

# **TMS320x28xx, 28xxx** **エンハンスドPWM (ePWM) モジュール**

## **リファレンス・ガイド**

---

この資料は、Texas Instruments Incorporated (TI) が英文で記述した資料を、皆様のご理解の一助として頂くために日本テキサス・インスツルメンツ (日本 TI) が英文から和文へ翻訳して作成したものです。  
資料によっては正規英語版資料の更新に対応していないものがあります。  
日本 TI による和文資料は、あくまでも TI 正規英語版をご理解頂くための補助的参考資料としてご使用下さい。  
製品のご検討およびご採用にあたりましては必ず正規英語版の最新資料をご確認下さい。  
TI および日本 TI は、正規英語版にて更新の情報を提供しているにもかかわらず、更新以前の情報に基づいて発生した問題や障害等につきましては如何なる責任も負いません。



# ご注意

日本テキサス・インスツルメンツ株式会社（以下TIJといいます）及びTexas Instruments Incorporated（TIJの親会社、以下TIJないしTexas Instruments Incorporatedを総称してTIJといいます）は、その製品及びサービスを任意に修正し、改善、改良、その他の変更をし、もしくは製品の製造中止またはサービスの提供を中止する権利を留保します。従いまして、お客様は、発注される前に、関連する最新の情報を取得して頂き、その情報が現在有効かつ完全なものであるかどうかをご確認下さい。全ての製品は、お客様とTIJとの間に取引契約が締結されている場合は、当該契約条件に基づき、また当該取引契約が締結されていない場合は、ご注文の受諾の際に提示されるTIJの標準販売契約約款に従って販売されます。

TIは、そのハードウェア製品が、TIの標準保証条件に従い販売時の仕様に対応した性能を有していること、またはお客様とTIJとの間で合意された保証条件に従い合意された仕様に対応した性能を有していることを保証します。検査およびその他の品質管理技法は、TIが当該保証を支援するのに必要とみなす範囲で行なわれております。各デバイスの全てのパラメーターに関する固有の検査は、政府がそれ等の実行を義務づけている場合を除き、必ずしも行なわれておりません。

TIは、製品のアプリケーションに関する支援もしくはお客様の製品の設計について責任を負うことはありません。TI製部品を使用しているお客様の製品及びそのアプリケーションについての責任はお客様にあります。TI製部品を使用したお客様の製品及びアプリケーションについて想定される危険を最小のものとするため、適切な設計上および操作上の安全対策は、必ずお客様にてお取り下さい。

TIは、TIの製品もしくはサービスが使用されている組み合わせ、機械装置、もしくは方法に関連しているTIの特許権、著作権、回路配置利用権、その他のTIの知的財産権に基づいて何らかのライセンスを許諾するということは明示的にも黙示的にも保証も表明もしておりません。TIが第三者の製品もしくはサービスについて情報を提供することは、TIが当該製品もしくはサービスを使用することについてライセンスを与えるとか、保証もしくは是認するということを意味しません。そのような情報を使用するには第三者の特許その他の知的財産権に基づき当該第三者からライセンスを得なければならない場合もあり、またTIの特許その他の知的財産権に基づきTIからライセンスを得て頂かなければならない場合もあります。

TIのデータ・ブックもしくはデータ・シートの中にある情報を複製することは、その情報に一切の変更を加えること無く、かつその情報と結び付けられた全ての保証、条件、制限及び通知と共に複製がなされる限りにおいて許されるものとします。当該情報に変更を加えて複製することは不正で誤認を生じさせる行為です。TIは、そのような変更された情報や複製については何の義務も責任も負いません。

TIの製品もしくはサービスについてTIにより示された数値、特性、条件その他のパラメーターと異なる、あるいは、それを超えてなされた説明で当該TI製品もしくはサービスを再販売することは、当該TI製品もしくはサービスに対する全ての明示的保証、及び何らかの黙示的保証を無効にし、かつ不正で誤認を生じさせる行為です。TIは、そのような説明については何の義務も責任もありません。

TIは、TIの製品が、安全でないことが致命的となる用途ないしアプリケーション（例えば、生命維持装置のように、TI製品に不良があった場合に、その不良により相当な確率で死傷等の重篤な事故が発生するようなもの）に使用されることを認めておりません。但し、お客様とTIの双方の権限有る役員が書面でそのような使用について明確に合意した場合は除きます。たとえTIがアプリケーションに関連した情報やサポートを提供したとしても、お客様は、そのようなアプリケーションの安全面及び規制面から見た諸問題を解決するために必要とされる専門的知識及び技術を持ち、かつ、お客様の製品について、またTI製品をそのような安全でないことが致命的となる用途に使用することについて、お客様が全ての法的責任、規制を遵守する責任、及び安全に関する要求事項を満足させる責任を負っていることを認め、かつそのことに同意します。さらに、もし万一、TIの製品がそのような安全でないことが致命的となる用途に使用されたことによって損害が発生し、TIないしその代表者がその損害を賠償した場合は、お客様がTIないしその代表者にその全額の補償をするものとします。

TI製品は、軍事的用途もしくは宇宙航空アプリケーションないし軍事的環境、航空宇宙環境にて使用されるようには設計もされていませんし、使用されることを意図されておられません。但し、当該TI製品が、軍需対応グレード品、若しくは「強化プラスチック」製品としてTIが特別に指定した製品である場合は除きます。TIが軍需対応グレード品として指定した製品のみが軍需品の仕様書に合致いたします。お客様は、TIが軍需対応グレード品として指定していない製品を、軍事的用途もしくは軍事的環境下で使用することは、もっぱらお客様の危険負担においてなされるということ、及び、お客様がもっぱら責任をもって、そのような使用に関して必要とされる全ての法的要求事項及び規制上の要求事項を満足させなければならないことを認め、かつ同意します。

TI製品は、自動車用アプリケーションないし自動車の環境において使用されるようには設計されていませんし、また使用されることを意図されておられません。但し、TIがISO/TS 16949の要求事項を満たしていると特別に指定したTI製品は除きます。お客様は、お客様が当該TI指定品以外のTI製品を自動車用アプリケーションに使用しても、TIは当該要求事項を満たしていなかったことについて、いかなる責任も負わないことを認め、かつ同意します。

Copyright © 2007, Texas Instruments Incorporated  
日本語版 日本テキサス・インスツルメンツ株式会社

## 弊社半導体製品の取り扱い・保管について

半導体製品は、取り扱い、保管・輸送環境、基板実装条件によっては、お客様での実装前後に破壊/劣化、または故障を起こすことがあります。

弊社半導体製品のお取り扱い、ご使用にあたっては下記の点を遵守して下さい。

### 1. 静電気

- 素手で半導体製品単体を触らないこと。どうしても触る必要がある場合は、リストストラップ等で人体からアースをとり、導電性手袋等をして取り扱うこと。
- 弊社出荷梱包単位（外装から取り出された内装及び個装）又は製品単品で取り扱いを行う場合は、接地された導電性のテーブル上で（導電性マットにアースをとったもの等）、アースをした作業者が行うこと。また、コンテナ等も、導電性のものを使うこと。
- マウンタやはんだ付け設備等、半導体の実装に関わる全ての装置類は、静電気の帯電を防止する措置を施すこと。
- 前記のリストストラップ・導電性手袋・テーブル表面及び実装装置類の接地等の静電気帯電防止措置は、常に管理されその機能が確認されていること。

### 2. 温・湿度環境

- 温度：0～40℃、相対湿度：40～85%で保管・輸送及び取り扱いを行うこと。（但し、結露しないこと。）

- 直射日光があたる状態で保管・輸送しないこと。
3. 防湿梱包
    - 防湿梱包品は、開封後は個別推奨保管環境及び期間に従い基板実装すること。
  4. 機械的衝撃
    - 梱包品（外装、内装、個装）及び製品単品を落下させたり、衝撃を与えないこと。
  5. 熱衝撃
    - はんだ付け時は、最低限260℃以上の高温状態に、10秒以上さらさないこと。（個別推奨条件がある時はそれに従うこと。）
  6. 汚染
    - はんだ付け性を損なう、又はアルミ配線腐食の原因となるような汚染物質（硫黄、塩素等ハロゲン）のある環境で保管・輸送しないこと。
    - はんだ付け後は十分にフラックスの洗浄を行うこと。（不純物含有率が一定以下に保証された無洗浄タイプのフラックスは除く。）

以上

以下余白

# 目次

	最初にお読みください.....	13
<b>1</b>	<b>概要.....</b>	<b>17</b>
1.1	概要.....	18
1.2	サブモジュールの概要.....	18
1.3	レジスタ・マッピング.....	21
<b>2</b>	<b>ePWM サブモジュール.....</b>	<b>23</b>
2.1	概要.....	24
2.2	タイム・ベース (TB) サブモジュール.....	27
2.2.1	タイム・ベース・サブモジュールの目的.....	27
2.2.2	タイム・ベース・サブモジュールの制御と監視.....	28
2.2.3	PWM 周期および周波数の計算.....	29
2.2.3.1	タイム・ベース周期シャドウ・レジスタ.....	30
2.2.3.2	タイム・ベース・カウンタ同期.....	31
2.2.4	複数 ePWM モジュールのタイム・ベース・クロックの位相ロック.....	33
2.2.5	タイム・ベース・カウンタ・モードとタイミング波形.....	33
2.3	カウンタ・コンペア (CC) サブモジュール.....	36
2.3.1	カウンタ・コンペア・サブモジュールの目的.....	36
2.3.2	カウンタ・コンペア・サブモジュールの制御と監視.....	36
2.3.3	カウンタ・コンペア・サブモジュールの動作上の重要点.....	37
2.3.4	カウント・モード・タイミング波形.....	38
2.4	アクション選択 (AQ) サブモジュール.....	41
2.4.1	アクション選択サブモジュールの目的.....	41
2.4.2	アクション選択サブモジュールの制御およびステータス・レジスタ定義.....	41
2.4.3	アクション選択イベント優先順位.....	44
2.4.4	一般的な設定に対する波形.....	45
2.5	デッドバンド・ジェネレータ (DB) サブモジュール.....	53
2.5.1	デッドバンド・サブモジュールの目的.....	53
2.5.2	デッドバンド・サブモジュールの制御と監視.....	53
2.5.3	デッドバンド・サブモジュールの動作上の重要点.....	54
2.6	PWM チョップパ (PC) サブモジュール.....	58
2.6.1	PWM チョップパ・サブモジュールの目的.....	58
2.6.2	PWM チョップパ・サブモジュールの制御.....	58
2.6.3	PWM チョップパ・サブモジュールの動作上の重要点.....	58
2.6.4	波形.....	59
2.6.4.1	ワンショット・パルス.....	60
2.6.4.2	デューティ・サイクル制御.....	61
2.7	トリップ・ゾーン (TZ) サブモジュール.....	62
2.7.1	トリップ・ゾーン・サブモジュールの目的.....	62
2.7.2	トリップ・ゾーン・サブモジュールの制御と監視.....	63
2.7.3	トリップ・ゾーン・サブモジュールの動作上の重要点.....	63

---

2.7.4	トリップ・イベント割り込みの生成 .....	65
2.8	イベント・トリガ (ET) サブモジュール .....	66
2.8.1	イベント・トリガ・サブモジュールの動作概要 .....	67
<b>3</b>	<b>パワー・トポロジへの応用.....</b>	<b>71</b>
3.1	複数モジュールの概要.....	72
3.2	重要な設定機能 .....	72
3.3	独立した周波数による複数のバック・コンバータの制御.....	73
3.4	同じ周波数による複数のバック・コンバータの制御.....	77
3.5	複数のハーフHブリッジ (HHB) コンバータの制御.....	80
3.6	モータ (ACI および PMSM) 用デュアル3相インバータの制御.....	82
3.7	PWM モジュール間の位相制御を使用した実際の応用 .....	86
3.8	3相インターリーブ型 DC/DC コンバータの制御 .....	87
3.9	ZVSFB (Zero Voltage Switched Full Bridge : ゼロ電圧スイッチング・フル・ブリッジ) コンバータの制御.....	91
<b>4</b>	<b>レジスタ.....</b>	<b>95</b>
4.1	タイム・ベース・サブモジュール・レジスタ.....	96
4.2	カウンタ・コンペア・サブモジュール・レジスタ .....	99
4.3	アクション選択サブモジュール・レジスタ .....	102
4.4	デッドバンド・サブモジュール・レジスタ .....	106
4.5	PWM チョップ・サブモジュール制御レジスタ .....	108
4.6	トリップ・ゾーン・サブモジュール制御およびステータス・レジスタ .....	109
4.7	イベント・トリガ・サブモジュール・レジスタ .....	114
4.8	正しい割り込み初期化手順 .....	118

## 図目次

1-1	複数の ePWM モジュール	19
1-2	PWM モジュールのサブモジュールと信号	20
1-3	PWM サブモジュールと重要な内部信号相互接続	21
2-1	タイム・ベース・サブモジュールのブロック図	27
2-2	タイム・ベース・サブモジュールの信号とレジスタ	28
2-3	タイム・ベース周波数および周期	30
2-4	タイム・ベース・カウンタ同期構造 1	31
2-5	タイム・ベース・カウンタ同期構造 2	32
2-6	タイム・ベース・アップ・カウント・モード波形	34
2-7	タイム・ベース・ダウン・カウント・モード波形	34
2-8	タイム・ベース・アップ・ダウン・カウント波形、TBCTL[PHSDIR=0]、同期イベントで カウント・ダウン	35
2-9	タイム・ベース・アップ・ダウン・カウント波形、TBCTL[PHSDIR=1]、同期イベントで カウント・アップ	35
2-10	カウンタ・コンペア・サブモジュール	36
2-11	カウンタ・コンペア・サブモジュールの詳細図	37
2-12	アップ・カウント・モードのカウンタ・コンペア・イベント波形	39
2-13	ダウン・カウント・モードのカウンタ・コンペア・イベント	39
2-14	アップ・ダウン・カウント・モードのカウンタ・コンペア・イベント、TBCTL[PHSDIR=0]、 同期イベントでカウント・ダウン	40
2-15	アップ・ダウン・カウント・モードのカウンタ・コンペア・イベント、TBCTL[PHSDIR=1]、 同期イベントでカウント・アップ	40
2-16	アクション選択サブモジュール	41
2-17	アクション選択サブモジュールの入力と出力	42
2-18	EPWMxA および EPWMxB 出力に対する可能なアクション選択のアクション	43
2-19	アップ・ダウン・カウント・モード対称波形	46
2-20	アップ、シングル・エッジ非対称波形 (EPWMxA および EPWMxB での独立した変調 - アクティブ・ハイ)	47
2-21	アップ、シングル・エッジ非対称波形 (EPWMxA および EPWMxB での独立した変調 - アクティブ・ロー)	48
2-22	アップ・カウント、パルス配置非対称波形 (EPWMxA での独立した変調)	49
2-23	アップ・ダウン・カウント、デュアル・エッジ対称波形 (EPWMxA および EPWMxB での 独立した変調 - アクティブ・ロー)	50
2-24	アップ・ダウン・カウント、デュアル・エッジ対称波形 (EPWMxA および EPWMxB での 独立した変調 - 相補)	51
2-25	アップ・ダウン・カウント、デュアル・エッジ非対称波形 (EPWMxA での独立した変調 - アクティブ・ロー)	52
2-26	デッドバンド・サブモジュール	53
2-27	デッドバンド・サブモジュールの設定オプション	54
2-28	一般的なケースのデッドバンド波形 (0% < デューティ < 100%)	56
2-29	PWM チョップパ・サブモジュール	58
2-30	PWM チョップパ・サブモジュールの動作詳細	59
2-31	チョッピング・アクションだけを示している単純な PWM チョップパ・サブモジュール波形	59
2-32	最初のパルスとその後続くパルスを示している PWM チョップパ・サブモジュール波形	60
2-33	後続パルスのパルス幅 (デューティ・サイクル) 制御を示している PWM チョップパ・サブモジュール波形	61

2-34	トリップ・ゾーン・サブモジュール.....	62
2-35	トリップ・ゾーン・サブモジュール・モード制御ロジック.....	65
2-36	トリップ・ゾーン・サブモジュール割り込みロジック.....	66
2-37	イベント・トリガ・サブモジュール.....	66
2-38	ADC 変換開始および割り込み信号のイベント・トリガ・サブモジュール相互接続性.....	67
2-39	イベント入力とプリスケール出力を示しているイベント・トリガ・サブモジュール.....	68
2-40	イベント・トリガ割り込みジェネレータ.....	69
2-41	イベント・トリガ SOCA パルス・ジェネレータ.....	70
2-42	イベント・トリガ SOCB パルス・ジェネレータ.....	70
3-1	簡略化された ePWM モジュール.....	72
3-2	一般的なマスタとして設定された EPWM1、スレーブとして設定された EPWM2.....	73
3-3	4 つのバック・ステージの制御 ( $F_{PWM1} \neq F_{PWM2} \neq F_{PWM3} \neq F_{PWM4}$ ).....	74
3-4	図 3-3 のバック波形 (注: ここでは、3 つのバックだけを示しています).....	75
3-5	4 つのバック・ステージの制御 (注: $F_{PWM2} = N \times F_{PWM1}$ ).....	77
3-6	図 3-5 のバック波形 (注: $F_{PWM2} = F_{PWM1}$ ).....	78
3-7	2 つのハーフ・ブリッジ・ステージの制御 ( $F_{PWM2} = N \times F_{PWM1}$ ).....	80
3-8	図 3-7 のハーフ H ブリッジ波形 (注: $F_{PWM2} = F_{PWM1}$ ).....	81
3-9	モータ制御でよく用いられるデュアル 3 相インバータ・ステージの制御.....	83
3-10	図 3-9 の 3 相インバータ波形 (1 つのインバータだけを示す).....	84
3-11	フェーズ制御用の 2 つの PWM モジュールの設定.....	86
3-12	2 つのモジュール間の位相制御に関連するタイミング波形.....	87
3-13	3 相インターリーブ型 DC/DC コンバータの制御.....	88
3-14	図 3-13 の 3 相インターリーブ型 DC/DC コンバータ波形.....	89
3-15	フル H ブリッジ・ステージの制御 ( $F_{PWM2} = F_{PWM1}$ ).....	91
3-16	ZVS フル H ブリッジ波形.....	92
4-1	タイム・ベース周期レジスタ (TBPRD).....	96
4-2	タイム・ベース位相レジスタ (TBPHS).....	96
4-3	タイム・ベース・カウンタ・レジスタ (TBCTR).....	96
4-4	タイム・ベース制御レジスタ (TBCTL) 15141312111098.....	97
4-5	タイム・ベース・ステータス・レジスタ (TBSTS).....	99
4-6	カウンタ・コンペア A レジスタ (CMPA) のフィールドの説明.....	99
4-7	カウンタ・コンペア B レジスタ (CMPB).....	100
4-8	カウンタ・コンペア制御レジスタ (CMPCTL).....	101
4-9	アクション選択出力 A 制御レジスタ (AQCTLA).....	102
4-10	アクション選択出力 B 制御レジスタ (AQCTLB).....	103
4-11	アクション選択ソフトウェア強制レジスタ (AQSFRC).....	104
4-12	アクション選択連続ソフトウェア強制レジスタ (AQCSFRC).....	105
4-13	デッドバンド・ジェネレータ制御レジスタ (DBCTL).....	106
4-14	デッドバンド・ジェネレータ立ち上がりエッジ遅延レジスタ (DBRED).....	107
4-15	デッドバンド・ジェネレータ立ち下がりエッジ遅延レジスタ (DBFED).....	108
4-16	PWM チョップ制御レジスタ (PCCTL).....	108
4-17	トリップ・ゾーン・セレクト・レジスタ (TZSEL).....	109
4-18	トリップ・ゾーン制御レジスタ (TZCTL).....	111
4-19	トリップ・ゾーン・イネーブル割り込みレジスタ (TZEINT).....	111
4-20	トリップ・ゾーン・フラグ・レジスタ (TZFLG).....	112

---

4-21	トリップ・ゾーン・クリア・レジスタ (TZCLR) .....	113
4-22	トリップ・ゾーン強制レジスタ (TZFRC) .....	113
4-23	イベント・トリガ選択レジスタ (ETSEL) .....	114
4-24	イベント・トリガ・プリスケール・レジスタ (ETPS).....	115
4-25	イベント・トリガ・フラグ・レジスタ (ETFLG) .....	117
4-26	イベント・トリガ・クリア・レジスタ (ETCLR) .....	117
4-27	イベント・トリガ強制レジスタ (ETFRC) .....	118

