シンボル:



仕様:

パラメータ	機能
ボード番号	エンコーダを含んでいるPEVボードのボード番号
Z信号の使用	Z信号が使用有無の指定

カウント方向	カウントタ方向(「Forward(順方向)」か「Reverse(逆方向)」)を設定します。 「Forward」に設定するとエンコーダがカウントアップし、「Reverse」に設定するとカウントダウンします。
エンコーダ分解能	エンコーダ分解能を設定します。エンコーダカウンタは1と分解能-1の間でカウントアップ/カウントダウンします。

エンコーダ出力値の出力はカウンタ値となります。また、割り込みは入力信号Z+ とZ-により生成されます。エンコーダはZ信号により割り込みをトリガします。TOはエンコーダ割り込みルーチンを実装しており、割り込み素子を割り込みソース（エンコーダ素子に名前を入力）とどのエッジトリガが割込むか(立下りエッジ、立ち上がりエッジもしくは立ち上がり/立下りエッジなのか)を定義して使用する必要があります。

この素子はイベント接続ワイヤによってイベントサブ回路に接続されます。

10.4 PSPWM ボード

(注：PSPWM ボードは素子名です。実際のハードウェアは IPFPGA ボードです。)

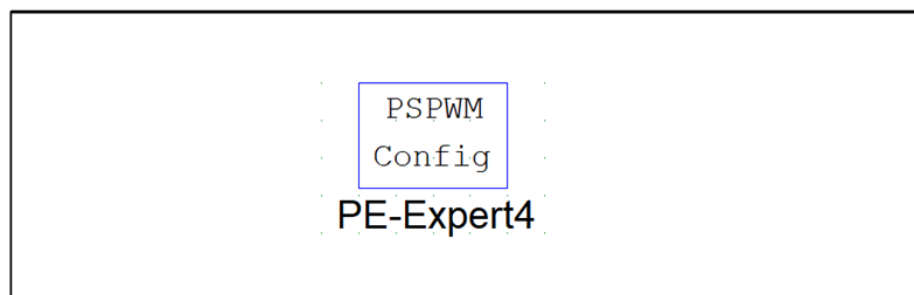
PSPWM ボードには、ハードウェアターゲットとして以下の機能があります。

- PSPWM ボードの構成
- PSPMW
- 2 スイッチ用セル
- 4 スイッチ用セル
- PSPMW (クイックライト)
- 2 スイッチ用セル(クイックライト)
- 4 スイッチ用セル(クイックライト)
- デジタル入力
- デジタル出力

10.4.1 PSPWM ボード構成

この素子は PSPWM ボード構成を定義します。もし PSIM 回路が PSPWM ボードエレメントのいずれかを使用する場合は要素を使用する場合は、このコンフィギュレーション・ブロックを使用する必要があります。

シンボル：



仕様：

パラメータ	機能
ボード番号	PSPWMボードのボード番号(0から4)
inti の割込みソース	現在のシステムでintiを使用する割込みソースを特定します。
inti の割込み優先	現在のシステムでinti を使用する割込み優先を特定します。

この素子は、割り込みソースで使用されるすべての割り込み信号を指定します。

各割り込み信号に対して、その割り込みの優先順位も指定できます。優先度の選択肢は、4（最高）から 13（最低）までです。

10.4.2 PSPWM ボードの PWM 生成器

PSPWM-Extension ボード用のコネクタを搭載しています。これにより 18 チャンネルの PSPWM-Extension ボードを最大 8 枚接続することができます。