## 表目次

1-1	サブモジュール別に分類された ePWM モジュール制御およびステータス・レジスタ・セット	22
2-1	サブモジュール設定パラメータ	24
2-2	タイム・ベース・サブモジュールのレジスタ	28
2-3	重要なタイム・ベース信号	29
2-4	カウンタ・コンペア・サブモジュールのレジスタ	36
2-5	カウンタ・コンペア・サブモジュールの重要な信号	37
2-6	アクション選択サブモジュールのレジスタ	41
2-7	アクション選択サブモジュールの可能な入カイベント	42
2-8	アップ・ダウン・カウント・モードのアクション選択イベント優先順位	44
2-9	アップ・カウント・モードのアクション選択イベント優先順位	44
2-10	ダウン・カウント・モードのアクション選択イベント優先順位	44
2-11	CMPA/CMPB が周期より大きい場合の動作	45
2-12	デッドバンド・ジェネレータ・サブモジュールのレジスタ	53
2-13	従来のデッドバンド動作モード	55
2-14	DBFED および DBRED の機能としてのデッドバンド遅延値 ( $\mu\text{S}$ 単位)	57
2-15	PWM チョップ・サブモジュールのレジスタ	58
2-16	SYSCLOCKOUT=100 MHz の場合の可能なパルス幅値	60
2-17	トリップ・ゾーン・サブモジュールのレジスタ	63
2-18	トリップ・イベントでの可能なアクション	64
2-19	イベント・トリガ・サブモジュールのレジスタ	68
4-1	タイム・ベース周期レジスタ (TBPRD) のフィールドの説明	96
4-2	タイム・ベース位相レジスタ (TBPHS) のフィールドの説明	96
4-3	タイム・ベース・カウンタ・レジスタ (TBCTR) のフィールドの説明	96
4-4	タイム・ベース制御レジスタ (TBCTL) のフィールドの説明	97
4-5	タイム・ベース・ステータス・レジスタ (TBSTS) のフィールドの説明	99
4-6	カウンタ・コンペア A レジスタ (CMPA) のフィールドの説明	100
4-7	カウンタ・コンペア B レジスタ (CMPB) のフィールドの説明	101
4-8	カウンタ・コンペア制御レジスタ (CMPCTL) のフィールドの説明	101
4-9	アクション選択出力 A 制御レジスタ (AQCTLA) のフィールドの説明	102
4-10	アクション選択出力 B 制御レジスタ (AQCTLB) のフィールドの説明	103
4-11	アクション選択ソフトウェア強制レジスタ (AQSFRC) のフィールドの説明	105
4-12	アクション選択連続ソフトウェア強制レジスタ (AQCSFRC) のフィールドの説明	106
4-13	デッドバンド・ジェネレータ制御レジスタ (DBCTL) のフィールドの説明	106
4-14	デッドバンド・ジェネレータ立ち上がりエッジ遅延レジスタ (DBRED) のフィールドの説明	108
4-15	デッドバンド・ジェネレータ立ち下がりエッジ遅延レジスタ (DBFED) のフィールドの説明	108
4-16	PWM チョップ制御レジスタ (PCCTL) のビットの説明	108
4-17	トリップ・ゾーン・サブモジュール・セレクト・レジスタ (TZSEL) のフィールドの説明	110
4-18	トリップ・ゾーン制御レジスタ (TZCTL) のフィールドの説明	111
4-19	トリップ・ゾーン・イネーブル割り込みレジスタ (TZEINT) のフィールドの説明	111
4-20	トリップ・ゾーン・フラグ・レジスタ (TZFLG) のフィールドの説明	112
4-21	トリップ・ゾーン・クリア・レジスタ (TZCLR) のフィールドの説明	113
4-22	トリップ・ゾーン強制レジスタ (TZFRC) のフィールドの説明	113
4-23	イベント・トリガ選択レジスタ (ETSEL) のフィールドの説明	114
4-24	イベント・トリガ・プリスケール・レジスタ (ETPS) のフィールドの説明	115

---

4-25	イベント・トリガ・フラグ・レジスタ (ETFLG) のフィールドの説明 .....	117
4-26	イベント・トリガ・クリア・レジスタ (ETCLR) のフィールドの説明 .....	117
4-27	イベント・トリガ強制レジスタ (ETFRC) のフィールドの説明 .....	118



## 最初にお読みください

本書では、エンハンスド PWM (ePWM) モジュールについて説明します。説明する内容には、モジュールの概要と以下の各サブモジュールに関する情報が含まれます。

- タイム・ベース・モジュール
- カウンタ・コンペア・モジュール
- アクション選択モジュール
- デッドバンド・ジェネレータ・モジュール
- PWM チョップパ (PC) モジュール
- トリップ・ゾーン・モジュール
- イベント・トリガ・モジュール

### 関連資料

以下の資料では、TI Web サイトで利用できる TMS320x280x および関連サポート・ツールについて説明しています。

#### データ・マニュアル

**SPRS230:** — [『TMS320F2809, F2808, F2806, F2802, F2801, F2801x UCD9501, C2802, C2801 DSPs Data Manual』](#)には、ピンアウト、信号表記、F280x デバイスの電氣的仕様およびタイミング仕様が含まれています。

**SPRS357:** — [『TMS320F28044 Digital Signal Processor Data Manual』](#)には、ピンアウト、信号表記、F28044 デバイスの電氣的仕様およびタイミング仕様が含まれています。

#### ユーザーズ・ガイド

**SPRU051:** — [『TMS320x28xx, 28xxx Serial Communication Interface \(SCI\) Reference Guide』](#)では、UART として一般的に知られる SCI (2 線式非同期シリアル・ポート) について説明しています。SCI モジュールは、標準 NRZ (non-return-to-zero) フォーマットを使用する CPU と他の非同期ペリフェラルの間のデジタル通信をサポートします。

**SPRU059:** — [『TMS320x28xx, 28xxx Serial Peripheral Interface \(SPI\) Reference Guide』](#)では、SPI (高速同期シリアル入力/出力 (I/O) ポート) について説明しています。このポートにより、プログラムされた長さ (1 ~ 16 ビット) のシリアル・ビット・ストリームを、プログラムされたビット転送レートで送受信することができます。

**SPRU074:** — [『TMS320x28xx, 28xxx Enhanced Controller Area Network \(eCAN\) Reference Guide』](#)では、確立されたプロトコルを使用して電氣的にノイズの多い環境で他のコントローラと逐次的に通信する eCAN について説明しています。

**SPRU430:** — [『TMS320C28x DSP CPU and Instruction Set Reference Guide』](#)では、中央演算処理ユニット (CPU) と、TMS320C28x 固定小数点デジタル・シグナル・プロセッサ (DSP) の命令セットについて説明しています。また、これらの DSP で利用できるエミュレーション機能についても説明します。

- SPRU513:** — [『TMS320C28x Assembly Language Tools User's Guide』](#)では、TMS320C28x デバイス用のアセンブリ言語ツール（アセンブリ言語コードの開発に使用するアセンブラや他のツール）、アセンブラ擬似命令、マクロ、共通オブジェクト・ファイル・フォーマット、およびシンボリック・デバッグの擬似命令について説明しています。
- SPRU514:** — [『TMS320C28x Optimizing C Compiler User's Guide』](#)では、TMS320C28x™ C/C++ コンパイラについて説明しています。このコンパイラは、ANSI 標準 C/C++ ソース・コードを受け、TMS320C28x デバイス用の TMS320 DSP アセンブリ言語ソース・コードを生成します。
- SPRU566:** — [『TMS320x28xx, 28xxx Peripheral Reference Guide』](#)には、28x デジタル・シグナル・プロセッサ（DSP）のペリフェラル・リファレンス・ガイドが記載されています。
- SPRU608:** — [『TMS320C28x Instruction Set Simulator Technical Overview』](#)では、C28x™ コアの命令セットをシミュレートするシミュレータ（TMS320C2000 IDE 用の Code Composer Studio 内で利用可能）について説明しています。
- SPRU625:** — [『TMS320C28x DSP/BIOS Application Programming Interface \(API\) Reference Guide』](#)では、DSP/BIOS を使用した開発について説明します。
- SPRU712:** — [『TMS320x28xx, 28xxx System Control and Interrupts Reference Guide』](#)では、280x デジタル・シグナル・プロセッサ（DSP）のさまざまな割り込みとシステム制御機能について説明します。
- SPRU716:** — [『TMS320x280x, 2801x, 2804x Analog-to-Digital Converter \(ADC\) Reference Guide』](#)では、12 ビット・パイプライン ADC であるオンチップ ADC モジュールの設定方法と使用方法について説明しています。
- SPRU721:** — [『TMS320x28xx, 28xxx Inter-Integrated Circuit \(I2C\) Reference Guide』](#)では、TMS320x280x デジタル・シグナル・プロセッサ（DSP）で利用できる Inter-Integrated Circuit (I<sup>2</sup>C) モジュールの機能と動作について説明しています。
- SPRU722:** — [『TMS320x280x, 2801x, 2804x Boot ROM Reference Guide』](#)では、ブートローダ（工場プログラムされたブート・ローディング・ソフトウェア）の用途と機能について説明しています。また、デバイス・オンチップ・ブート ROM の他の内容についても説明し、そのメモリ内のどこに情報のすべてが配置されているかを示します。
- SPRU790:** — [『TMS320x28xx, 28xxx Enhanced Quadrature Encoder Pulse \(eQEP\) Reference Guide』](#)では、eQEP モジュールについて説明しています。高性能のモーションおよび位置制御システムでは、このモジュールをリニアまたはロータリ・インクリメンタル・エンコーダとのインターフェイスに使用して、位置、方向、および速度情報をエンコーダから取得します。この資料には、モジュールとレジスタの説明が含まれます。
- SPRU791:** — [『TMS320x28xx, 28xxx Enhanced Pulse Width Modulator \(ePWM\) Module Reference Guide』](#)では、デジタル・モータ制御、スイッチ・モード電源制御、UPS（無停電電源装置）、他の形式の電源変換など、エンハンスト PWM モジュールについて説明しています。
- SPRU807:** — [『TMS320x28xx, 28xxx Enhanced Capture \(eCAP\) Module Reference Guide』](#)では、エンハンスト・キャプチャ・モジュールについて説明しています。この資料には、モジュールとレジスタの説明が含まれます。
- SPRU924:** — [『TMS320x28xx, 28xxx High-Resolution Pulse Width Modulator \(HRPWM\)』](#)では、パルス幅モジュレータに対する高分解能拡張（HRPWM）の動作について説明しています。
- アプリケーション・レポート**
- SPRAA58:** — [『TMS320x281x to TMS320x280x Migration Overview』](#)では、テキサス・インスツルメンツの TMS320x281x DSP と TMS320x280x DSP の違いを説明して、281x から 280x へのアプリケーション移行を支援します。本書の主な焦点は 281x から 280x への移行ですが、反対の移行（280x から 281x へ）を検討しているユーザーにも役立ちます。

- SPRA550:** — [『3.3 V DSP for Digital Motor Control』](#) では、3.3V 専用モータ・コントローラのシナリオについて説明し、ほとんどのアプリケーションについて 3.3V ~ 5V 間のインターフェイスの重要な問題が存在しないことを示します。オンチップ 3.3V アナログ・デジタル・コンバータ (ADC) と 5V ADC についても説明します。システム・ノイズと EMI 効果を低減できるコンポーネント・レイアウトとプリント回路基板 (PCB) 設計のガイドラインが要約されています。
- SPRA820:** — [『Online Stack Overflow Detection on the TMS320C28x DSP』](#) では、TMS320C28x™ DSP でのオンライン・スタック・オーバーフロー検出の方法論を示します。DSP/BIOS™ アプリケーションと非DSP/BIOSアプリケーションの両方でオーバーフロー検出を実装するための関数が含まれている C ソース・コードが提供されます。
- SPRA861:** — [『RAMDISK: A Sample User-Defined C I/O Driver』](#) では、任意のデバイスでハイレベルな CIO 機能の高度なバッファリングを使用するための簡単な方法を提供します。このアプリケーション・レポートは、ユーザー定義デバイス・ドライバのサンプル実装を示します。
- SPRA953:** — [『IC Package Thermal Metrics』](#) では、従来と新規のサーマル・メトリクスについて説明し、システム・レベルのジャンクション温度推定を各種アプリケーションに関連づけて取り上げます。
- SPRA963:** — [『Reliability Data for TMS320LF24x and TMS320F281x Devices』](#) では、TMS320LF24x および TMS320F281x デバイスの信頼性データについて説明しています。
- SPRA991:** — [『Simulation Fulfills its Promise for Enhancing Debug and Analysis - A White Paper』](#) では、より効果的に代替システムを評価できるようにすることによって開発者の開発サイクルの短縮を可能にする拡張されたシミュレーションについて説明しています。

## 商標

TMS320C28x、C28x は、テキサス・インスツルメンツの商標です。



## 概要

エンハンスド PWM (ePWM) ペリフェラルは、民生 / 産業装置における電力関連システムの多くを制御する際のキーとなる要素です。これらのシステムには、デジタル・モータ制御、スイッチ・モード電源制御、無停電電源装置 (UPS)、その他電力変換が含まれます。ePWM ペリフェラルは、デジタル・アナログ・コンバータ (DAC) 機能としても使用できます。この用途では、デューティ・サイクルは DAC アナログ値に相当します。この機能は Power DAC と呼ばれることもあります。

このリファレンス・ガイドは、TMS320x280x、TMS320x2801x、および TMS320x2804x プロセッサに搭載されている ePWM に適用できます。これには、フラッシュベース、ROM ベース、および RAM ベースのデバイスがすべて含まれます。

項目	ページ
1.1 概要 .....	18
1.2 サブモジュールの概要 .....	18
1.3 レジスタ・マッピング .....	21

## 1.1 概要

効果的な PWM ペリフェラルは、最小限の CPU オーバーヘッドで複雑なパルス幅波形を生成できなければなりません。このペリフェラルは、高度にプログラマブルかつ非常に柔軟でありながら、容易に理解して使用できる必要があります。ここで説明する ePWM ユニットの、必要なすべてのタイミングおよび制御リソースを各 PWM チャンネルごとに持つことで、これらの要求に対応しております。リソースの共有は回避されています。代わりに、ePWM は、比較的小さな独立したリソースを備えたシングル・チャンネル・モジュールで構成されています。これらのモジュールは、必要に応じて協調動作してシステムを形成できます。このモジュール化手法により、ペリフェラル構造も分かりやすく、ユーザーはすぐに動作を理解できるようになります。

本書では、デバイス上の 1 つの ePWM モジュール単体を示すために、信号名またはモジュール名の中で文字  $x$  を使用しています。たとえば、出力信号の EPWMxA と EPWMxB は、ePWMx からの出力信号を表します。したがって、EPWM1A と EPWM1B は ePWM1 に属し、同様に EPWM4A と EPWM4B は ePWM4 に属します。

## 1.2 サブモジュールの概要

ePWM モジュールは、2 つの PWM 出力 (EPWMxA と EPWMxB) で構成された 1 つの完全な PWM チャンネルを表します。図 1-1 に示すように、デバイス内では複数の同じ機能を持った ePWM モジュールがあります。各 ePWM モジュールは、1 つの機能を除いて同一です。より正確に PWM 出力を制御できるようにするハードウェア拡張が含まれているモジュールもあります。この拡張は高分解能パルス幅モジュレータ (HRPWM) であり、『TMS320x28xx, 28xxx High-Resolution Pulse Width Modulator (HRPWM) Reference Guide』(SPRU924) に記載されています。どの ePWM モジュールにこの機能が含まれているのかを確認するには、各デバイスのデータ・マニュアルを参照してください。各 ePWM モジュールは、1 から始まる数値によって示されます。たとえば、ePWM1 はシステムにおける最初のモジュール、ePWM3 は 3 番目のモジュールであり、ePWMx は任意のモジュールを表します。

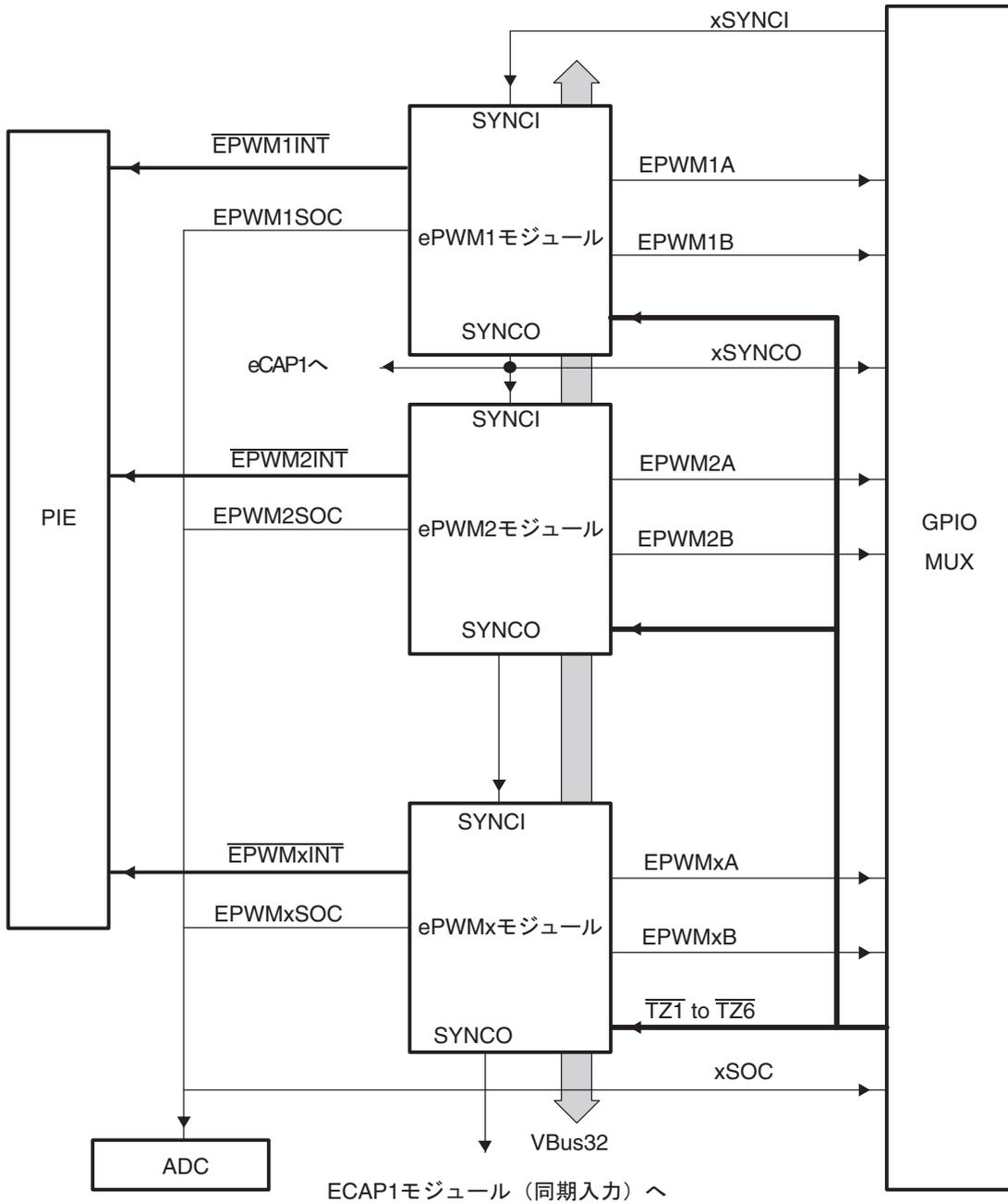
各 ePWM モジュールは、クロック同期信号によって連結されます。この同期機能により、これらのモジュールは必要に応じて単一のシステムとして動作できます。また、この同期機能はキャプチャ・ペリフェラル・モジュール (eCAP) に拡張することもできます。モジュールの数はデバイスに依存し、ターゲット・アプリケーションのニーズに基づきます。各モジュールは単独で動作することもできます。

各 ePWM モジュールでは、以下の機能がサポートされます。

- 周期および周波数を制御できる独立した 16 ビット・タイマ・カウンタ
- 以下の設定で使用できる 2 つの PWM 出力 (EPWMxA と EPWMxB)
  - シングルエッジ動作の 2 つの独立した PWM 出力
  - デュアルエッジ対称動作の 2 つの独立した PWM 出力
  - デュアルエッジ非対称動作の 1 つの独立した PWM 出力
- ソフトウェアによる PWM 信号の非同期オーバーライド制御
- 他の ePWM モジュールに対するプログラマブル進みまたは遅れ位相制御フェーズ制御サポート
- 各サイクルごとの位相ハードウェアロック (同期)
- 立ち上がりエッジおよび立ち下がりエッジ遅延を独立して制御可能なデッドバンド生成
- フェール時におけるサイクル・バイ・サイクル・トリップとワンショット・トリップ
- トリップ状態を、PWM 出力に対して High、Low、またはハイ・インピーダンス状態に設定可能
- すべてのイベントで、CPU 割り込みと ADC 変換開始 (SOC) の両方をトリガ可能
- プログラマブル・イベント・プリスケールリングにより、割り込みでの CPU オーバーヘッドが最小限に抑えられる
- 高周波キャリア信号による PWM チョッピング (パルス・トランス・ゲート・ドライブに役立つ)

各 ePWM モジュールは、図 1-1 に示した入力 / 出力信号に接続されます。信号の詳細については、後続のセクションで説明します。

図 1-1 複数の ePWM モジュール



ePWM モジュールの接続順序は、図 1-1 とは異なる場合があります。各デバイスの同期構成については、[セクション 2.2.3.2](#) を参照してください。各 ePWM モジュールは、7 つのサブモジュールで構成され、[図 1-2](#) に示した信号によってシステム内で接続されています。

図 1-2 PWM モジュールのサブモジュールと信号

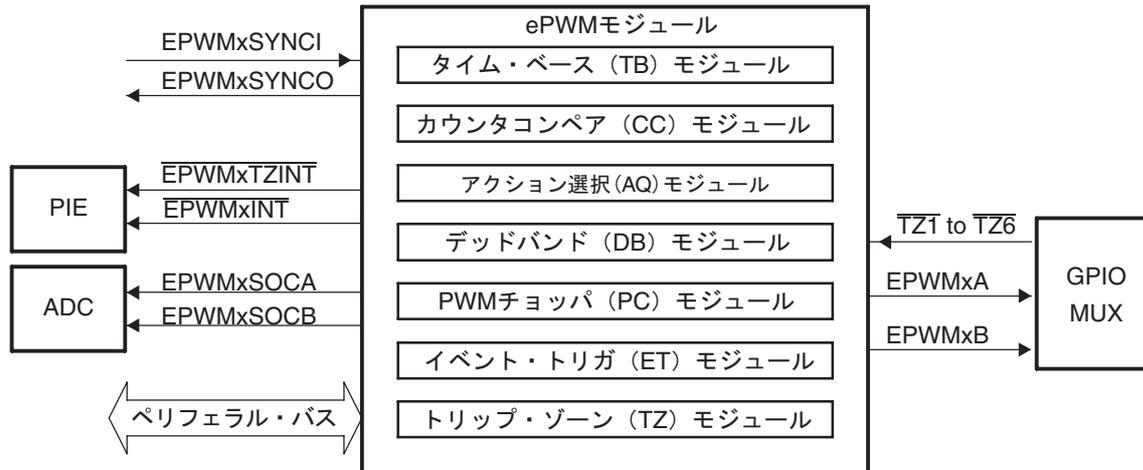


図 1-3 は、単一 ePWM モジュールの内部詳細を表します。ePWM モジュールによって使用されるメイン信号は以下のとおりです。

- **PWM 出力信号 (EPWMxA と EPWMxB)**

PWM 出力信号は、『TMS320x280x, 2801x, 2804x System Control and Interrupts Reference Guide』(SPRU712)に記載されている GPIO ペリフェラルを通じてデバイスの外部で利用できるようになります。

- **トリップ・ゾーン信号 (TZ1 ~ TZ6)**

これらの入力信号は、外部のフェール状態を ePWM モジュールに通知します。デバイスの各モジュールは、トリップ・ゾーン信号を使用または無視するように設定できます。トリップ・ゾーン信号は、GPIO ペリフェラルを通じた非同期入力として設定できます。

- **同期入力 (EPWMxSYNCl) および出力 (EPWMxSYNCO) 信号**

これらの同期信号によって、ePWM モジュールが継続接続されます。各モジュールは、同期入力を使用または無視するように設定できます。クロック同期入力および出力信号は、ePWM1 (ePWM モジュール #1) にのみに送られます。ePWM1 の同期出力 (EPWM1SYNCO) は、最初のエンハンスド・キャプチャ・モジュール (eCAP1) の SYNCl にも接続されます。セクション 2.2.3.2 と各デバイスのデータ・マニュアルを参照してください。

- **ADC 変換開始信号 (EPWMxSOCA と EPWMxSOCB)**

各 ePWM モジュールには、2 つの ADC 変換開始信号があります (シーケンサごとに 1 つ)。ePWM モジュールは、どちらのシーケンサについても変換開始をトリガできます。どのイベントが変換開始をトリガするのは、ePWM のイベント・トリガ・サブモジュールで設定されます。

- **ペリフェラル・バス**

ペリフェラル・バスは 32 ビット幅であり、ePWM レジスタ・ファイルに対する 16 ビット・ライトと 32 ビット・ライトの両方を可能にします。

図 1-3 PWM サブモジュールと重要な内部信号相互接続

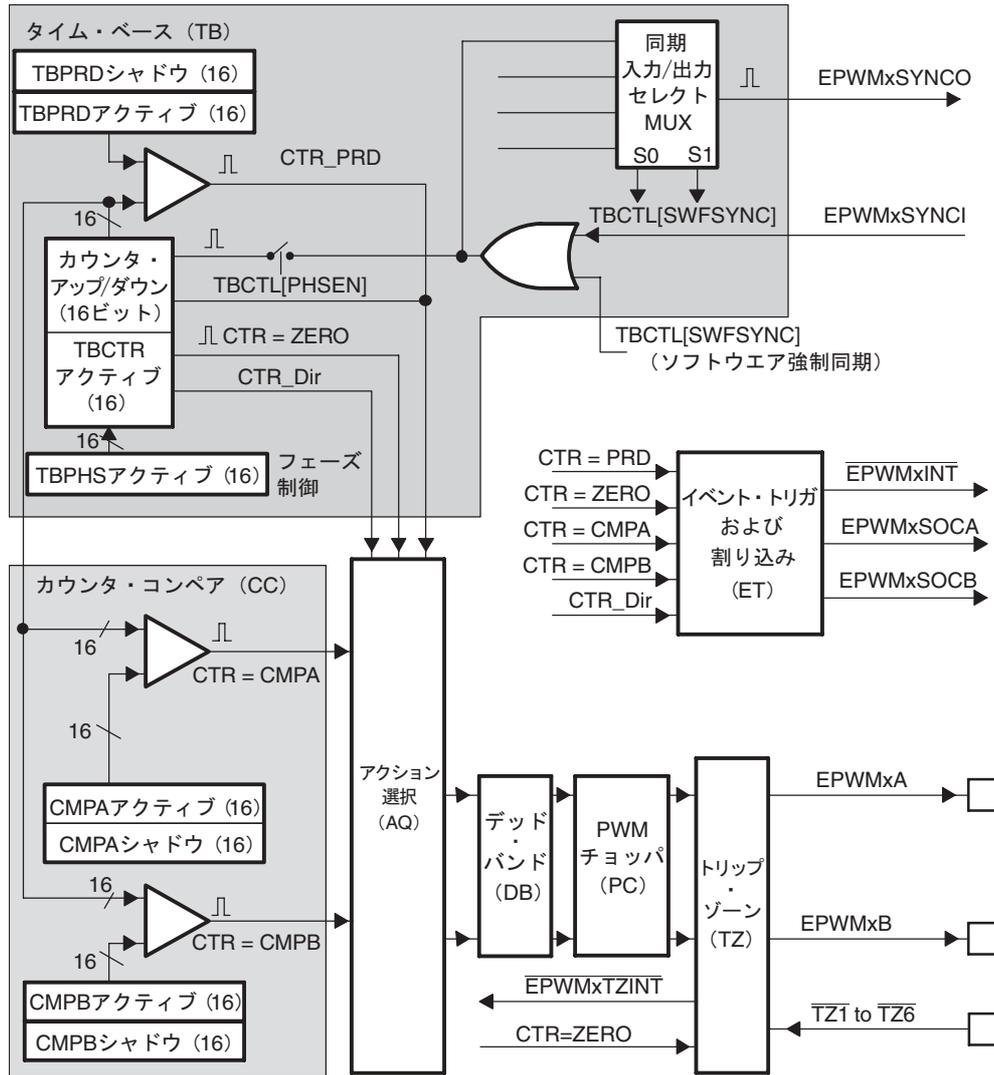


図 1-3 は重要な内部サブモジュール相互接続信号も示しています。各サブモジュールの詳細については、それぞれのセクションで説明します。

### 1.3 レジスタ・マッピング

ePWM モジュール制御およびステータス・レジスタ・セットは、表 1-1 に示すようにサブモジュール別  
に分類されます。各レジスタ・セットは、ePWM モジュールごとに同じセットが用意されています。  
デバイス上の各 ePWM レジスタの開始アドレスは、対応するデータ・マニュアルに記載されています。

表 1-1 サブモジュール別に分類された ePWM モジュール制御  
およびステータス・レジスタ・セット

名前	オフセット <sup>(1)</sup>	サイズ (x16)	シャドウ	説明
<b>タイム・ベース・サブモジュール・レジスタ</b>				
TBCTL	0x0000	1	無	タイム・ベース制御レジスタ
TBSTS	0x0001	1	無	タイム・ベース・ステータス・レジスタ
TBPHSHR	0x0002	1	無	HRPWM 位相レジスタの拡張 <sup>(2)</sup>
TBPHS	0x0003	1	無	タイム・ベース位相レジスタ
TBCTR	0x0004	1	無	タイム・ベース・カウンタ・レジスタ
TBPRD	0x0005	1	有	タイム・ベース周期レジスタ
<b>カウンタ・コンペア・サブモジュール・レジスタ</b>				
CMPCTL	0x0007	1	無	カウンタ・コンペア制御レジスタ
CMPAHR	0x0008	1	無	HRPWM カウンタ・コンペア A レジスタの拡張 <sup>(2)</sup>
CMPA	0x0009	1	有	カウンタ・コンペア A レジスタ
CMPB	0x000A	1	有	カウンタ・コンペア B レジスタ
<b>アクション選択サブモジュール・レジスタ</b>				
AQCTLA	0x000B	1	無	出力 A のアクション選択制御レジスタ (EPWMxA)
AQCTLB	0x000C	1	無	出力 B のアクション選択制御レジスタ (EPWMxB)
AQSFRC	0x000D	1	無	アクション選択ソフトウェア強制レジスタ
AQCSFRC	0x000E	1	有	アクション選択連続 S/W 強制レジスタ・セット
<b>デッドバンド・ジェネレータ・サブモジュール・レジスタ</b>				
DBCTL	0x000F	1	無	デッドバンド・ジェネレータ制御レジスタ
DBRED	0x0010	1	無	デッドバンド・ジェネレータ立ち上がりエッジ 遅延カウンタ・レジスタ
DBFED	0x0011	1	無	デッドバンド・ジェネレータ立ち下がりエッジ 遅延カウンタ・レジスタ
<b>トリップ・ゾーン・サブモジュール・レジスタ</b>				
TZSEL	0x0012	1	無	トリップ・ゾーン・セレクト・レジスタ
TZCTL	0x0014	1	無	トリップ・ゾーン制御レジスタ <sup>(3)</sup>
TZEINT	0x0015	1	無	トリップ・ゾーン・イネーブル割り込みレジスタ <sup>(3)</sup>
TZFLG	0x0016	1	無	トリップ・ゾーン・フラグ・レジスタ <sup>(3)</sup>
TZCLR	0x0017	1	無	トリップ・ゾーン・クリア・レジスタ <sup>(3)</sup>
TZFRC	0x0018	1	無	トリップ・ゾーン強制レジスタ <sup>(3)</sup>
<b>イベント・トリガ・サブモジュール・レジスタ</b>				
ETSEL	0x0019	1	無	イベント・トリガ選択レジスタ
ETPS	0x001A	1	無	イベント・トリガ・プリスケール・レジスタ
ETFLG	0x001B	1	無	イベント・トリガ・フラグ・レジスタ
ETCLR	0x001C	1	無	イベント・トリガ・クリア・レジスタ
ETFRC	0x001D	1	無	イベント・トリガ強制レジスタ
<b>PWM チョップ・サブモジュール・レジスタ</b>				
PCCTL	0x001E	1	無	PWM チョップ制御レジスタ
<b>高分解能パルス幅モジュレータ (HRPWM) 拡張レジスタ</b>				
HRCNFG	0x0020	1	無	HRPWM コンフィギュレーション・レジスタ <sup>(2) (3)</sup>

(1) 記載されていないロケーションは予約されています。

(2) これらのレジスタは、高分解能 PWM 拡張が含まれている ePWM モジュールでのみ利用できます。それ以外の場合は、これらのロケーションは予約されます。これらのレジスタの説明は、『TMS320x28xx, 28xxx High-Resolution Pulse Width Modulator (HRPWM) Reference Guide』(SPRU924)に記載されています。どのモジュールに HRPWM が含まれているのかを確認するには、各デバイスのデータ・マニュアルを参照してください。

(3) EALLOW 保護されているレジスタ。『TMS320x280x, 2801x, 2804x System Control and Interrupts Reference Guide』(SPRU712)で説明しています。

## ePWM サブモジュール

---

---

---

各 ePWM ペリフェラルには、7 つのサブモジュールが含まれています。これらのサブモジュールは、ソフトウェアで設定できる特定のタスクをそれぞれ実行します。

項目	ページ
2.1 概要 .....	24
2.2 タイム・ベース (TB) サブモジュール .....	27
2.3 カウンタ・コンペア (CC) サブモジュール .....	36
2.4 アクション選択 (AQ) サブモジュール .....	41
2.5 デッドバンド・ジェネレータ (DB) サブモジュール .....	53
2.6 PWM チョップパ (PC) サブモジュール .....	58
2.7 トリップ・ゾーン (TZ) サブモジュール .....	62
2.8 イベント・トリガ (ET) サブモジュール .....	66

## 2.1 概要

7つの重要なサブモジュールとそれぞれの主な設定パラメータのリストを表 2-1 に示します。たとえば、PWM 波形のデューティ・サイクルを調整または制御する必要がある場合は、関連する詳細について [セクション 2.3](#) のカウンタ・コンペア・サブモジュールを参照してください。

表 2-1 サブモジュール設定パラメータ

サブモジュール	設定パラメータまたはオプション
タイム・ベース (TB)	<ul style="list-style-type: none"> <li>• システム・クロック (SYSCLKOUT) をスケーリングし、タイム・ベース・クロック (TBCLK) を設定します。</li> <li>• PWM タイム・ベース・カウンタ (TBCTR) の周波数または周期を設定します。</li> <li>• タイム・ベース・カウンタのモードを設定します。 <ul style="list-style-type: none"> <li>- カウントアップ・モード：非対称 PWM の場合に使用される</li> <li>- カウントダウン・モード：非対称 PWM の場合に使用される</li> <li>- カウントアップ &amp; ダウン・モード：対称 PWM の場合に使用される</li> </ul> </li> <li>• 別の ePWM モジュールを基準とした位相を設定します。</li> <li>• ハードウェアまたはソフトウェアによってモジュール間でタイム・ベース・カウンタを同期させます。</li> <li>• 同期イベント後のタイム・ベース・カウンタの方向 (アップまたはダウン) を設定します。</li> <li>• エミュレータによってデバイスが Halt 状態になった場合にタイム・ベース・カウンタがどのように動作するのかを設定します。</li> <li>• ePWM モジュールの同期出力のソースを指定します。 <ul style="list-style-type: none"> <li>- 同期入力信号が入った時</li> <li>- タイム・ベース・カウンタはゼロに等しい時</li> <li>- タイム・ベース・カウンタはカウンタ・コンペア B (CMPB) に等しい時</li> <li>- 出力同期信号は生成しない</li> </ul> </li> </ul>
カウンタ・コンペア (CC)	<ul style="list-style-type: none"> <li>• 出力 EPWMxA または出力 EPWMxB (あるいはその両方) の PWM デューティ・サイクルを指定します。</li> <li>• EPWMxA または EPWMxB 出力でスイッチング・イベントが発生するタイミングを指定します。</li> </ul>
アクション選択 (AQ)	<ul style="list-style-type: none"> <li>• タイム・ベースまたはカウンタ・コンペア・サブモジュール・イベントの発生時にとられるアクションのタイプを指定します。 <ul style="list-style-type: none"> <li>- アクションなし</li> <li>- 出力 EPWMxA または EPWMxB (あるいはその両方) を High に切り替える</li> <li>- 出力 EPWMxA または EPWMxB (あるいはその両方) を Low に切り替える</li> <li>- 出力 EPWMxA または EPWMxB (あるいはその両方) をトグルする</li> </ul> </li> <li>• ソフトウェア制御による PWM 強制出力状態の設定。</li> <li>• ソフトウェアで PWM デッドバンドを設定および制御します。</li> </ul>
デッドバンド (DB)	<ul style="list-style-type: none"> <li>• 上位スイッチと下位スイッチの間における相補デッドバンド関係の制御。</li> <li>• 出力立ち上がりエッジ遅延値を指定します。</li> <li>• 出力立ち下がりエッジ遅延値を指定します。</li> <li>• デッドバンド・モジュールを完全にバイパスします。この場合、PWM 波形は変更なしで渡されます。</li> </ul>
PWM チョップパ (PC)	<ul style="list-style-type: none"> <li>• チョッピング (キャリア) 周波数を作成します。</li> <li>• チョッピングされたパルス列における最初のパルスのパルス幅。</li> <li>• 2 番目以降のパルスのデューティ・サイクル。</li> <li>• PWM チョップパ・モジュールを完全にバイパスします。この場合、PWM 波形は変更なしで渡されます。</li> </ul>
トリップ・ゾーン (TZ)	<ul style="list-style-type: none"> <li>• 1 つまたはすべてのトリップ・ゾーン・ピンに反応するように、あるいはどのトリップ・ゾーン・ピンにも反応しないように ePWM モジュールを設定します。</li> <li>• フェール発生時にとられるトリップ・アクションを指定します。 <ul style="list-style-type: none"> <li>- EPWMxA または EPWMxB (あるいはその両方) を High にする</li> <li>- EPWMxA または EPWMxB (あるいはその両方) を Low にする</li> <li>- EPWMxA または EPWMxB (あるいはその両方) をハイ・インピーダンス状態にする</li> <li>- トリップ状態を無視するように EPWMxA または EPWMxB (あるいはその両方) を設定する</li> </ul> </li> <li>• ePWM が各トリップ・ゾーン・ピンにどのように反応するかを設定します。 <ul style="list-style-type: none"> <li>- ワンショット</li> <li>- サイクル・バイ・サイクル</li> </ul> </li> <li>• トリップ・ゾーンをイネーブルにして割り込みを開始します。</li> <li>• トリップ・ゾーン・モジュールを完全にバイパスします。</li> </ul>

表 2-1 サブモジュール設定パラメータ (続き)

サブモジュール	設定パラメータまたはオプション
イベント・トリガ (ET)	<ul style="list-style-type: none"> <li>• 割り込みをトリガする ePWM イベントをイネーブルにします。</li> <li>• ADC 変換開始イベントをトリガする ePWM イベントをイネーブルにします。</li> <li>• イベントがトリガを引き起こす割合を指定します (発生ごと、2 回の発生ごと、または 3 回の発生ごと)。</li> <li>• イベントフラグをポーリング、セット、またはクリアします。</li> </ul>

コード例は本書の残りの部分に記載されており、さまざまな ePWM モジュール・コンフィギュレーションの実装方法を示しています。これらの例では、例 2-1 に示した定数定義が使用されます。これらの定義は、『C280x C/C++ Header Files and Peripheral Examples』(SPRC191) でも使用されています。

**例 2-1 コード例で使用される定数定義**

```

// TBCTL (Time-Base Control)
// = = = = =
// TBCTR MODE bits
#define TB_COUNT_UP 0x0
#define TB_COUNT_DOWN 0x1
#define TB_COUNT_UPDOWN 0x2
#define TB_FREEZE 0x3
// PHSEN bit
#define TB_DISABLE 0x0
#define TB_ENABLE 0x1
// PRDL bit
#define TB_SHADOW 0x0
#define TB_IMMEDIATE 0x1
// SYNCSEL bits
#define TB_SYNC_IN 0x0
#define TB_CTR_ZERO 0x1
#define TB_CTR_CMPB 0x2
#define TB_SYNC_DISABLE 0x3
// HSPCLKDIV and CLKDIV bits
#define TB_DIV1 0x0
#define TB_DIV2 0x1
#define TB_DIV4 0x2
// PHSDIR bit
#define TB_DOWN 0x0
#define TB_UP 0x1
// CMPCTL (Compare Control)
// = = = = =
// LOADAMODE and LOADBMODE bits
#define CC_CTR_ZERO 0x0
#define CC_CTR_PRD 0x1
#define CC_CTR_ZERO_PRD 0x2
#define CC_LD_DISABLE 0x3
// SHDWAMODE and SHDWBMODE bits
#define CC_SHADOW 0x0
#define CC_IMMEDIATE 0x1
// AQCTLA and AQCTLB (Action-qualifier Control)
// = = = = =
// ZRO, PRD, CAU, CAD, CBU, CBD bits
#define AQ_NO_ACTION 0x0
#define AQ_CLEAR 0x1
#define AQ_SET 0x2
#define AQ_TOGGLE 0x3
// DBCTL (Dead-Band Control)
// = = = = =
// MODE bits
#define DB_DISABLE 0x0
#define DBA_ENABLE 0x1
  
```

**例 2-1 コード例で使用される定数定義 (続き)**

```

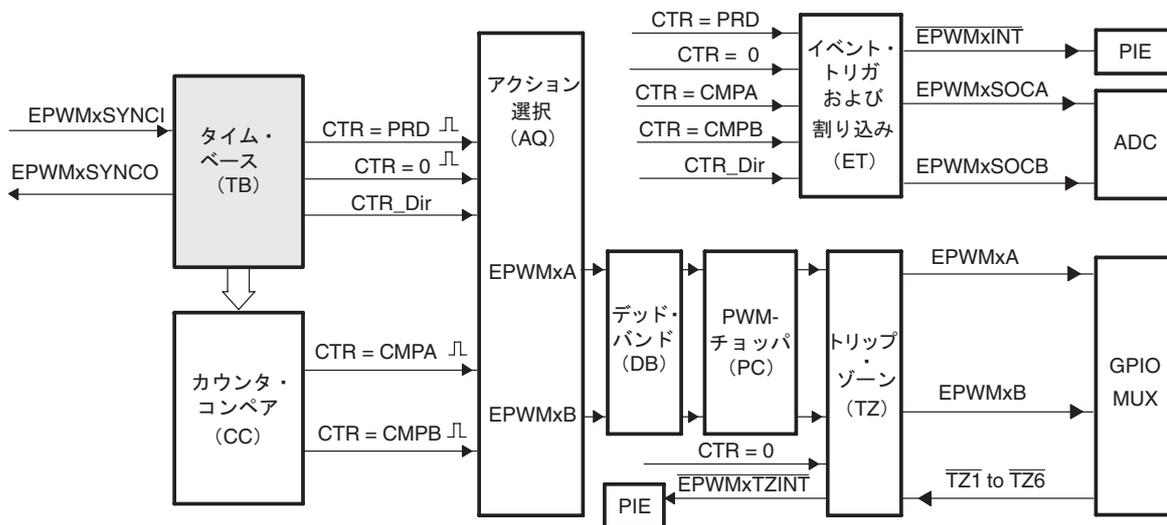
#define          DBB_ENABLE          0x2
#define DB_FULL_ENABLE 0x3
// POLSEL bits
#define          DB_ACTV_HI          0x0
#define          DB_ACTV_LO         0x1
#define          DB_ACTV_HIC        0x2
#define          DB_ACTV_LO         0x3
// CHPCTL (chopper control)
// = = = = =
// CHPEN bit
#define          CHP_ENABLE          0x0
#define CHP_DISABLE 0x1
// CHPFREQ bits
#define          CHP_DIV1           0x0
#define          CHP_DIV2           0x1
#define          CHP_DIV3           0x2
    #define          CHP_DIV4           0x3
#define          CHP_DIV5           0x4
#define          CHP_DIV6           0x5
#define          CHP_DIV7           0x6
#define          CHP_DIV8           0x7
// CHPDUTY bits
#define          CHP1_8TH           0x0
#define          CHP2_8TH           0x1
#define          CHP3_8TH           0x2
#define          CHP4_8TH           0x3
#define          CHP5_8TH           0x4
#define          CHP6_8TH           0x5
#define          CHP7_8TH           0x6
// TZSEL (Trip-zone Select)
// = = = = =
// CBCn and OSHTn bits
#define          TZ_ENABLE          0x0
#define          TZ_DISABLE         0x1
// TZCTL (Trip-zone Control)
// = = = = =
// TZA and TZB bits
#define          TZ_HIZ             0x0
#define          TZ_FORCE_HI        0x1
#define          TZ_FORCE_LO        0x2
#define          TZ_DISABLE         0x3
// ETSEL (Event-trigger Select)
// = = = = =
// INTSEL, SOCASEL, SOCBSEL bits
#define          ET_CTR_ZERO        0x1
#define          ET_CTR_PRD         0x2
#define          ET_CTRU_CMPA       0x4
#define          ET_CTRD_CMPA       0x5
#define          ET_CTRU_CMPB       0x6
#define          ET_CTRD_CMPB       0x7
// ETPS (Event-trigger Prescale)
// = = = = =
// INTPRD, SOCAPRD, SOCBPRD bits
#define          ET_DISABLE         0x0
#define          ET_1ST             0x1
#define          ET_2ND             0x2
#define          ET_3RD             0x3