次の素子は、PWM 設定のためのものです。

PSPWM：拡張ボード上のセルの PWM 設定で、2 スイッチ/4 スイッチの

セルブロックと組み合わせて使用します。この PSPWM ブロック

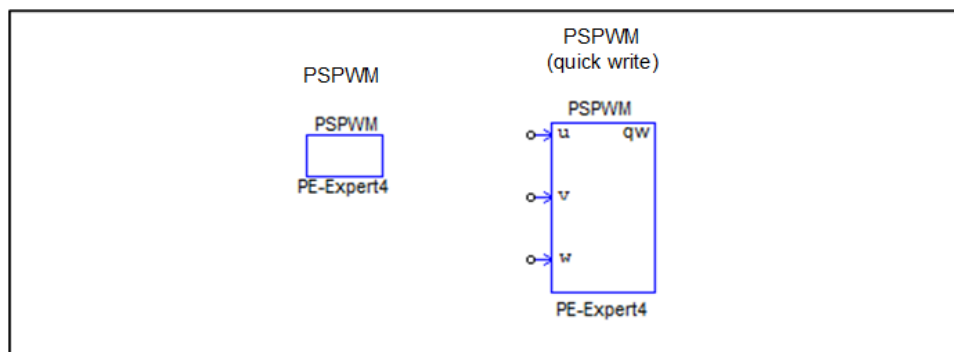
で制御される各セルは、異なる変調値に設定することができます。

PSPMW（クイックライト）：2 スイッチ/4 スイッチのセル（クイックライト）ブロックで使用

する、拡張ボード上のセルの PWM 設定です。各相の変調値は同じです。

シンボルと仕様は次のようになります。

シンボル：



仕様：

パラメータ	機能
ボード番号	PWM生成を含むPEVボードのボード番号。
ボードタイプ	PSPWMボードタイプを設定します。PSPWM-24かPSPWM-144を選択できます。

スケーリングファクタ	スケーリングファクタは、1～1000の範囲の整数で、2つのPWMIによる割り込み間のPWM周期の数を指定します。
入力基準値更新モード	入力基準値更新モードは、次のいずれかに設定できます。 - 各キャリアで：各キャリアの指定位置で実変調値が適用されます。 - 最初のキャリアで：最初のキャリアの指定位置で実変調値が適用されます。
入力基準値更新位置	入力基準値更新位置は、次のいずれかに設定できます： - 山：新しい変調値は、各キャリア又は最初のキャリアのピークに適用されます。 - 谷：新しい変調値は、各キャリア又は最初のキャリアのピークに適用されます。 - 山/谷：新しい変調値は、各キャリア又は最初のキャリアのピークと谷に適用されます。
割り込みモード	PWMの割り込みモードは次のいずれかになります： - Interrupt at peak: 割り込みがPWM搬送波の山を基準にします - Interrupt at valley: 割り込みがPWM搬送波の谷を基準にします - Interrupt at peak/valley: 割り込みがPWM搬送波の山/谷の両方を基準にします 割り込みモードが山と谷の両方に設定されている場合、割り込み周波数は搬送波周波数の2倍で発生します。言い換えると、搬送波の周波数が10kHzの場合、割り込みレートは20kHzになります。
割り込み位置	割り込み位置の範囲は-0.5～+0.5です。値が正の場合、PWMサイクルの開始後になります。値が負の場合は、PWM周期の開始前になります。 A/Dコンバータが使用され、その変換モードが'PWM Triggered'に設定されている場合、このパラメータは、PWMがADCを変換のためにトリガする位置を指定します。 PWMがA/Dコンバータをトリガしない場合は、PWM割り込みのトリガ位置となります。 このパラメータが0のとき、A/Dコンバータ（またはPWM割り込み）はPWMサイクルの開始時にトリガされます。0.5の場合、A/Dコンバータ（またはPWM割り込み）は、PWM周期の半分の位置でトリガされます。
コンバータレベルの数	コンバータのレベル数を指定します。
同期モード	同期モードを以下の中から選択できます。 - 無効 - スレーブ(Sync.iを使用) - マスター(Sync.iを使用) iは1から8までです。
デッドタイム	PWMジェネレータのデッドタイム（秒）
サンプリング周波数	PWMジェネレータのサンプリング周波数（Hz）
動作モード	動作モードは次のいずれかに設定できます。 - ダイオード・クランプ／フライング・コンデンサ ダイオード・クランプ／フライング・キャパシタ：（2スイッチ/セル）