```

## 2.2 タイム・ベース (TB) サブモジュール

各 ePWM モジュールには、ePWM モジュールのイベント・タイミングのすべてを決定する独立したタイム・ベース・サブモジュールがあります。同期ロジックが組み込まれているので、複数の ePWM モジュールのタイム・ベースを単一のシステムとして協調動作させることができます。図 2-1 は、ePWM 内のタイム・ベース・モジュールの場所を表します。

図 2-1 タイム・ベース・サブモジュールのブロック図



### 2.2.1 タイム・ベース・サブモジュールの目的

タイム・ベース・サブモジュールは、以下について設定できます。

- ePWM タイム・ベース・カウンタ (TBCTR) の周波数または周期を指定して、イベントの発生頻度を制御する。
- 他の ePWM モジュールとのタイム・ベース同期を管理する。
- 他の ePWM モジュールとの位相関係を管理する。
- タイム・ベース・カウンタをカウントアップ、カウントダウン、またはカウントアップ & ダウン・モードに設定する。
- 以下のイベントを生成する。
  - CTR=PRD: タイム・ベース・カウンタは指定された周期に等しい (TBCTR=TBPRD)
  - CTR=Zero: タイム・ベース・カウンタはゼロに等しい (TBCTR=0x0000)
- タイム・ベース・クロックのレートを設定する。CPU システム・クロック (SYSCLKOUT) をプリスケールします。これにより、タイム・ベース・カウンタをより遅いレートでインクリメント/デクリメントできるようになります。



表 2-3 重要なタイム・ベース信号

信号	説明
EPWMxSYNCI	<p>タイム・ベース同期入力</p> <p>同期チェーンの前の ePWM モジュールのカウントと同期させるための入力パルス。ePWM ペリフェラルは、この信号を使用または無視するように設定できます。最初の ePWM モジュール (EPWM1) の場合、この信号はデバイス・ピンから受け取ります。後続の ePWM モジュールの場合、この信号は別の ePWM ペリフェラルから渡されます。たとえば、EPWM2SYNCI は ePWM1 ペリフェラルによって生成され、EPWM3SYNCI は ePWM2 によって生成されます。特定のデバイスの同期順序については、<a href="#">セクション 2.2.3.2</a> を参照してください。</p>
EPWMxSYNCO	<p>タイム・ベース同期出力</p> <p>この出力パルスは、同期チェーンの後の ePWM モジュールのカウントを同期させるために使用されます。ePWM モジュールは、この信号を以下の 3 つのイベント・ソースのいずれかから生成します。</p> <ol style="list-style-type: none"> <li>1. EPWMxSYNCI (同期入力パルス)</li> <li>2. CTR=Zero: タイム・ベース・カウンタがゼロに等しい時 (TBCTR=0x0000)</li> <li>3. CTR=CMPB: タイム・ベース・カウンタがカウンタ・コンペア B レジスタに等しい時 (TBCTR=CMPB)</li> </ol>
CTR=PRD	<p>タイム・ベース・カウンタは指定された周期に等しい</p> <p>この信号は、カウンタ値がアクティブな周期レジスタ値と等しい場合 (つまり、TBCTR=TBPRD の場合) に必ず生成されます。</p>
CTR=Zero	<p>タイム・ベース・カウンタはゼロに等しい</p> <p>この信号は、カウンタ値がゼロの場合 (つまり、TBCTR が 0x0000 に等しい場合) に必ず生成されます。</p>
CTR=CMPB	<p>タイム・ベース・カウンタはアクティブ・カウンタ・コンペア B レジスタに等しい (TBCTR=CMPB)</p> <p>このイベントは、カウンタ・コンペア・サブモジュールによって生成され、同期出力ロジックによって使用されます。</p>
CTR_dir	<p>タイム・ベース・カウンタの方向</p> <p>ePWM タイム・ベース・カウンタの現在の方向を示します。この信号は、カウンタが増加しているときは High、減少しているときは Low になります。</p>
CTR_max	<p>タイム・ベース・カウンタは最大値に等しい (TBCTR=0xFFFF)</p> <p>TBCTR 値が最大値に達したときに生成されるイベントです。この信号は、ステータス・ビットとしてのみ使用されます。</p>
TBCLK	<p>タイム・ベース・クロック</p> <p>これは、システム・クロック (SYSCLKOUT) がプリスケールされたものであり、ePWM 内のすべてのサブモジュールによって使用されます。このクロックによって、タイム・ベース・カウンタがインクリメントまたはデクリメントされるレートが決まります。</p>

### 2.2.3 PWM 周期および周波数の計算

PWM イベントの周波数は、タイム・ベース周期 (TBPRD) レジスタとタイム・ベース・カウンタのモードによって制御されます。[図 2-3](#) は、周期を 4 (TBPRD=4) に設定した場合のアップ・カウント、ダウン・カウント、およびアップ・ダウン・カウント・タイム・ベース・カウンタモードの周期 ( $T_{pwm}$ ) と周波数 ( $F_{pwm}$ ) の関係を示しています。各ステップのタイム・ベース・インクリメント時間は、システム・クロック (SYSCLKOUT) がプリスケールされたタイム・ベース・クロック (TBCLK) によって定義されます。

タイム・ベース・カウンタには、タイム・ベース制御レジスタ (TBCTRL) によって選択される 3 つの動作モードがあります。

- **アップ・ダウン・カウント・モード:**

アップ・ダウン・カウント・モードでは、タイム・ベース・カウンタはゼロから始まり、周期 (TBPRD) 値に達するまでインクリメントされます。周期値に達すると、次にタイム・ベース・カウンタはゼロに達するまでデクリメントされます。この時点で、カウンタはパターンを繰り返し、インクリメントを開始します。

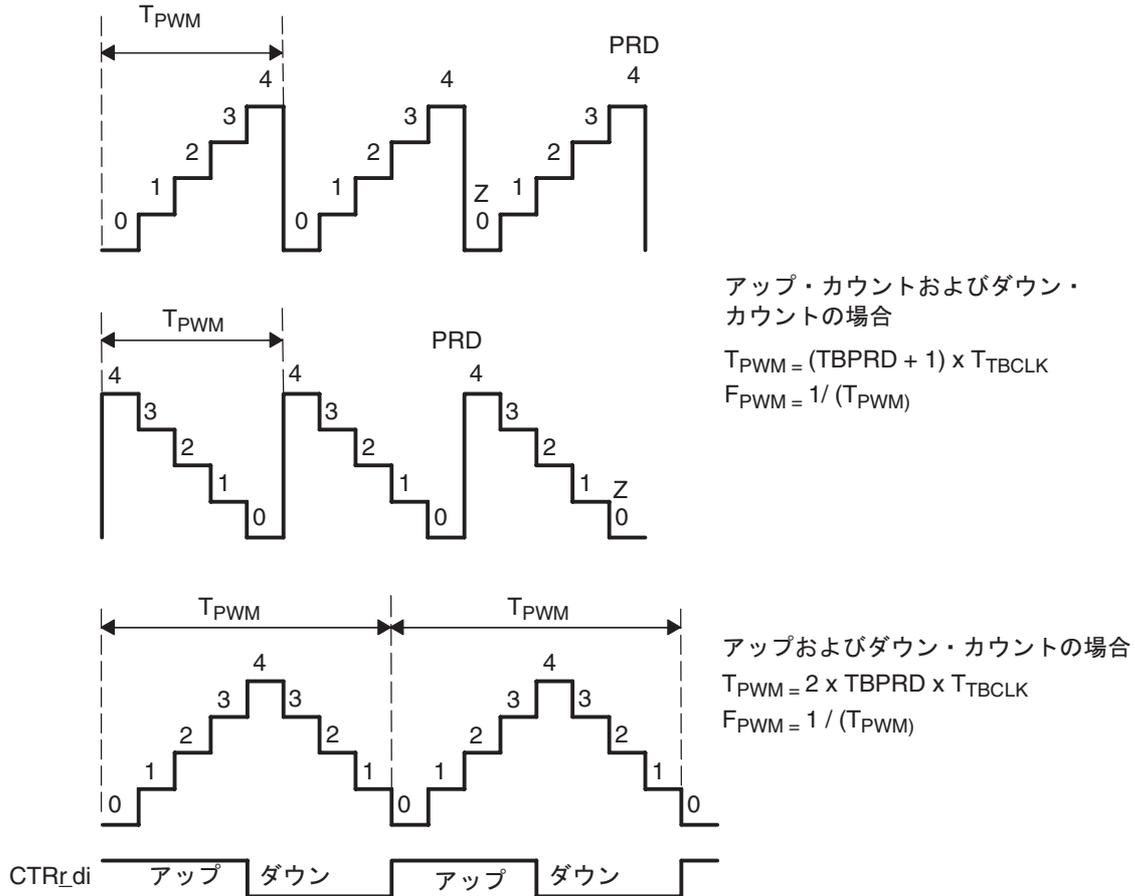
• **アップ・カウント・モード :**

このモードでは、タイム・ベース・カウンタはゼロから始まり、周期レジスタ (TBPRD) の値に達するまでインクリメントされます。周期値に達すると、タイム・ベース・カウンタはゼロにリセットされ、もう一度インクリメントを開始します。

• **ダウン・カウント・モード :**

ダウン・カウント・モードでは、タイム・ベース・カウンタは周期 (TBPRD) 値から始まり、ゼロに達するまでデクリメントされます。ゼロに達すると、タイム・ベース・カウンタは周期値にリセットされ、再びデクリメントを開始します。

図 2-3 タイム・ベース周波数および周期



2.2.3.1 タイム・ベース周期シャドウ・レジスタ

タイム・ベース周期レジスタ (TBPRD) にはシャドウ・レジスタがあります。シャドウ化により、レジスタ更新をハードウェアと同期させることができます。

以下の定義は、ePWM モジュールのすべてのシャドウ・レジスタを表すために使用されます。

• **アクティブ・レジスタ :**

アクティブ・レジスタはハードウェアを制御し、ハードウェアが実行するアクションまたは呼び出すアクションの要因となります。

• **シャドウ・レジスタ :**

シャドウ・レジスタは、アクティブ・レジスタの一時的な保持を行うバッファです。制御ハードウェアには直接影響しません。ある特定のタイミングで、シャドウ・レジスタの内容はアクティブ・レジスタに転送されます。これにより、ソフトウェアによるレジスタの非同期変更が原因となる破損や誤動作が生じることはなくなります。

シャドウ周期レジスタのメモリ・アドレスは、アクティブ・レジスタと同じです。ライト先のレジスタやリード元のレジスタは、TBCTL[PRDL] ビットによって決定されます。このビットは、以下のように TBPRD シャドウ・レジスタをイネーブルまたはディスエーブルにします。

- **タイム・ベース周期シャドウ・モード:**

TBCTL[PRDL]=0 の場合、TBPRD シャドウ・レジスタはイネーブルになります。TBPRD メモリ・アドレスからのリードと TBPRD メモリ・アドレスへのライトは、シャドウ・レジスタに対して行われます。タイム・ベース・カウンタがゼロに等しい場合は (TBCTR=0x0000)、シャドウ・レジスタの内容がアクティブ・レジスタに転送されます (TBPRD (アクティブ) ←TBPRD (シャドウ))。デフォルトでは、TBPRD シャドウ・レジスタはイネーブルになります。

- **タイム・ベース周期即時ロード・モード:**

即時ロード・モードを選択した場合 (TBCTL[PRDL]=1)、TBPRD メモリ・アドレスからのリードや TBPRD メモリ・アドレスへのライトはアクティブ・レジスタに対して直接行われます。

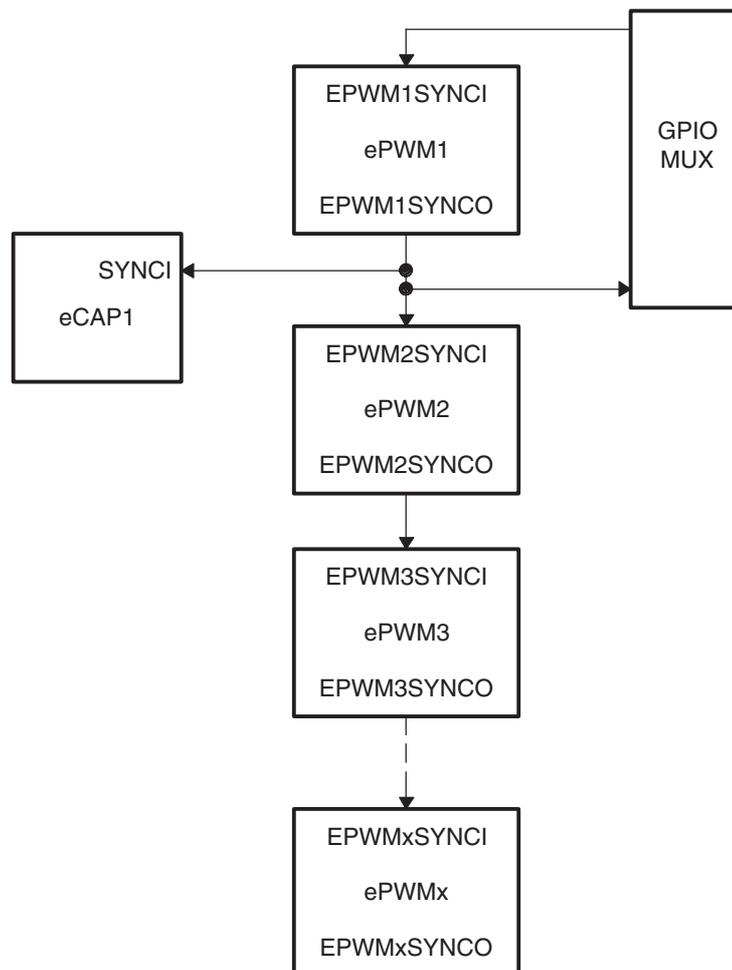
### 2.2.3.2 タイム・ベース・カウンタ同期

タイム・ベース同期構造により、デバイス上の ePWM モジュールがすべて接続されています。各 ePWM モジュールには、同期入力 (EPWMxSYNCI) と同期出力 (EPWMxSYNCO) があります。最初のモジュール (ePWM1) の同期入力は、外部ピンから受け取ります。残りの ePWM モジュールの可能な同期接続を [図 2-4](#) と [図 2-5](#) に示します。

[図 2-4](#) に示した構造 1 は、280x および 2801x デバイスに適用されます。

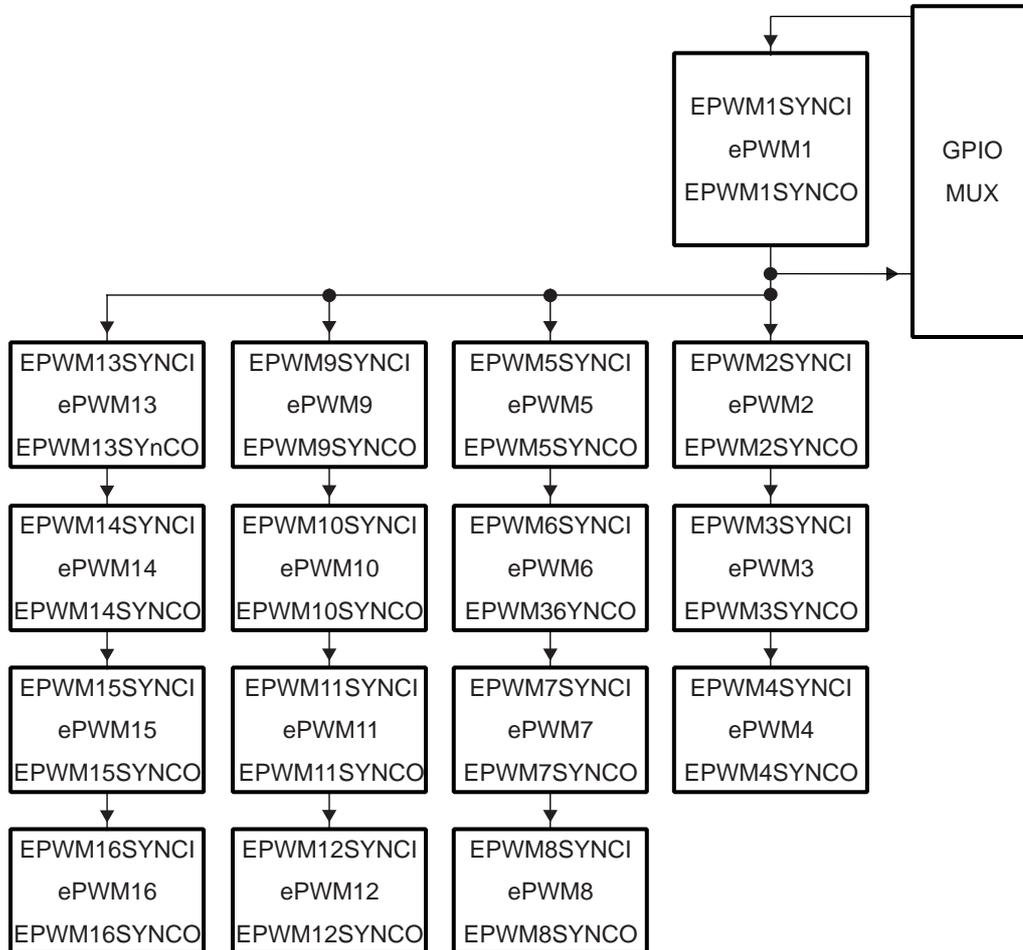
ePWM ピンアウトが 280x 互換モード用に設定されている場合は (GPAMCFG[EPWMMODE]=0)、2804x デバイスにも構造 1 が適用されます。

図 2-4 タイム・ベース・カウンタ同期構造 1



2804x デバイスにおいて ePWM ピンアウトが A チャンネル専用モードに設定されている場合は (GPAMCFG[EPWMMODE]=3)、[図 2-5](#) に示した構造 2 が使用されます。2804x ePWM ピンアウトを 280x 互換モード用に設定した場合は (GPAMCFG[EPWMMODE]=0)、構造 1 が使用されます。

図 2-5 タイム・ベース・カウンタ同期構造 2



各 ePWM モジュールは、同期入力を使用または無視するように設定できます。TBCTL[PHSEN] ビットをセットした場合は、以下のいずれかの状態が発生すると、ePWM モジュールのタイム・ベース・カウンタ (TBCTR) に位相レジスタ (TBPHS) の内容が自動的にロードされます。

• **EPWMxSYNCI: 同期入力パルス :**

入力同期パルスが検出されると、位相レジスタの値がカウンタ・レジスタにロードされます (TBPHS → TBCNT)。この動作は、次の有効なタイム・ベース・クロック (TBCLK) エッジで行われます。

• **ソフトウェア強制同期パルス :**

TBCTL[SWFSYNC] 制御ビットに 1 をライトすると、ソフトウェア強制同期が実行されます。このパルスは同期入力信号と論理和演算されるため、EPWMxSYNCI のパルスと同じ効果があります。

この機能により、ePWM モジュールは別の ePWM モジュールのタイム・ベースと自動的に同期できます。異なる ePWM モジュールを同期させることにより、位相を進めたり遅らせたりすることができます。アップ・ダウン・カウント・モードでは、TBCTL[PSHDIR] ビットによって、同期イベントの直後にタイム・ベース・カウンタの方向が設定されます。新しい方向は、同期イベント以前の方向に依存しません。TBPHS ビットは、カウントアップ・モードやカウントダウン・モードでは無視されます。例については、[図 2-6](#) ~ [図 2-9](#) を参照してください。

TBCTL[PHSEN] ビットをクリアすると、同期入力パルスを見捨てるように ePWM が設定されます。入力された同期パルスは、そのまま EPWMxSYNCO へ出力でき、他の ePWM モジュールの同期に使用できます。このようにして、マスタ・タイム・ベース（たとえば、ePWM1）をセットアップし、以降のモジュール（ePWM2 ~ ePWMx）では、そのマスタと同期して動作することを選択できます。同期の使い方の詳細については、第 3 章「パワー・トポロジへの応用」を参照してください。