	- MMC (チョッパー) : ハーフブリッジセル (2スイッチ/セル) - MMC H-bridge : フルブリッジセル (4スイッチ/セル)
相数	相数は次のいずれかに設定できます。 - ニュートラル付き1相 - 2leg付1相 - 3相
H-bridge位相使用	フルブリッジセルの2leg間に180度の位相シフトを適用する場合に指定します。 この設定は、ダイオードハーフブリッジおよびクランプ/フライングキャパシタセルでは無視されます。
Shift for Up/Lw Arms	ハーフブリッジまたはフルブリッジセルの上アームと下アームの間の位相差を指定します。 ダイオードクランプ/フライングキャパシタセルの場合、この設定は無視されます。
Start PWM at Beginning	PWMを最初から発生するかどうかを指定します。 - Start : PWM信号を最初から発生します。 - Do not start : "Start PWM"素子を使用してPWM信号発生を開始する必要があります。

PWM 発生器は、三相システム用の正弦波変調 PWM 信号を生成します。入力 "u""v", "w"は三相入力の変調信号で、入力範囲はパラメータ "PeaktoPeak Value"と"Offset Value"によって決定されます。

最大入力値は $(\text{Peak-to-Peak Value})/2 + (\text{Offset Value})$

最小入力値は $(\text{オフセット値}) - (\text{Peak-to-Peak Value})/2$

となります。

PWM のキャリア波形は、立ち上がりとしち下りのスロープの間隔が等しい三角波です。

PSIM では、PWM サイクルが搬送波の "谷"から始まると仮定しています。搬送波の "ピーク"は、PWM サイクルの中間点です。この搬送波の周波数は、PWM スイッチング周波数 fsw と同じです。

スケーリングファクターは、PWMによる割り込みの発生頻度を指定します。

パラメータ「Sampling Frequency」では、サンプリング周波数を指定します。この周波数は、PWM のスイッチング周波数と同じであってもなくても構いません。

サンプリング周波数 fsmp, スケーリング係数 KScale, PWM キャリア周波数 fsw の関係は次のようになります。

パラメータ「Interrupt Mode」が「Interrupt at peak」または「Interrupt at valley」に設定されている場合。

$$fsw = Kscale * fsmp$$

例えば、Kscale = 3 の場合、PWM による割り込みは、3 つの PWM 周期に 1 回発生します。

パラメータ「Interrupt Mode」が「Interrupt at peak/valley」に設定されている場合。

$$fsw = Kscale * fsmp/2$$

例えば、Kscale=3 の場合、PWM による割り込みは $3*2=6$ の PWM 周期に 1 回発生することになります。

同期モードは、マスターモードまたはスレーブモードに設定できます。同期信号の数は 1~8 の間で選択で

きます。システム内の複数の PWM/ADC を同期させるには、関連するすべての PWM/ADC を同じ同期信号に指定する必要があります。同期信号番号を同じにして、1 つの PWM をマスターモード、他の PWM をスレーブモードに設定します。

この PWM が ADC と連動している場合、PSIM は ADC のモードに応じて異なるターゲットプログラムを生成します。ADC モードが "Continuous "の場合、PWM は自分自身で割り込みを起こします。ADC モードが "Pwm Triggered "の場合、同じボードの PWM がパラメータ "Interrupt Position "で指定された一定の時間に ADC をトリガします。

この設定は、PEV4 ボードの構成素子にも影響します。ADC モードが "PWM Triggered "である場合、PEV4 ボード構成素子は ADC を割り込みソースとして設定すべきであり、そうでなければ、ADC をトリガ/同期させる PWM を割り込みソースとして設定します。

パラメータ Number of Converter Levels, Phase Number, Operation Mode により、次表のように PWM 出力の数を指定する。ダイオードクランプ/フライングキャップと MMC (チョッパ) セルにはそれぞれ 2 つのスイッチがあり、H ブリッジセルには 4 つのスイッチがあります。

24 Channel PWM Generator (PSPWM Board) Output Signals			
Conversion Level	Phase Number	PWM outputs for 2sw	PWM outputs for 4sw
3	1 phase with neutral	4	8
	1 phase with 2 legs	8	16
	3 phases	12	24
5	1 phase with neutral	8	16
	1 phase with 2 legs	16	N/A
	3 phases	24	N/A
7	1 phase with neutral	12	24
	1 phase with 2 legs	24	N/A
	3 phases	N/A	N/A
9	1 phase with neutral	16	N/A
	1 phase with 2 legs	N/A	N/A
	3 phases	N/A	N/A
11	1 phase with neutral	20	N/A
	1 phase with 2 legs	N/A	N/A
	3 phases	N/A	N/A
13	1 phase with neutral	24	N/A
	1 phase with 2 legs	N/A	N/A
	3 phases	N/A	N/A