#### 2.2.4 複数 ePWM モジュールのタイム・ベース・クロックの位相ロック

TBCLKSYNC ビットを使用すると、デバイス上のイネーブルになっている ePWM モジュールすべてのタイム・ベース・クロックをグローバルに同期させることができます。このビットは DSP クロック・イネーブル・レジスタの一部であり、『TMS320x280x, 2801x, 2804x System Control and Interrupts Reference Guide』（SPRU712）に記載されています。TBCLKSYNC=0 の場合は、すべての ePWM モジュールのタイム・ベース・クロックが停止されます（デフォルト）。TBCLKSYNC=1 の場合は、調整された TBCLK の立ち上がりエッジですべての ePWM タイム・ベース・クロックが開始されます。完全に TBCLK を同期するには、各 ePWM モジュールの TBCTL レジスタのプリスケアラ・ビットを同じ設定にする必要があります。ePWM クロックをイネーブルにする正しい手順は以下のとおりです。

1. 個々の ePWM モジュール・クロックをイネーブルにします。これは、『TMS320x280x, 2801x, 2804x System Control and Interrupts Reference Guide』（SPRU712）に記載されています。
2. TBCLKSYNC=0 を設定します。これにより、イネーブルになっている ePWM モジュール内のタイム・ベース・クロックが停止されます。
3. プリスケアラ値と必要な ePWM モードを設定します。
4. TBCLKSYNC=1 を設定します。

#### 2.2.5 タイム・ベース・カウンタ・モードとタイミング波形

タイム・ベース・カウンタは、以下の 4 つのモードのいずれかで動作します。

- 非対称であるアップ・カウント・モード
- 非対称であるダウン・カウント・モード
- 対称であるアップ・ダウン・カウント
- タイム・ベース・カウンタが現在値で一定に保たれている

最初の 3 つのモードの動作を説明するために、いつイベントが生成され、どのようにタイム・ベースが EPWMxSYNCl 信号に反応するのかを以下のタイミング図に示します。

図 2-6 タイム・ベース・アップ・カウント・モード波形

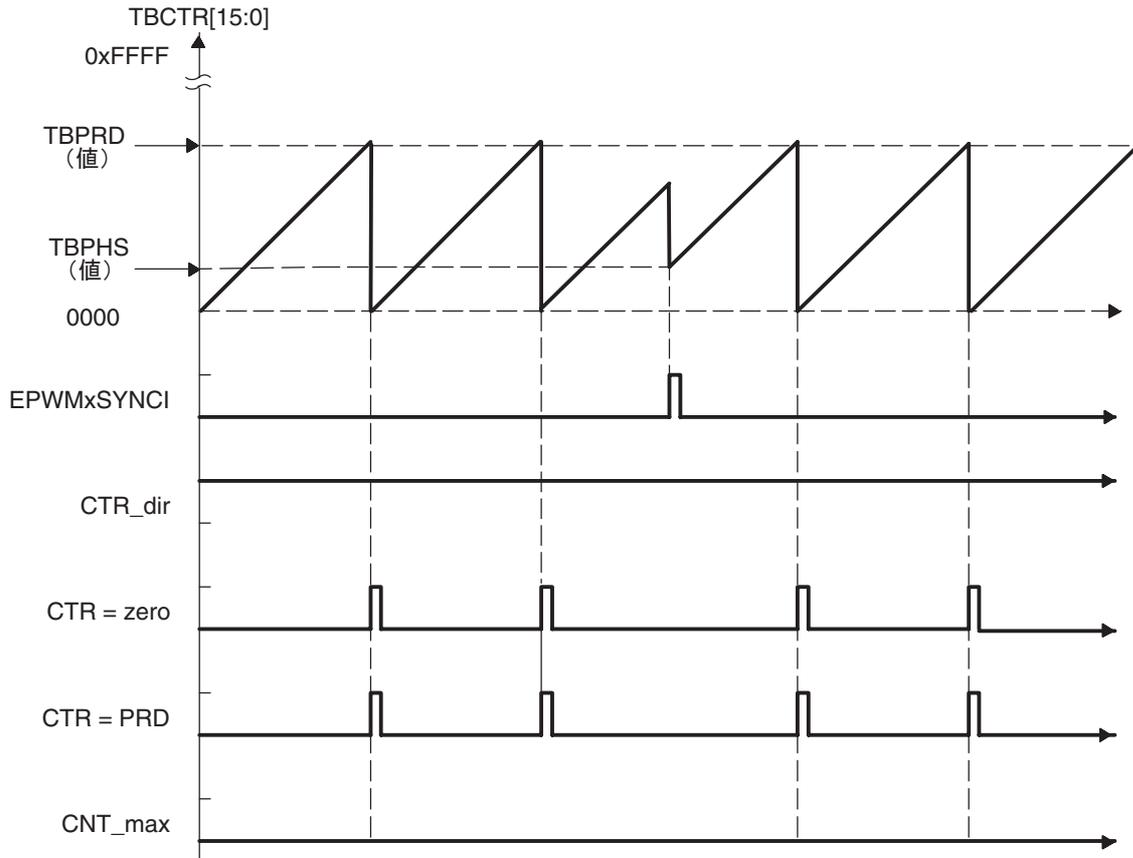


図 2-7 タイム・ベース・ダウン・カウント・モード波形

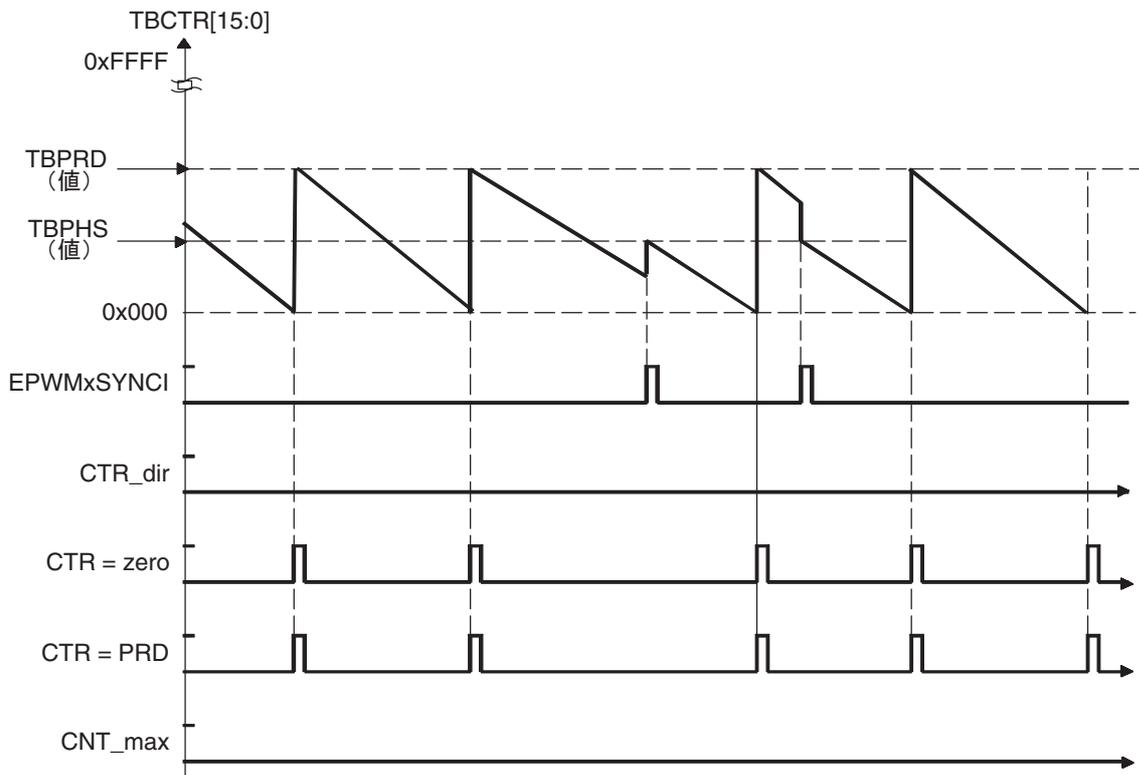


図 2-8 タイム・ベース・アップ・ダウン・カウント波形、TBCTL[PHSDIR=0]、同期イベントでカウント・ダウン

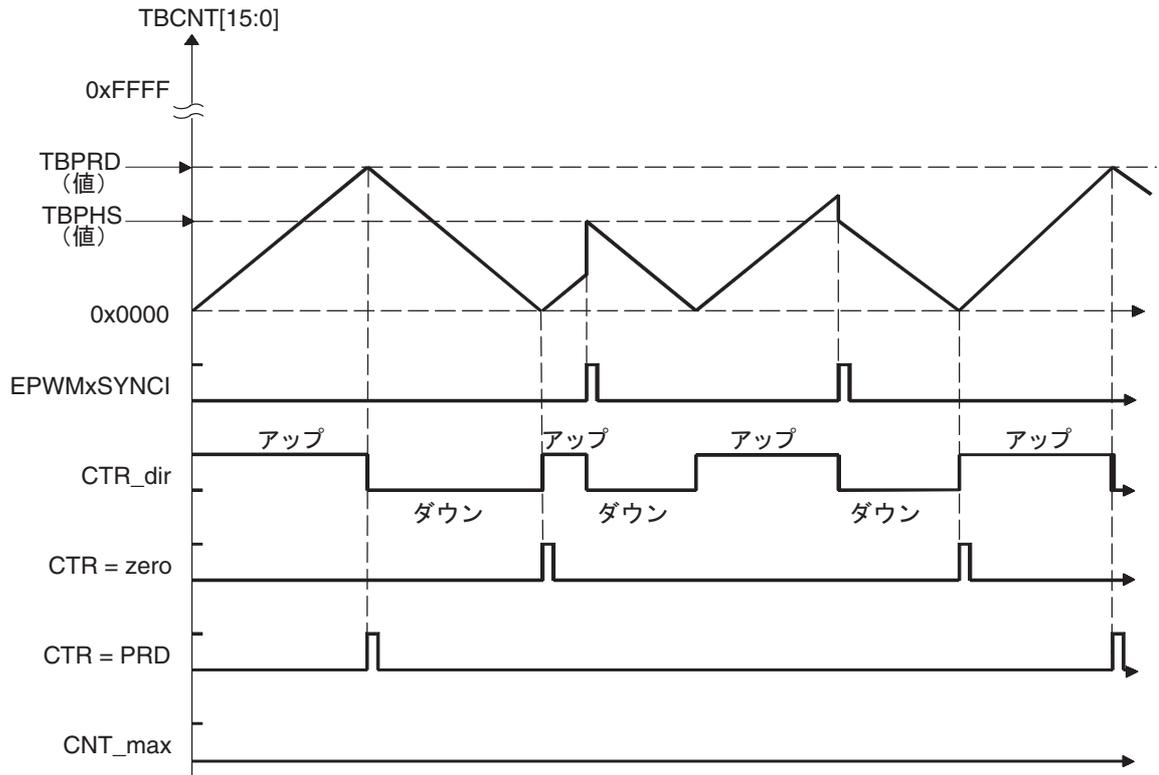
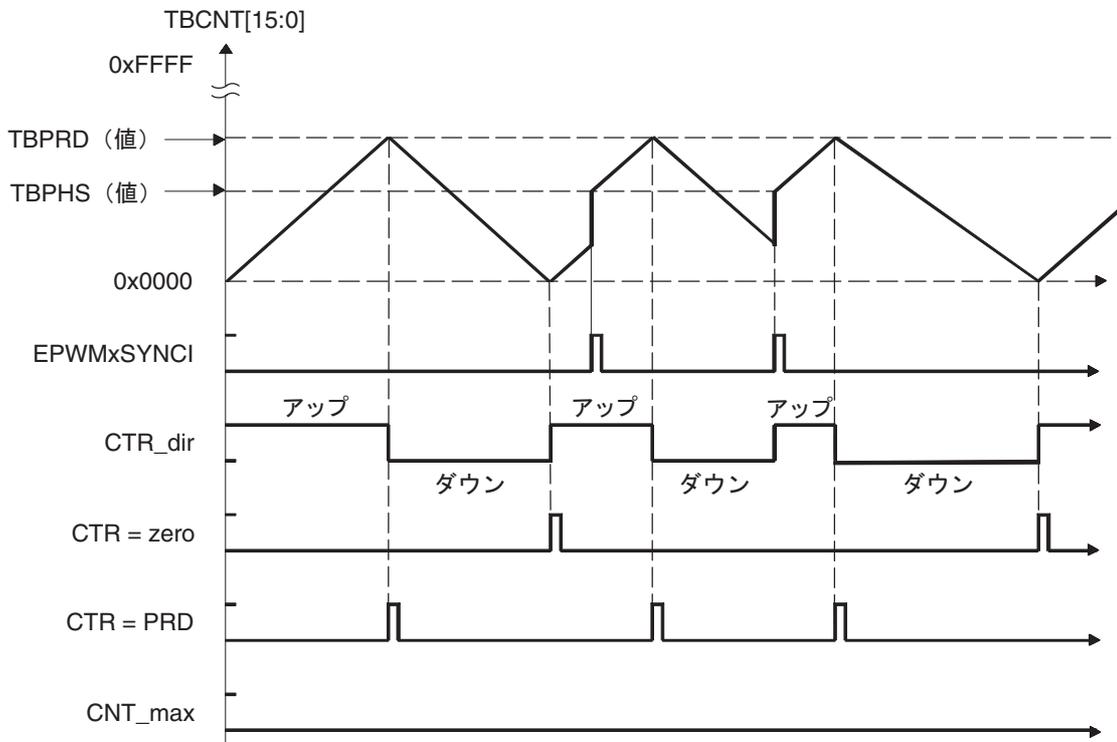


図 2-9 タイム・ベース・アップ・ダウン・カウント波形、TBCTL[PHSDIR=1]、同期イベントでカウント・アップ



## 2.3 カウンタ・コンペア (CC) サブモジュール

図 2-10 は、ePWM 内のカウンタ・コンペア・サブモジュールを表します。

図 2-10 カウンタ・コンペア・サブモジュール

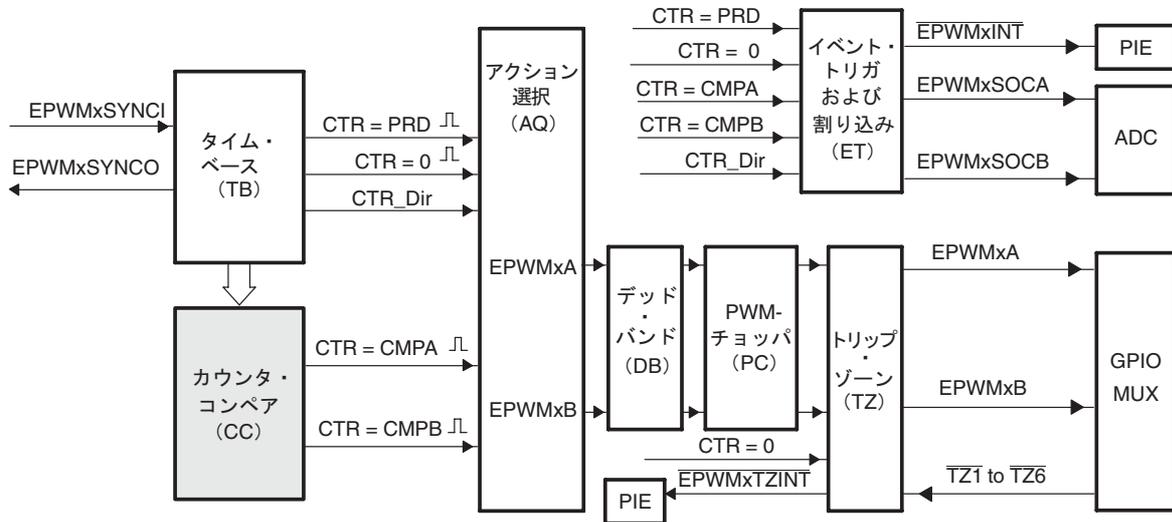


図 2-11 は、カウンタ・コンペア・サブモジュールの基本構造を示しています。

### 2.3.1 カウンタ・コンペア・サブモジュールの目的

カウンタ・コンペア・サブモジュールは、タイム・ベース・カウンタ値を入力として使用します。この値は、カウンタ・コンペア A (CMPA) およびカウンタ・コンペア B (CMPB) レジスタと常にコンペアされます。

タイム・ベース・カウンタがいずれかのコンペア・レジスタと等しい場合は、カウンタ・コンペア・ユニットが適切なイベントを生成します。

カウンタ・コンペアでは、以下の処理が行われます。

- CMPA および CMPB レジスタを使用し、プログラマブル・タイム・スタンプに基づいてイベントを生成する
  - CTR=CMPA: タイム・ベース・カウンタはカウンタ・コンペア A レジスタに等しい (TBCTR=CMPA)
  - CTR=CMPB: タイム・ベース・カウンタはカウンタ・コンペア B レジスタに等しい (TBCTR=CMPB)
- アクション選択サブモジュールが適切に設定されている場合は、PWM デューティ・サイクルを制御する
- 新しいコンペア値をシャドウ化して、アクティブ PWM サイクル中の破損やグリッチを防止する

### 2.3.2 カウンタ・コンペア・サブモジュールの制御と監視

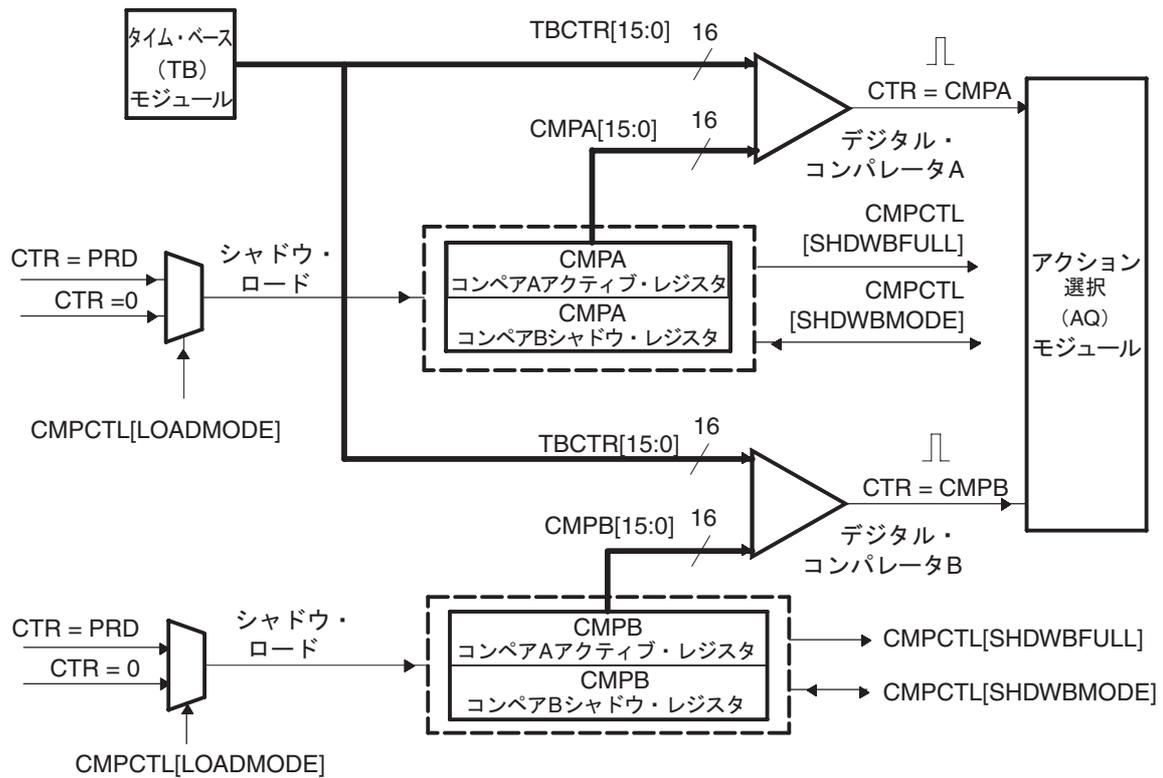
カウンタ・コンペア・サブモジュールの動作は、表 2-4 に示すレジスタによって制御および監視されます。

表 2-4 カウンタ・コンペア・サブモジュールのレジスタ

レジスタ名	アドレス・オフセット	シャドウ化	説明
CMPCTL	0x0007	無	カウンタ・コンペア制御レジスタ
CMPAHR	0x0008	有	HRPWM カウンタ・コンペア A 拡張レジスタ (1)
CMPA	0x0009	有	カウンタ・コンペア A レジスタ
CMPB	0x000A	有	カウンタ・コンペア B レジスタ

(1) このレジスタは、高分解能拡張 (HRPWM) が含まれている ePWM モジュールでのみ利用できます。HRPWM が含まれていない ePWM モジュールでは、このロケーションは予約されます。このレジスタの説明は、『TMS320x28xx, 28xxx High-Resolution Pulse Width Modulator (HRPWM) Reference Guide』(SPRU924)に記載されています。どの ePWM モジュールにこの機能が含まれているのかを確認するには、各デバイスのデータ・マニュアルを参照してください。

図 2-11 カウンタ・コンペア・サブモジュールの詳細図



カウンタ・コンペア・サブモジュールに関する重要な信号は、表 2-5 に記載されています。

表 2-5 カウンタ・コンペア・サブモジュールの重要な信号

信号	イベントの説明	コンペアされるレジスタ
CTR=CMPA	タイム・ベース・カウンタはアクティブなカウンタ・コンペア A 値に等しい	TBCTR=CMPA
CTR=CMPB	タイム・ベース・カウンタはアクティブなカウンタ・コンペア B 値に等しい	TBCTR=CMPB
CTR=PRD	タイム・ベース・カウンタはアクティブな周期に等しい アクティブなカウンタ・コンペア A および B レジスタをシャドウ・レジスタからロードするために使用される	TBCTR=TBPRD
CTR=ZERO	タイム・ベース・カウンタはゼロに等しい アクティブなカウンタ・コンペア A および B レジスタをシャドウ・レジスタからロードするために使用される	TBCTR=0x0000