PWM Generator (PSPWM-144 Board) Output Signals

Conversion Level	Phase Number	PWM outputs for 2sw	PWM outputs for 4sw
3	1 phase with neutral	4	8
	1 phase with 2 legs	8	16
	3 phases	12	24
5	1 phase with neutral	8	16
	1 phase with 2 legs	16	32
	3 phases	24	48
7	1 phase with neutral	12	24
	1 phase with 2 legs	24	48
	3 phases	36	72
9	1 phase with neutral	16	32
	1 phase with 2 legs	32	64
	3 phases	48	96
11	1 phase with neutral	20	40
	1 phase with 2 legs	40	80
	3 phases	60	120
13	1 phase with neutral	24	48
	1 phase with 2 legs	48	96
	3 phases	72	144

PWM 出力は、最初の出力端子から順に、U 相出力が割り当てられ、次に V 相、W 相が割り当てられます。

例えば、3 相 5 段、ハーフブリッジセルの場合、PWM 出力は以下のように割り当てられます。

u1p, u1n, u2p, u2n, u3p, u3n, u4p, u4n,

v1p, v1n, v2p, v2n, v3p, v3n, v4p, v4n,

W1P、W1N、W2P、W2N、W3P、W3N、W4P、W4N

ここで

U1P は、フェーズ U の最初のセルの S1p に接続します。

U1N は、U 相の最初のセルの S1n に接続します。

1 相、5 レベル、フルブリッジセルの場合、PWM 出力は以下のように割り当てられます。

u1p1, u1n1, u1p2, u1n2,

u2p1, u2n1, u2p2, u2n2,

u3p1, u3n1, u3p2, u3n2,

u4p1, u4n1, u4p2, u4n2,

残りの 8 つの出力は使用されません。ここで

U1P1 は、フェーズ U の第 1 セルの S1p に接続します。

U1N2 は、フェーズ U の第 1 セルの S1n に接続します。

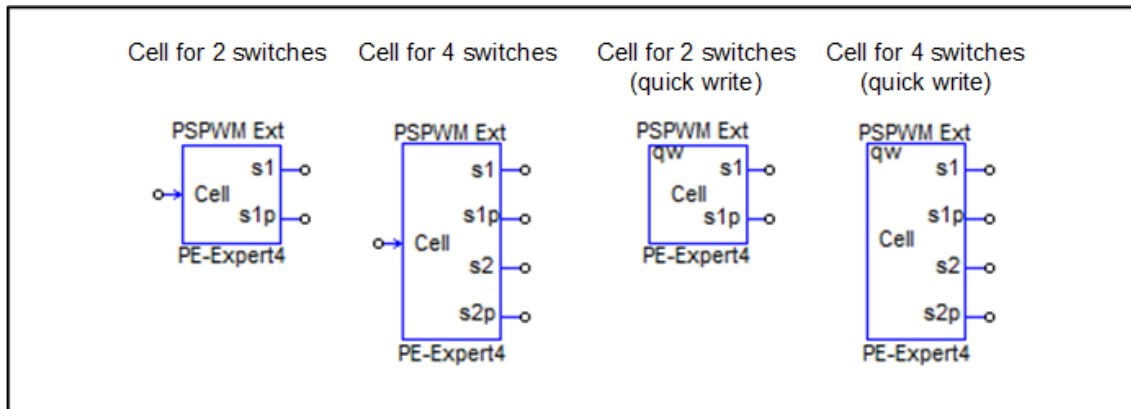
U1P2 は、U 相の第 1 セルの S2p に接続します。

U1N2 は、U 相の最初のセルの S2n に接続します。

PWM を開始するには、「PWM 開始」素子の入力にハイロジック信号（1）を印加します。

10.4.3 PSPWM 拡張ボード上の素子

シンボル :