### 2.3.3 カウンタ・コンペア・サブモジュールの動作上の重要点

カウンタ・コンペア・サブモジュールは、以下の 2 つのコンペア・レジスタに基づいて 2 つの独立したコンペア・イベントを生成します。

1. CTR=CMPA: タイム・ベース・カウンタはカウンタ・コンペア A レジスタに等しい (TBCTR=CMPA)
2. CTR=CMPB: タイム・ベース・カウンタはカウンタ・コンペア B レジスタに等しい (TBCTR=CMPB)

アップ・カウントまたはダウン・カウント・モードでは、各イベントはサイクルごとに 1 回だけ発生します。

アップ・ダウン・カウント・モードでは、各イベントは、コンペア値が 0x0000 と TBPRD の間の場合はサイクルごとに 2 回、コンペア値が 0x0000 または TBPRD に等しい場合はサイクルごとに 1 回発生します。

これらのイベントは、アクション選択サブモジュールに供給されます。アクション選択サブモジュールでは、これらのイベントがカウンタ方向によって判断され、イネーブルになっている場合はアクションに変換されます。詳細については [セクション 2.4.1](#) を参照してください。

カウンタ・コンペア・レジスタの CMPA と CMPB には、それぞれ対応するシャドウ・レジスタがあります。シャドウ化により、ハードウェアと同期しているレジスタの更新のタイミングを維持する方法が提供されます。シャドウ化を使用すると、アクティブ・レジスタの更新は特定の時点でのみ行われます。これにより、ソフトウェアによるレジスタの非同期変更が原因となって破損や誤動作が生じることはなくなります。アクティブ・レジスタとシャドウ・レジスタのメモリ・アドレスは同一です。ライト先のレジスタやリード元のレジスタは、CMPCTL[SHDWAMODE] および CMPCTL[SHDWBMODE] ビットによって決定されます。これらのビットを使用すると、それぞれ CMPA シャドウ・レジスタと CMPB シャドウ・レジスタがイネーブルまたはディスエーブルになります。2 つのロード・モードの動作を以下に説明します。

• **シャドウ・モード：**

CMPA のシャドウ・モードは、CMPCTL[SHDWAMODE] ビットをクリアするとイネーブルになります。CMPB のシャドウ・レジスタは、CMPCTL[SHDWBMODE] ビットをクリアするとイネーブルになります。CMPA と CMPB のどちらについても、シャドウ・モードはデフォルトでイネーブルになります。CMPA シャドウ・レジスタがイネーブルになっている場合は、以下のいずれかのイベントでシャドウ・レジスタの内容がアクティブ・レジスタに転送されます。

- CTR=PRD: タイム・ベース・カウンタは周期に等しい (TBCTR=TBPRD)
- CTR=Zero: タイム・ベース・カウンタはゼロに等しい (TBCTR=0x0000)
- CTR=PRD と CTR=Zero の両方

この3つのイベントのどれを使用するのは、CMPCTL[LOADAMODE] および CMPCTL[LOADBMODE] レジスタ・ビットによって指定されます。カウンタ・コンペア・サブモジュールでは、アクティブ・レジスタの内容だけを使用して、アクション選択へ送るイベントが生成されます。

• **即時ロード・モード：**

CMPA 即時ロード・モードを選択した場合 (TBCTL[SHADWAMODE]=1 または TBCTL[SHADWBMODE]=1)、レジスタからのリードやレジスタへのライトはアクティブ・レジスタに対して直接行われます。

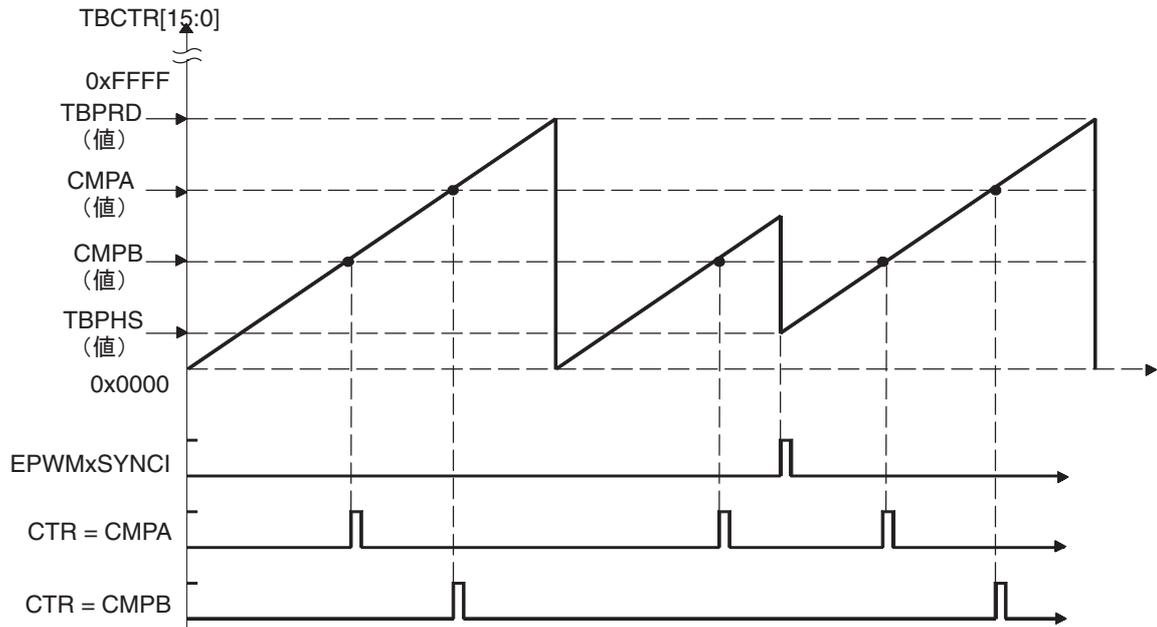
### 2.3.4 カウント・モード・タイミング波形

カウンタ・コンペア・モジュールは、以下の3つのカウント・モードでコンペア・イベントを生成できます。

- アップ・カウント・モード：非対称 PWM 波形の生成に使用されます。
- ダウン・カウント・モード：非対称 PWM 波形の生成に使用されます。
- アップ・ダウン・カウント・モード：対称 PWM 波形の生成に使用されます。

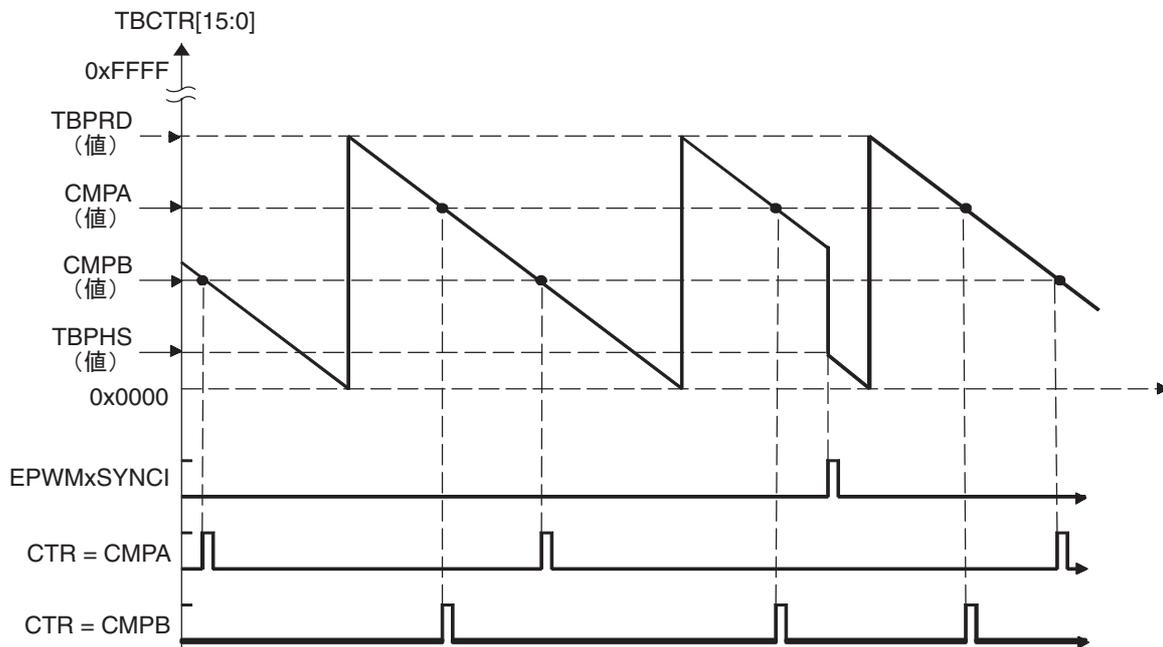
最初の 3 つのモードの動作を分かりやすく説明するために、いつイベントが生成され、どのように EPWMxSYNCl 信号が相互作用するのかを図 2-12 ~ 図 2-15 のタイミング図に示します。

図 2-12 アップ・カウント・モードのカウンタ・コンペア・イベント波形



注： EPWMxSYNCI 外部同期イベントは、TBCTR カウンタ・シーケンスの不連続性を引き起こす可能性があります。これにより、場合によってはコンペア・イベントがスキップされることがあります。このスキップは通常の動作とみなされるため、考慮に入れる必要があります。

図 2-13 ダウン・カウント・モードのカウンタ・コンペア・イベント

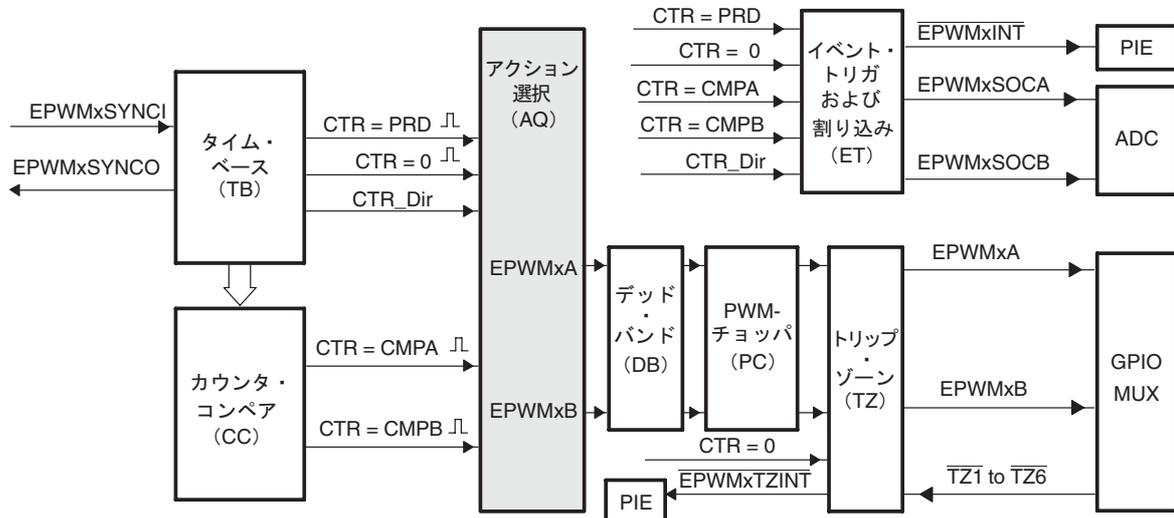




## 2.4 アクション選択 (AQ) サブモジュール

図 2-16 は、ePWM システムのアクション選択 (AQ) サブモジュールを示しています (濃く塗られたブロックを参照)。

図 2-16 アクション選択サブモジュール



アクション選択サブモジュールには、波形構成と PWM 生成における最も重要な役割があります。このサブモジュールは、どのイベントが各種アクション・タイプに変換されるのかを決定し、それによって必要なスイッチング波形が EPWMxA および EPWMxB 出力で生成されます。

### 2.4.1 アクション選択サブモジュールの目的

アクション選択サブモジュールは、以下の処理を担当します。

- 以下のイベントに基づいてアクション (セット、クリア、トグル) を選択および生成する
  - CTR=PRD: タイム・ベース・カウンタは周期に等しい (TBCTR=TBPRD)
  - CTR=Zero: タイム・ベース・カウンタはゼロに等しい (TBCTR=0x0000)
  - CTR=CMPA: タイム・ベース・カウンタはカウンタ・コンペア A レジスタに等しい (TBCTR=CMPA)
  - CTR=CMPB: タイム・ベース・カウンタはカウンタ・コンペア B レジスタに等しい (TBCTR=CMPB)
- これらのイベントが同時に発生した場合に優先順位を管理する
- タイム・ベース・カウンタが増加しているとき、および減少しているときに独立したイベント制御を提供する

### 2.4.2 アクション選択サブモジュールの制御およびステータス・レジスタ定義

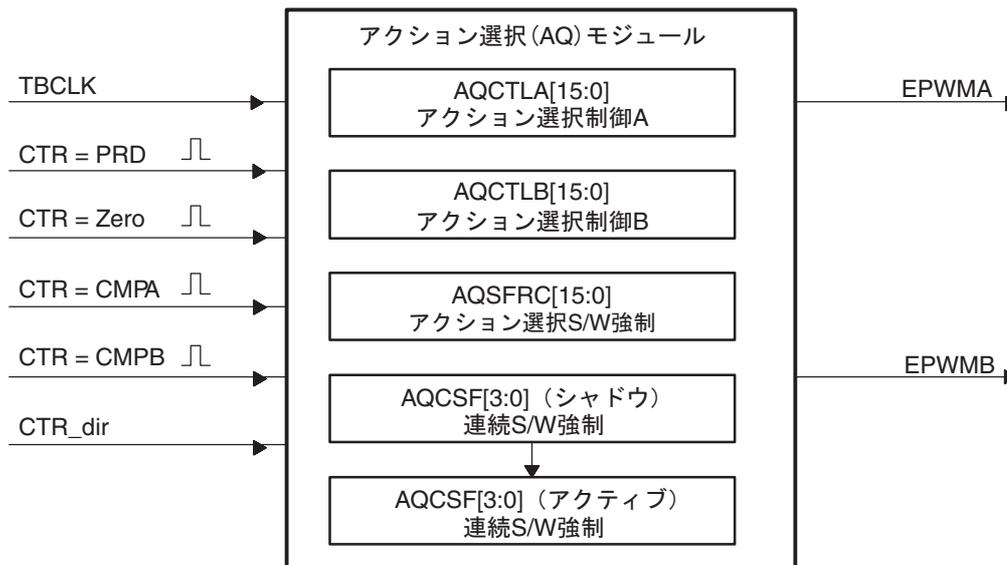
アクション選択サブモジュールの動作は、表 2-6 のレジスタによって制御および監視されます。

表 2-6 アクション選択サブモジュールのレジスタ

レジスタ名	アドレス・オフセット	シャドウ化	説明
AQCTLA	0x000B	無	出力 A のアクション選択制御レジスタ (EPWMxA)
AQCTLB	0x000C	無	出力 B のアクション選択制御レジスタ (EPWMxB)
AQSFRC	0x000D	無	アクション選択ソフトウェア強制レジスタ
AQCSFRC	0x000E	有	アクション選択連続ソフトウェア強制

アクション選択サブモジュールは、イベント駆動ロジックに基づいています。このサブモジュールは、入力がイベントで出力がアクションのプログラマブル・クロス・スイッチとみなすことができます。これらのイベントやアクションは、表 2-6 に示す一連のレジスタによってソフトウェア制御されます。

図 2-17 アクション選択サブモジュールの入力と出力



便宜上、可能な入カイベントの概要を表 2-7 にもう一度示します。

表 2-7 アクション選択サブモジュールの可能な入カイベント

信号	説明	コンペアされるレジスタ
CTR=PRD	タイム・ベース・カウンタは周期値に等しい	TBCTR=TBPRD
CTR=Zero	タイム・ベース・カウンタはゼロに等しい	TBCTR=0x0000
CTR=CMPA	タイム・ベース・カウンタはカウンタ・コンペア A に等しい	TBCTR=CMPA
CTR=CMPB	タイム・ベース・カウンタはカウンタ・コンペア B に等しい	TBCTR=CMPB
ソフトウェア強制イベント	ソフトウェアによって開始された非同期イベント	

ソフトウェア強制アクションは、便利な非同期イベントです。この制御は、AQSFRC および AQCSF レジスタによって処理されます。

アクション選択サブモジュールでは、特定のイベントの発生時に 2 つの出力 EPWMxA と EPWMxB がどのように動作するのが制御されます。アクション選択サブモジュールへのイベント入力は、カウンタ方向（アップまたはダウン）によってさらに判断されます。これにより、カウントアップ時カウントダウン時の両方の出力で独立したアクションが可能です。

EPWMxA および EPWMxB 出力に対して設定できるアクションは以下のとおりです。

- **High にセット：**  
EPWMxA または EPWMxB 出力を High レベルに設定します。
- **Low にクリア：**  
EPWMxA または EPWMxB 出力を Low レベルに設定します。
- **トグル：**  
EPWMxA または EPWMxB が High になっている場合は、その出力を Low にします。EPWMxA または EPWMxB が Low になっている場合は、その出力を High にします。

• 何もしない :

EPWMxA および EPWMxB 出力を現在設定されているレベルと同じレベルに維持します。「何もしない」オプションを使用すると、イベントが EPWMxA および EPWMxB 出力でアクションを引き起こすことはなくなりますが、このイベントは割り込みと ADC 変換開始を引き続きトリガすることができます。詳細については、[セクション 2.8](#) のイベント・トリガ・サブモジュールの説明を参照してください。

アクションは、どちらの出力 (EPWMxA または EPWMxB) についても独立して指定されます。特定の出力でアクションを生成するように任意またはすべてのイベントを設定できます。たとえば、CTR=CMPA と CTR=CMPB は、どちらも EPWMxA 出力で機能できます。選択アクションは、このセクションの最後に記載されている制御レジスタによって設定されます。

分かりやすくするために、本書の図では一連のアクションを示す記号で表しています。これらの記号の概略を [図 2-18](#) に示します。各記号は、アクションを時間のマーカとして表します。一部のアクションは時間が固定されていますが (ゼロおよび周期)、CMPA アクションと CMPB アクションは移動可能であり、それぞれカウンタ・コンペア A レジスタとカウンタ・コンペア B レジスタで時間位置がプログラムされます。アクションをオフまたはディスエーブルにするには、「何もしない」オプションを使用します。このオプションはリセット時のデフォルトです。

図 2-18 EPWMxA および EPWMxB 出力に対する可能なアクション選択のアクション

S/W 強制	TBカウンタが以下に等しい				アクション
	ゼロ	コンペア A	コンペア B	周期	
					何もしない
					Lowにクリア
					Highにセット
					トグル

### 2.4.3 アクション選択イベント優先順位

ePWM アクション選択では、複数のイベントを同時に受信できます。この場合は、ハードウェアによってイベントに優先順位が割り当てられます。一般的なルールとしては、後で発生するイベントほど優先順位が高くなり、ソフトウェア強制イベントは常に最優先になります。アップ・ダウン・カウント・モードのイベント優先順位レベルを表 2-8 に示します。優先順位レベル 1 は最も高い優先順位、レベル 7 は最も低い優先順位です。優先順位は、TBCNTR の方向に応じて多少変わります。

表 2-8 アップ・ダウン・カウント・モードのアクション選択イベント優先順位

優先順位レベル	TBCNTR=Zero から TBCNTR=TBPRD まで TBCNTR がインクリメント されている場合のイベント	TBCNTR=TBPRD から TBCNTR=1 まで TBCNTR がデクリメント されている場合のイベント
1 (最上位)	ソフトウェア強制イベント	ソフトウェア強制イベント
2	アップ・カウントでカウンタが CMPB に 等しい (CBU)	ダウン・カウントでカウンタが CMPB に 等しい (CBD)
3	アップ・カウントでカウンタが CMPA に 等しい (CAU)	ダウン・カウントでカウンタが CMPA に 等しい (CAD)
4	カウンタはゼロに等しい	カウンタは周期に等しい (TBPRD)
5	アップ・カウントでカウンタが CMPB に 等しい (CBD) <sup>(1)</sup>	アップ・カウントでカウンタが CMPB に 等しい (CBU) <sup>(1)</sup>
6 (最下位)	アップ・カウントでカウンタが CMPA に 等しい (CAD) <sup>(1)</sup>	アップ・カウントでカウンタが CMPA に 等しい (CBU) <sup>(1)</sup>

(1) アップ・ダウン・カウント・モードの対称性を維持するには、TBPRD と Zero についてアップイベント (CAU/CBU) とダウンイベント (CAD/CBD) の両方を生成できます。それ以外の場合は、アップイベントはカウンタがインクリメントされている場合にのみ、ダウンイベントはカウンタがデクリメントされている場合にのみ生成できます。

表 2-9 に、アップ・カウント・モードのアクション選択優先順位を示します。この場合、カウンタ方向は常にアップとして定義されるため、ダウン・カウント・イベントが使用されることはありません。

表 2-9 アップ・カウント・モードのアクション選択イベント優先順位

優先順位	レベルイベント
1 (最上位)	ソフトウェア強制イベント
2	カウンタは周期に等しい (TBPRD)
3	アップ・カウントでカウンタが CMPB に等しい (CBU)
4	アップ・カウントでカウンタが CMPA に等しい (CAU)
5 (最下位)	カウンタはゼロに等しい

表 2-10 に、ダウン・カウント・モードのアクション選択優先順位を示します。この場合、カウンタ方向は常にダウンとして定義されるため、アップ・カウント・イベントが使用されることはありません。

表 2-10 ダウン・カウント・モードのアクション選択イベント優先順位

優先順位	レベルイベント
1 (最上位)	ソフトウェア強制イベント
2	カウンタはゼロに等しい
3	ダウン・カウントでカウンタが CMPB に等しい (CBD)
4	ダウン・カウントでカウンタが CMPA に等しい (CAD)
5 (最下位)	カウンタは周期に等しい (TBPRD)

コンペア値は周期より大きい値に設定できます。この場合、アクションは表 2-11 に示したように行われます。

表 2-11 CMPA/CMPB が周期より大きい場合の動作

カウンタ・モード	アップ・カウント・イベントでのコンペア CAU/CBU	ダウン・カウント・イベントでのコンペア CAU/CBU
アップ・カウント・モード	CMPA/CMPB ≤ TBPRD 周期の場合は、コンペア・マッチ (TBCTR=CMPA または CMPB) でイベントが発生します。 CMPA/CMPB > TBPRD の場合は、イベントは発生しません。	行われません。
ダウン・カウント・モード	行われません。	CMPA/CMPB < TBPRD の場合は、コンペア・マッチ (TBCTR=CMPA または CMPB) でイベントが発生します。 CMPA/CMPB ≥ TBPRD の場合は、周期マッチ (TBCTR=TBPRD) でイベントが発生します。
アップ・ダウン・カウント・モード	CMPA/CMPB < TBPRD で、カウンタがインクリメントされている場合は、コンペア・マッチ (TBCTR=CMPA または CMPB) でイベントが発生します。 CMPA/CMPB ≥ TBPRD の場合は、周期マッチ (TBCTR=TBPRD) でイベントが発生します。	CMPA/CMPB < TBPRD で、カウンタがデクリメントされている場合は、コンペア・マッチ (TBCTR=CMPA または CMPB) でイベントが発生します。 CMPA/CMPB ≥ TBPRD の場合は、周期マッチ (TBCTR=TBPRD) でイベントが発生します。

#### 2.4.4 一般的な設定に対する波形

注：本書の波形は、静的コンペア・レジスタ値に対する ePWM 動作を示しています。一般に、稼働中のシステムでは、アクティブ・コンペア・レジスタ (CMPA と CMPB) はそれぞれのシャドウ・レジスタから周期ごとに 1 回更新されます。ユーザーは、いつ更新を行うのか (タイム・ベース・カウンタがゼロに達したとき、またはタイム・ベース・カウンタが周期に達したとき) を指定します。場合によっては、新しい値に基づいたアクションが 1 周期遅れたり、古い値に基づいたアクションが 1 周期余分に有効になったりすることがあります。この状況は、いくつかの PWM 設定によって回避されます。その設定には以下のものが含まれますが、これらに限定されるわけではありません。

##### アップ・ダウン・カウント・モードを使用して対称 PWM を生成：

- ゼロで CMPA/CMPB をロードする場合は、1 以上の CMPA/CMPB 値を使用します。
- 周期で CMPA/CMPB をロードする場合は、TBPRD-1 以下の CMPA/CMPB 値を使用します。つまり、PWM 周期内に 1 TBCLK サイクル以上のパルス (非常に短い場合はシステムによって無視される傾向がある) が常に存在することになります。

##### アップ・ダウン・カウント・モードを使用して非対称 PWM を生成：

- 50～0% 非対称 PWM を実現するには、以下の設定を使用します。周期で CMPA/CMPB をロードし、周期アクションを使用して PWM をクリアし、コンペアアップ・アクションを使用して PWM をセットします。コンペア値を 0 から TBPRD まで変調して、50～0% PWM デューティを実現します。

##### アップ・カウント・モードを使用して非対称 PWM を生成する場合：

- 0～100% 非対称 PWM を実現するには、以下の設定を使用します。TBPRD で CMPA/CMPB をロードします。ゼロ・アクションを使用して PWM をセットし、コンペアアップ・アクションを使用して PWM をクリアします。コンペア値を 0 から TBPRD+1 まで変調して、0～100% PWM デューティを実現します。

図 2-19 は、TBCTR のアップ・ダウン・カウント・モードを使用して対称 PWM 波形を生成する方法を示しています。このモードでは、波形のアップ・カウント部分およびダウン・カウント部分でのイコール・コンペア・マッチを使用して 0 ~ 100% DC 変調が実現されます。例では、CMPA を使用してコンペアを行っています。カウンタがインクリメントされている場合は、CMPA マッチによって PWM 出力が High になります。同様に、カウンタがデクリメントされている場合は、コンペア・マッチによって PWM 信号が Low になります。CMPA=0 の場合は、周期全体にわたって PWM 信号が Low であるため、0% デューティの波形が生成されます。CMPA=TBPRD の場合は、PWM 信号が High であるため、100% デューティが実現されます。

この設定の実際の使用時にゼロで CMPA/CMPB をロードする場合は、1 以上の CMPA/CMPB 値を使用します。周期で CMPA/CMPB をロードする場合は、TBPRD-1 以下の CMPA/CMPB 値を使用します。つまり、PWM 周期内に 1TBCLK サイクル以上のパルス（非常に短い場合はシステムによって無視される傾向がある）が常に存在することになります。

図 2-19 アップ・ダウン・カウント・モード対称波形

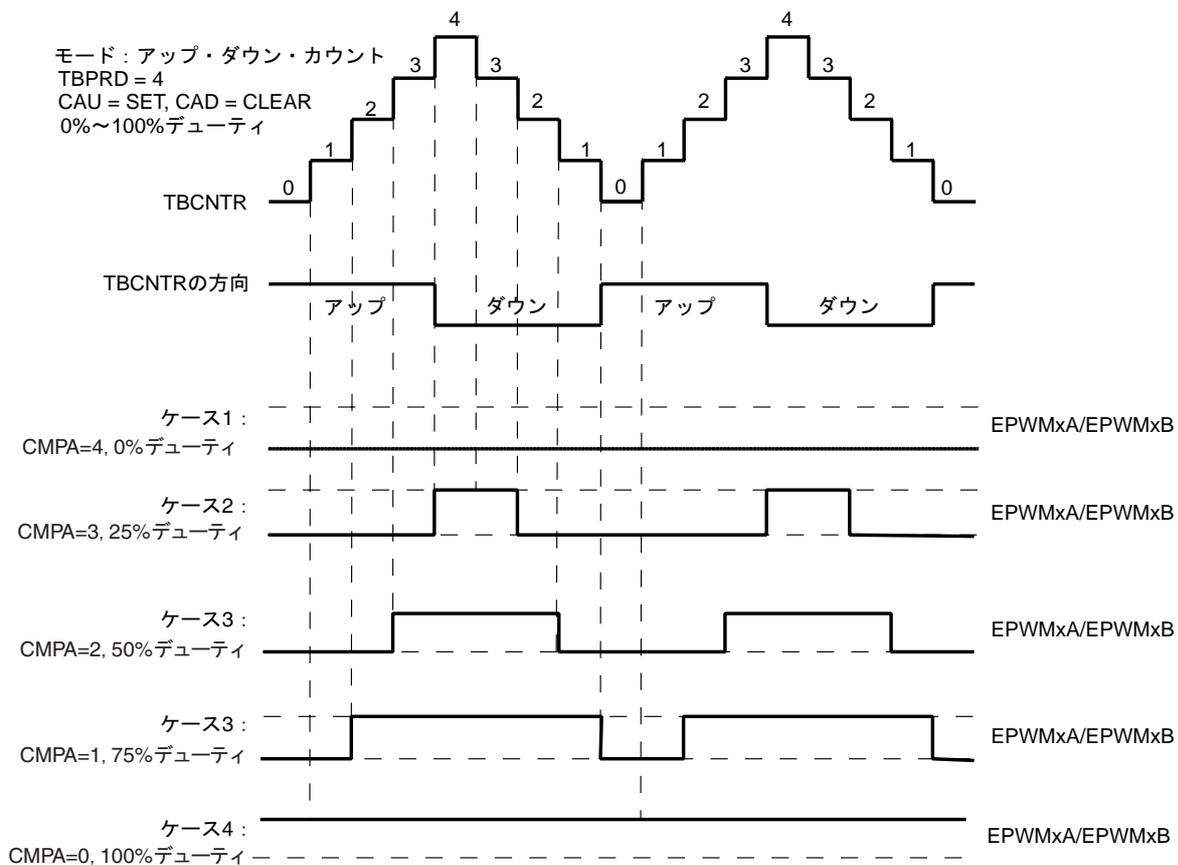
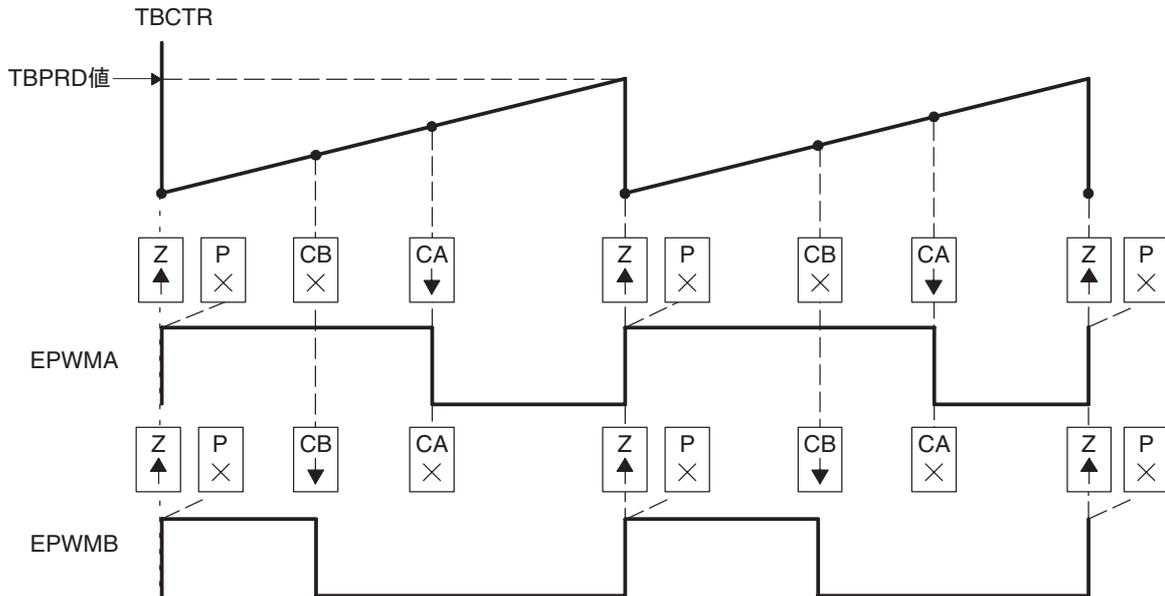


図 2-20 ~ 図 2-25 の PWM 波形は、いくつかの一般的なアクション選択設定を示しています。例 2-2 ~ 例 2-7 のサンプル C コードは、それぞれのケースについて ePWM モジュールの設定方法を示しています。図と例で使用している一部の表記規則は以下のとおりです。

- TBPRD、CMPA、および CMPB は、それぞれのレジスタでライトされた値を意味します。アクティブ・レジスタ（シャドウ・レジスタではない）は、ハードウェアによって使用されます。
- CMPx は、CMPA または CMPB のいずれかを意味します。
- EPWMxA と EPWMxB は、ePWMx からの出力信号を意味します。
- Up-Down はカウントアップ & ダウン・モード、Up はアップ・カウント・モード、Dwn はダウン・カウント・モードを意味します。
- Sym は対称、Asym は非対称を意味します。

図 2-20 アップ、シングル・エッジ非対称波形 (EPWMxA および EPWMxB での独立した変調  
- アクティブ・ハイ)



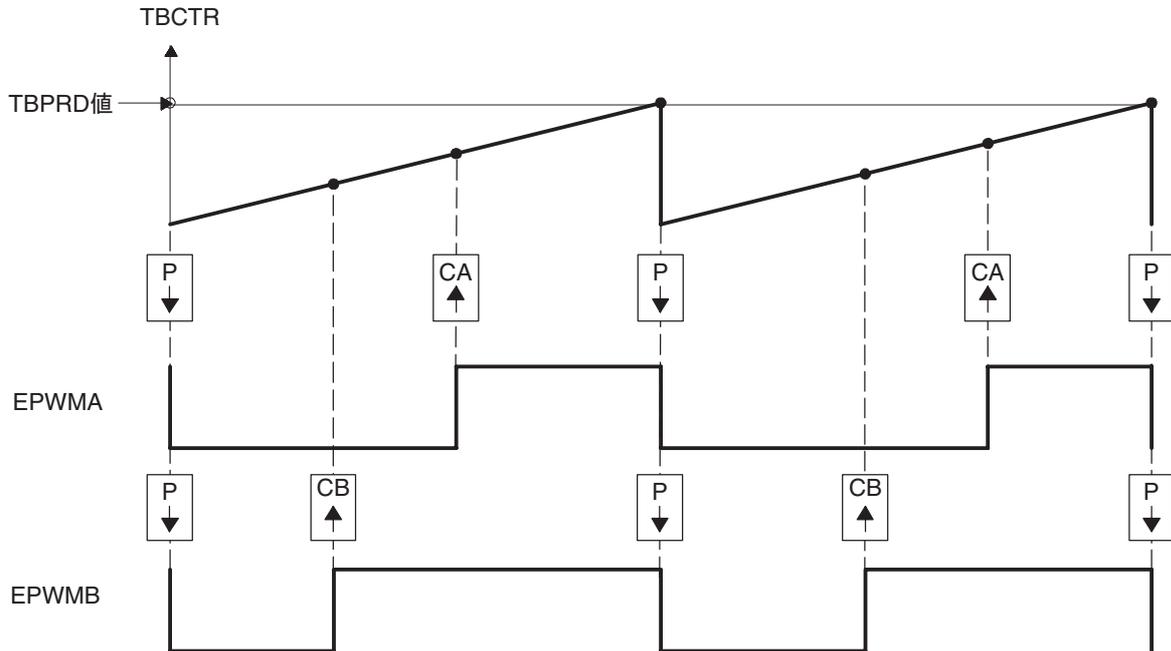
- A PWM 周期 = (TBPRD+1) × T<sub>TBCLK</sub>
- B EPWMxA のデューティ変調は、CMPA によって設定され、アクティブ・ハイです (つまり、CMPA に比例した High 時間デューティ)。
- C EPWMxB のデューティ変調は、CMPB によって設定され、アクティブ・ハイです (つまり、CMPB に比例した High 時間デューティ)。
- D 完全を期するために「何もしない」アクション (×) も示してありますが、後続の図では示していません。
- E ゼロと周期におけるアクションは、同時に発生しているように見えますが、実際には 1 TBCLK 周期のずれがあります。TBCTR は、周期から 0000 まで折り返します。

例 2-2 には、図 2-20 の波形の初期化および実行時を示すサンプル・コードが含まれています。

例 2-2 図 2-20 のサンプル・コード

```
// Initialization Time
// =====
EPwm1Regs.TBPRD = 600; // Period = 601 TBCLK counts
EPwm1Regs.CMPA.half.CMPA = 350; // Compare A = 350 TBCLK counts
EPwm1Regs.CMPB = 200; // Compare B = 200 TBCLK counts
EPwm1Regs.TBPHS = 0; // Set Phase register to zero
EPwm1Regs.TBCTR = 0; // clear TB counter
EPwm1Regs.TBCTL.bit.CTRMODE = TB_UP;
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Phase loading disabled
EPwm1Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCSEL = TB_SYNC_DISABLE;
EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1; // TBCLK = SYSCLK
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR = Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR = Zero
EPwm1Regs.AQCTLA.bit.ZRO = AQ_SET;
EPwm1Regs.AQCTLA.bit.CAU = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.ZRO = AQ_SET;
EPwm1Regs.AQCTLB.bit.CBU = AQ_CLEAR;
//
// Run Time
// =====
EPwm1Regs.CMPA.half.CMPA = Duty1A; // adjust duty for output EPWM1A
EPwm1Regs.CMPB = Duty1B; // adjust duty for output EPWM1B
```

図 2-21 アップ、シングル・エッジ非対称波形 (EPWMxA および EPWMxB での独立した変調  
— アクティブ・ロー)



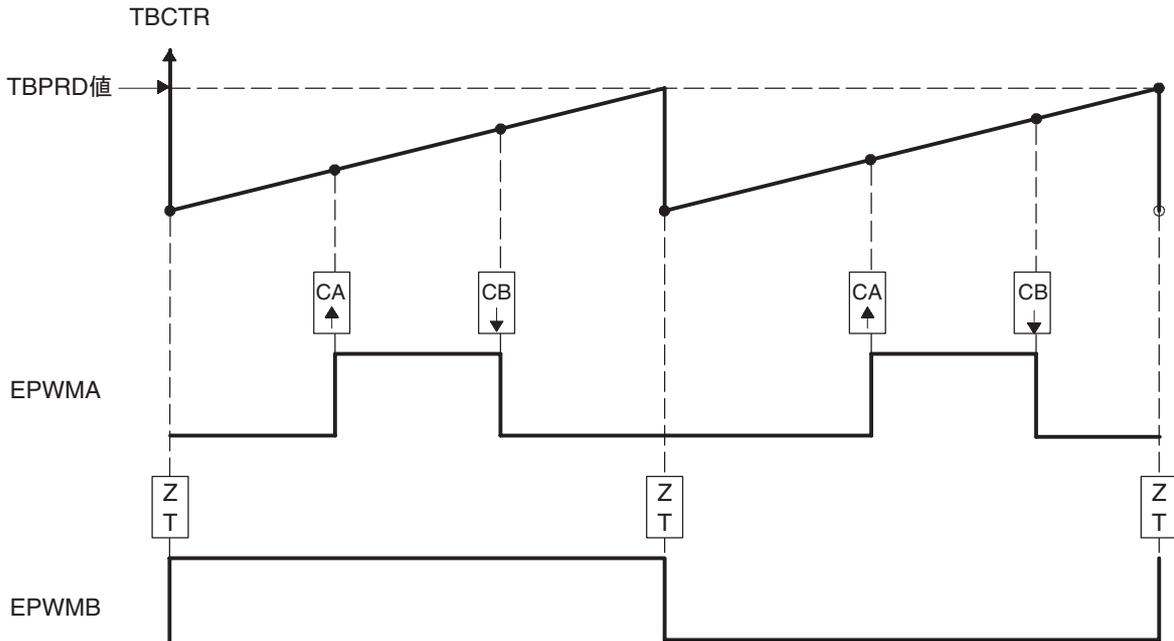
- A PWM 周期 = (TBPRD+1) × T<sub>TBCLK</sub>
- B EPWMxA のデューティ変調は、CMPA によって設定され、アクティブ・ローです (つまり、CMPA に比例した Low 時間デューティ)。
- C EPWMxB のデューティ変調は、CMPB によって設定され、アクティブ・ローです (つまり、CMPB に比例した Low 時間デューティ)。
- D ここでは完全を期すために「何もしない」アクション (×) も示してありますが、後続の図では示していません。
- E ゼロと周期におけるアクションは、同時に発生しているように見えますが、実際には 1 TBCLK 周期のずれがあります。TBCTR は、周期から 0000 まで折り返します。

例 2-3 には、図 2-21 の波形の初期化および実行時を示すサンプル・コードが含まれています。

### 例 2-3 図 2-21 のサンプル・コード

```
// Initialization Time
// = = = = =
EPwm1Regs.TBPRD = 600; // Period = 601 TBCLK counts
EPwm1Regs.CMPA.half.CMPA = 350; // Compare A = 350 TBCLK counts
EPwm1Regs.CMPB = 200; // Compare B = 200 TBCLK counts
EPwm1Regs.TBPHS = 0; // Set Phase register to zero
EPwm1Regs.TBCTR = 0; // clear TB counter
EPwm1Regs.TBCTL.bit.CTRMODE = TB_UP;
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Phase loading disabled
EPwm1Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCOSEL = TB_SYNC_DISABLE;
EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1; // TBCLK = SYSCLKOUT
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on TBCTR = Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on TBCTR = Zero
EPwm1Regs.AQCTLA.bit.PR = AQ_CLEAR;
EPwm1Regs.AQCTLA.bit.CAU = AQ_SET;
EPwm1Regs.AQCTLB.bit.PR = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.CBU = AQ_SET;
//
// Run Time
// = = = = =
EPwm1Regs.CMPA.half.CMPA = Duty1A; // adjust duty for output EPWM1A
EPwm1Regs.CMPB = Duty1B; // adjust duty for output EPWM1B
```

図 2-22 アップ・カウント、パルス配置非対称波形 (EPWMxA での独立した変調)



- A PWM 周波数 =  $1 / ((TBPRD+1) \times T_{TBCLK})$
- B パルスは、PWM サイクル (0000 ~ TBPRD) 内の任意の場所に配置できます。
- C (CMPB-CMPA) に比例した High 時間デューティ
- D EPWMxB を使用して、周波数が  $1/2 \times ((TBPRD + 1) \times TBCLK)$  の 50% デューティ矩形波を生成できます。