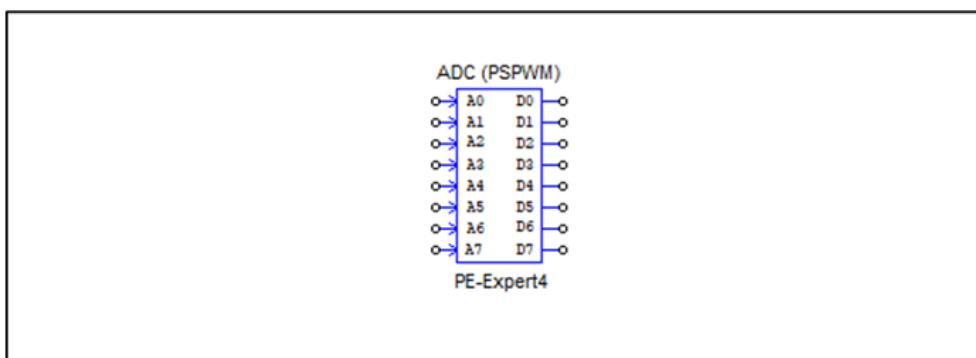


仕様 :

パラメータ	機能
ボード No.	拡張ボードをサポートするメインのPSPWMボードのボード番号です。
位相	位相No.を指定します。U相、V相、W相のいずれかになります。
セル No.	セル番号を指定します。Cell 1、Cell2、...、Cell12.を選択できます。

10.4.4 A/D コンバータ

シンボル :



仕様 :

パラメータ	機能
ボード 番号.	A/Dコンバータを含むPEVボードのボード番号を指定します。
コンバージョンモード	AD変換モードです。連続変換もしくはPWMトリガを選択できます。
Ch Ai Range	i番目のA/Dチャンネルの出力範囲Vrange (i=0~7) .

Ch	Ai	Offset	i番目のA/Dチャンネルの出力オフセットVoffset (i=0~7)
----	----	--------	-------------------------------------

A/D コンバータの入力レンジは-5V~+5V、出力レンジは-Vrange~Vrange です。出力は出力は以下に基づいてスケーリングされます。

$$V_o = V_i * V_{range} / 5 - V_{offset}$$

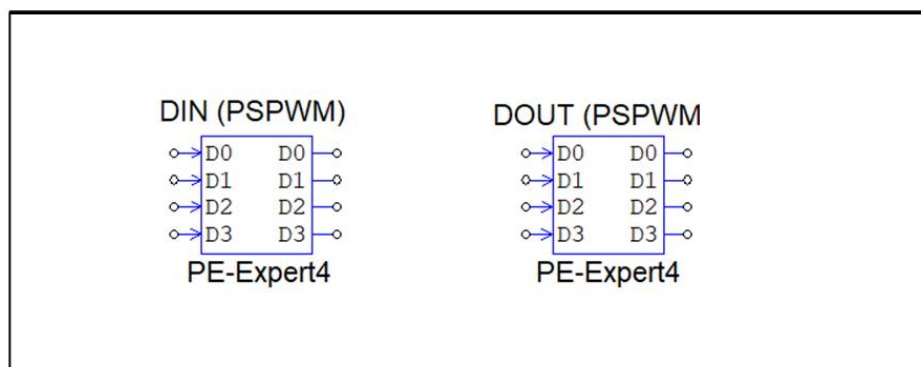
例えば、A/D 入力 $V_i=2$, $V_{range}=20$, $V_{offset}=0.5$ とすると、 $V_o=2 * 20 / 5 - 0.5 = 7.5$ V となります。

この A/D コンバータは、同期信号ではトリガされません。同じ PSPWM ボードでのみ起動することができます。

10.4.5 デジタル入出力

デジタル入出力素子のシンボルと属性を以下に示します。

シンボル：



仕様：

パラメータ	機能
ボード No.	エレメントが搭載されているPSPWMボードのボード番号です。

10.5 ADC ボード

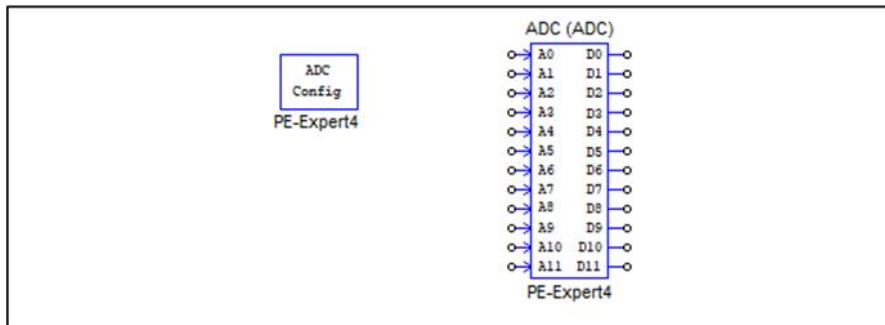
MSPE4-ADC ボードは、14 ビットの分解能を持つ 12 チャンネルの A/D コンバータを搭載しています。この ADC は、アナログ信号を DSP が処理可能デジタル信号に変換します。

このボードを設定するために、2 つの要素が用意されています。

- ADC ボードの設定
- ADC コンバータ

これらの要素のイメージとパラメータを以下に示します。

シンボル：



仕様：ADC ボード構成仕様

パラメータ	機能
ボード番号	PSPWMボードのボード番号(0から4)
inti の割込みソース	現在のシステムでintiを使用する割込みソースを特定します。
inti の割込み優先	現在のシステムでinti を使用する割込み優先を特定します。

この要素では、割り込みソースが使用するすべての割り込み信号を指定し、各割り込み信号に対して、その割り込み優先順位も指定することができます。

仕様：ADC コンバータ仕様

パラメータ	機能
ボード No.	A/Dコンバータを搭載しているPEVボードのボード番号です。
コンバージョンモード	ADCの変換モード。連続またはExt.Sync.ilによるトリガーのいずれかで、ilは1～8です。
Ch Ai Range	i番目のA/Dチャンネルの出力範囲Vrange (i=0～11) .
Ch Ai Offset	i番目のA/Dチャンネルの出力オフセットVoffset (i=0～11)

A/D コンバータの入力レンジは-5V～+5V、出力レンジは-Vrange～Vrange す。出力は以下に基づいてスケールリングされます。

$$V_o = V_i * V_{range} / 5 - V_{offset}$$

例えば、A/D 入力の $V_i = 2$ 、 $V_{range} = 20$ 、 $V_{offset} = 0.5$ とすると、 $V_o = 2 * 20 / 5 - 0.5 = 7.5 \text{ V}$ となります。

10.6 PE-Expert4 ランタイムライブラリ

PE-Expert4は高速計算のためにランタイムライブラリを提供します。SimCoderを利用してコードを作成する時、PSIM素子の対応するPE-Expert4ランタイムライブラリを使用したコードが生成されます。

PSIM素子と対応するPE-Expert4ランタイムライブラリを以下の表に示します。

表 10-6- 1 PSIM 素子と PE-Expert4 ランタイムライブラリ対応表

PSIM素子	PE-Expert4ランタイムライブラリ
Sine or Sine (in rad.)	mwsin(float x)
Cosine or Cosine (in rad.)	mwcoss(float x)
Tangent Inverse	mwarctan2(float y, float x)
Square-root	mwsqrt (float x)
abc-alpha/beta Transformation	uvw2ab(float u, float v, float w, float *a, float *b)
ab-alpha/beta Transformation	uv2ab(float u, float v, float *a, float *b)
ac-alpha/beta Transformation	uw2ab(float u, float w, float *a, float *b)
alpha/beta-abc Transformation	ab2uvw(float a, float b, float *u, float *v, float*w)
alpha/beta-dq Transformation	ab2dq (float a, float b, float *d, float *q)
dq-alpha/beta Transformation	dq2ab (float d, float q, float *a, float *b)
xy-r/angle Transformation	xy2ra (float y, float x, float *a, float * b)
r/angle-xy Transformation	ra2zy (float r, float a, float *x, float *y)

SimCoder User Manual

発行: Myway プラス株式会社
〒220-0022
神奈川県横浜市西区花咲町 6-145
横浜花咲ビル
TEL.045-548-8836
FAX.045-548-8832

ホームページ: <https://www.myway.co.jp>
Eメール: sales@myway.co.jp