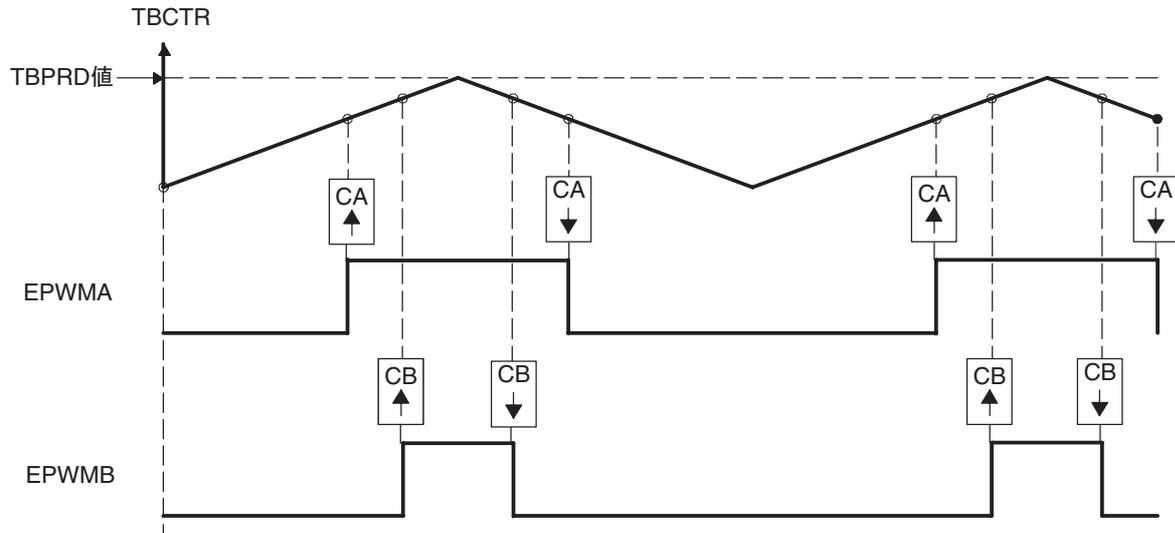
例 2-4 には、図 2-22 の波形の初期化および実行時を示すサンプル・コードが含まれています。ヘッダを定義するには、例 2-1 のコードを使用してください。

**例 2-4 図 2-22 のサンプル・コード**

```

// Initialization Time
// =====
EPwm1Regs.TBPRD = 600; // Period = 601 TBCLK counts
EPwm1Regs.CMPA.half.CMPA = 200; // Compare A = 200 TBCLK counts
EPwm1Regs.CMPB = 400; // Compare B = 400 TBCLK counts
EPwm1Regs.TBPHS = 0; // Set Phase register to zero
EPwm1Regs.TBCTR = 0; // clear TB counter
EPwm1Regs.TBCTL.bit.CTRMODE = TB_UP;
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Phase loading disabled
EPwm1Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCSEL = TB_SYNC_DISABLE;
EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1; // TBCLK = SYSCLKOUT
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on TBCTR = Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on TBCTR = Zero
EPwm1Regs.AQCTLA.bit.CAU = AQ_SET;
EPwm1Regs.AQCTLA.bit.CBU = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.ZRO = AQ_TOGGLE;
//
// Run Time
// =====
EPwm1Regs.CMPA.half.CMPA = EdgePosA; // adjust duty for output EPWM1A only
EPwm1Regs.CMPB = EdgePosB;
  
```

図 2-23 アップ・ダウン・カウンタ、デュアル・エッジ対称波形 (EPWMxA および EPWMxB での独立した変調 - アクティブ・ロー)



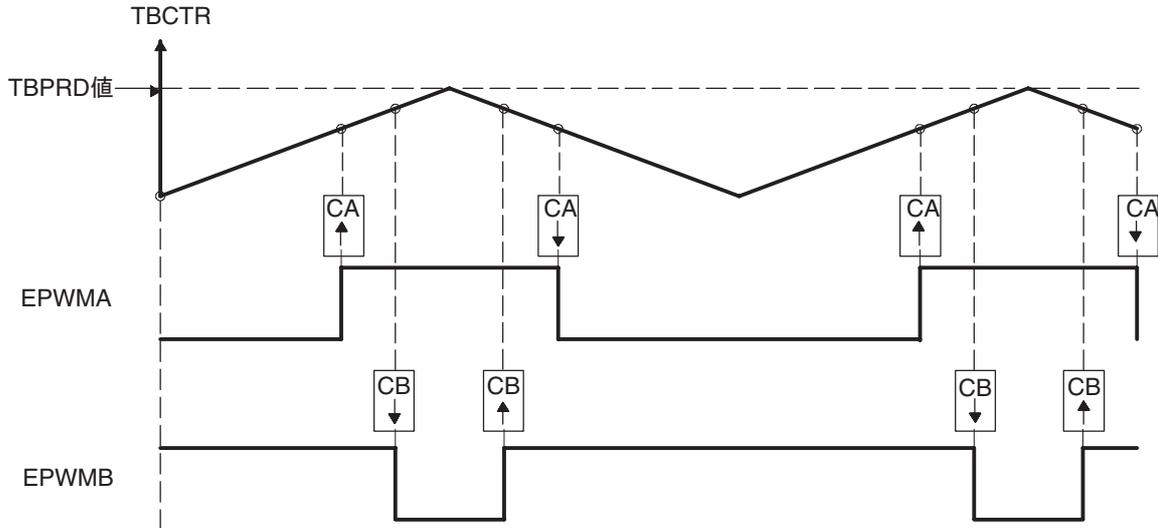
- A PWM 周期 =  $2 \times TBPRD \times T_{TBCLK}$
- B EPWMxA のデューティ変調は、CMPA によって設定され、アクティブ・ローです (つまり、CMPA に比例した Low 時間デューティ)。
- C EPWMxB のデューティ変調は、CMPB によって設定され、アクティブ・ローです (つまり、CMPB に比例した Low 時間デューティ)。
- D 出力の EPWMxA と EPWMxB は、独立したパワー・スイッチをドライブできます。

例 2-5 には、図 2-23 の波形の初期化および実行時を示すサンプル・コードが含まれています。ヘッダを定義するには、例 2-1 のコードを使用してください。

例 2-5 図 2-23 のサンプル・コード

```
// Initialization Time
// =====
EPwm1Regs.TBPRD = 600; // Period = 2 L600 TBCLK counts
EPwm1Regs.CMPA.half.CMPA = 400; // Compare A = 400 TBCLK counts
EPwm1Regs.CMPB = 500; // Compare B = 500 TBCLK counts
EPwm1Regs.TBPHS = 0; // Set Phase register to zero
EPwm1Regs.TBCNT = 0; // clear TB counter
EPwm1Regs.TBCTL.bit.CTRMODE = TB_UPDOWN; // Symmetric
xEPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Phase loading disabled
EPwm1Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCOSEL = TB_SYNC_DISABLE;
EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1; // TBCLK = SYSCLKOUT
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR = Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR = Zero
EPwm1Regs.AQCTLA.bit.CAU = AQ_SET;
EPwm1Regs.AQCTLA.bit.CAD = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.CBU = AQ_SET;
EPwm1Regs.AQCTLB.bit.CBD = AQ_CLEAR;
//
// Run Time
// =====
EPwm1Regs.CMPA.half.CMPA = Duty1A; // adjust duty for output EPWM1A
EPwm1Regs.CMPB = Duty1B; // adjust duty for output EPWM1B
```

図 2-24 アップ・ダウン・カウント、デュアル・エッジ対称波形 (EPWMxA および EPWMxB での独立した変調 - 相補)



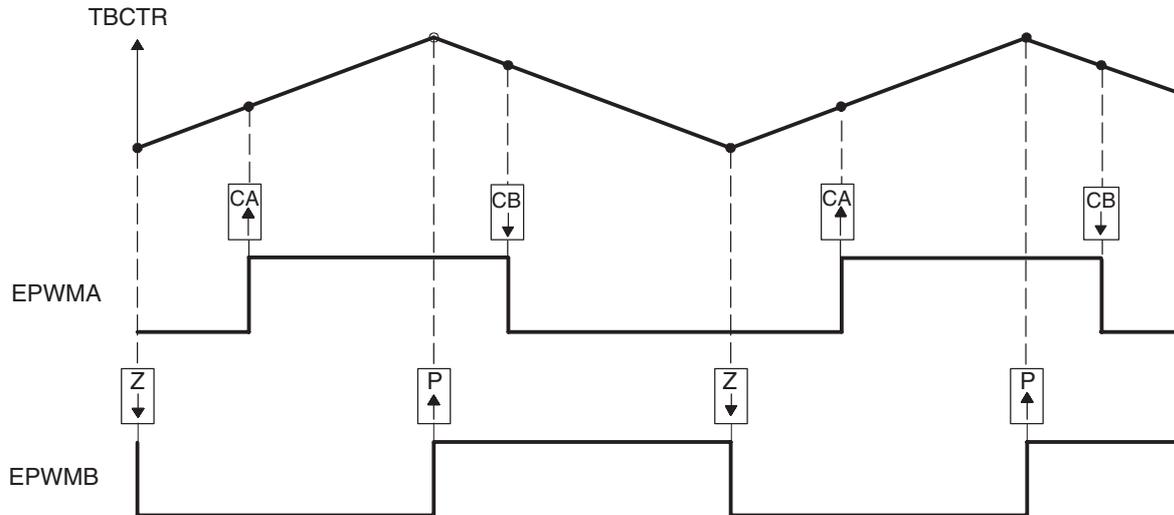
- A PWM 周期 =  $2 \times \text{TBPRD} \times T_{\text{TBCLK}}$
- B EPWMxA のデューティ変調は、CMPA によって設定され、アクティブ・ローです (つまり、CMPA に比例した Low 時間デューティ)。
- C EPWMxB のデューティ変調は、CMPB によって設定され、アクティブ・ハイです (つまり、CMPB に比例した High 時間デューティ)。
- D EPWMx 出力は、上位 / 下位 (相補) パワー・スイッチをドライブできます。
- E デッドバンド =  $\text{CMPB} - \text{CMPA}$  (ソフトウェアでフルにプログラムできるエッジ配置)。従来のエッジ遅延方法が必要な場合にもデッドバンド・モジュールを利用することに注意してください。

例 2-6 には、図 2-24 の波形の初期化および実行時を示すサンプル・コードが含まれています。ヘッダを定義するには、例 2-1 のコードを使用してください。

**例 2-6 図 2-24 のサンプル・コード**

```
// Initialization Time
// =====
EPwm1Regs.TBPRD = 600; // Period = 2 L600 TBCLK counts
EPwm1Regs.CMPA.half.CMPA = 350; // Compare A = 350 TBCLK counts
EPwm1Regs.CMPB = 400; // Compare B = 400 TBCLK counts
EPwm1Regs.TBPHS = 0; // Set Phase register to zero
EPwm1Regs.TBCNT = 0; // clear TB counter
EPwm1Regs.TBCTL.bit.CTRMODE = TB_UPDOWN; // Symmetric
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Phase loading disabled
EPwm1Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCSEL = TB_SYNC_DISABLE;
EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1; // TBCLK = SYSCLKOUT
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR = Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR = Zero
EPwm1Regs.AQCTLA.bit.CAU = AQ_SET;
EPwm1Regs.AQCTLA.bit.CAD = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.CBU = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.CBD = AQ_SET;
// Run Time
// =====
EPwm1Regs.CMPA.half.CMPA = Duty1A; // adjust duty for output EPWM1A
EPwm1Regs.CMPB = Duty1B; // adjust duty for output EPWM1B
```

図 2-25 アップ・ダウン・カウント、デュアル・エッジ非対称波形 (EPWMxA での独立した変調 - アクティブ・ロー)



- A PWM 周期 =  $2 \times \text{TBPRD} \times \text{TBCLK}$
- B 立ち上がりエッジと立ち下がりエッジは、PWM サイクル内で非対称に配置できます。これにより、パルス配置手法が可能になります。
- C EPWMxA のデューティ変調は、CMPA と CMPB によって設定されます。
- D  $(\text{CMPA} + \text{CMPB})$  に比例した EPWMxA の Low 時間デューティ
- E この例をアクティブ・ハイに変更するには、CMPA および CMPB アクションを反転する必要があります (つまり、セット・クリアとクリア・セット)。
- F EPWMxB のデューティ変調は 50% で固定されます (EPWMxB の予備アクション・リソースを利用する)。

例 2-7 には、図 2-25 の波形の初期化および実行時を示すサンプル・コードが含まれています。ヘッダを定義するには、例 2-1 のコードを使用してください。

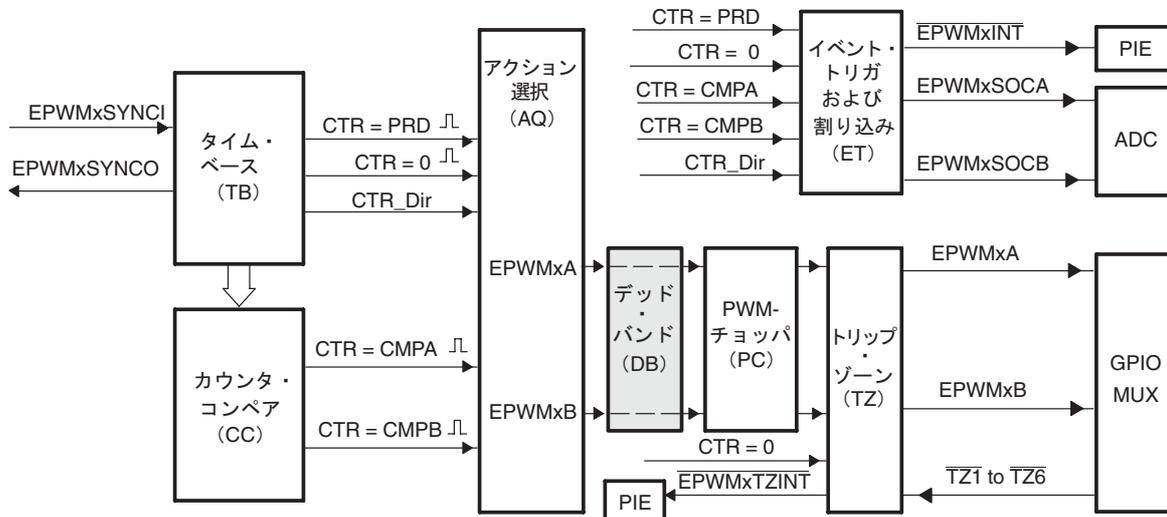
例 2-7 図 2-25 のサンプル・コード

```
// Initialization Time
// =====
EPwm1Regs.TBPRD = 600; // Period = 2 L 600 TBCLK counts
EPwm1Regs.CMPA.half.CMPA = 250; // Compare A = 250 TBCLK counts
EPwm1Regs.CMPB = 450; // Compare B = 450 TBCLK counts
EPwm1Regs.TBPHS = 0; // Set Phase register to zero
EPwm1Regs.TBCNT = 0; // clear TB counter
EPwm1Regs.TBCTL.bit.CTRMODE = TB_UPDOWN; // Symmetric
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Phase loading disabled
EPwm1Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCOSEL = TB_SYNC_DISABLE;
EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1; // TBCLK = SYSCLKOUT
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR = Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR = Zero
EPwm1Regs.AQCTLA.bit.CAU = AQ_SET;
EPwm1Regs.AQCTLA.bit.CBD = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.ZRO = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.PRD = AQ_SET;
// Run Time
// =====
EPwm1Regs.CMPA.half.CMPA = EdgePosA; // adjust duty for output EPWM1A only
EPwm1Regs.CMPB = EdgePosB;
```

## 2.5 デッドバンド・ジェネレータ (DB) サブモジュール

図 2-26 は、ePWM モジュール内のデッドバンド・サブモジュールを表します。

図 2-26 デッドバンド・サブモジュール



### 2.5.1 デッドバンド・サブモジュールの目的

「アクション選択 (AQ) モジュール」セクションでは、ePWM モジュールの CMPA および CMPB リソースをフル制御することにより必要なデッドバンドを生成する方法について説明しました。ただし、従来のエッジ遅延ベースの極性制御付きデッドバンドが必要な場合は、ここで説明するデッドバンド・サブモジュールを使用してください。

デッドバンド・モジュールの主要機能は以下のとおりです。

- シングル EPWMxA 入力からのデッドバンド関係を使用した適切な信号ペア (EPWMxA と EPWMxB) の生成
- 以下に対する信号ペアのプログラミング
  - アクティブ・ハイ (AH)
  - アクティブ・ロー (AL)
  - アクティブ・ハイ相補 (AHC)
  - アクティブ・ロー相補 (ALC)
- 立ち上がりエッジ (RED) へのプログラマブル遅延の追加
- 立ち下がりエッジ (FED) へのプログラマブル遅延の追加
- 完全にバイパスできる (図の点線に注目)

### 2.5.2 デッドバンド・サブモジュールの制御と監視

デッドバンド・サブモジュールの動作は、以下のレジスタによって制御および監視されます。

表 2-12 デッドバンド・ジェネレータ・サブモジュールのレジスタ

レジスタ名	アドレス・オフセット	シャドウ化	説明
DBCTL	0x000F	無	デッドバンド制御レジスタ
DBRED	0x0010	無	デッドバンド立ち上がりエッジ遅延カウント・レジスタ
DBFED	0x0011	無	デッドバンド立ち下がりエッジ遅延カウント・レジスタ

### 2.5.3 デッドバンド・サブモジュールの動作上の重要点

以下のセクションでは、動作上の重要点について説明します。

図 2-27 に示すように、デッドバンド・サブモジュールには独立した選択オプションのグループが 2 つあります。

• **入力ソース選択：**

デッドバンド・モジュールへの入力信号は、アクション選択からの EPWMxA および EPWMxB 出力信号です。このセクションでは、それらの入力を EPWMxA In および EPWMxB In と呼びます。DBCTL[IN\_MODE] 制御ビットを使用すると、各遅延、立ち下がりエッジ、または立ち上がりエッジの信号ソースを選択できます。

- EPWMxA In は立ち下がりエッジ遅延と立ち上がりエッジ遅延の両方に対するソースである (デフォルトのモード)
- EPWMxA In は立ち下がりエッジ遅延のソース、EPWMxB In は立ち上がりエッジ遅延のソースである
- EPWMxA In は立ち上がりエッジ遅延のソース、EPWMxB In は立ち下がりエッジ遅延のソースである
- EPWMxB は立ち下がりエッジ遅延と立ち上がりエッジ遅延の両方に対するソースである

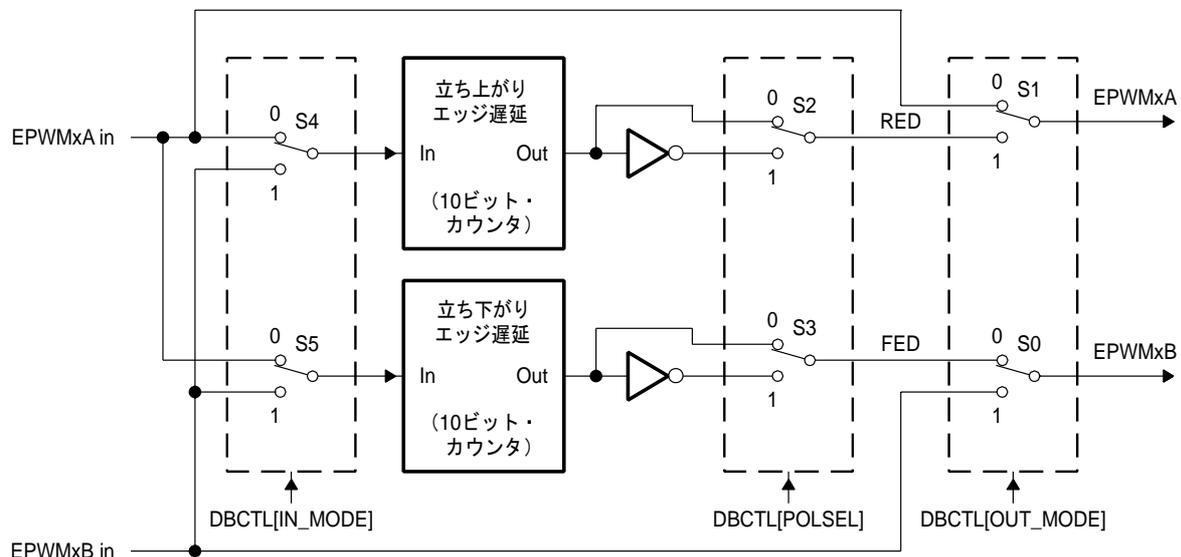
• **出力モードの制御：**

出力モードは、DBCTL[OUT\_MODE] ビットによって設定されます。これらのビットでは、入力信号に立ち下がりエッジ遅延を適用するか、立ち上がりエッジ遅延を適用するか、その両方を適用するか、またはどちらも適用しないかが決定されます。

• **極性の制御：**

極性の制御 (DBCTL[POLSEL]) では、デッドバンド・サブモジュールから送信する前に立ち上がりエッジ遅延信号または立ち下がりエッジ遅延信号 (あるいはその両方) を反転するかどうかを指定できます。

図 2-27 デッドバンド・サブモジュールの設定オプション



すべての組み合わせがサポートされていますが、そのすべてが一般的な使用モードというわけではありません。表 2-13 に、従来からよく用いられるデッドバンド設定をいくつか示します。これらのモードは、EPWMxA In が立ち下がりエッジ遅延と立ち上がりエッジ遅延の両方のソースになるように DBCTL[IN\_MODE] が設定されていることを前提としています。入力信号ソースを変更すると、高度なモードや従来にはないモードを実現できます。表 2-13 に示したモードは、以下のカテゴリに分類されます。

- **モード 1：立ち下がりエッジ遅延 (FED) と立ち上がりエッジ遅延 (RED) の両方をバイパス**  
デッドバンド・サブモジュールを PWM 信号パスから完全にディスエーブルにすることができます。
- **モード 2 ~ 5：従来からよく用いられるデッドバンド極性設定：**  
これらは、産業用パワー・スイッチ・ゲート・ドライバによって要求されるすべてのアクティブ・ハイ / ロー・モードに対処する一般的な極性設定を表します。これらの一般的なケースの波形を図 2-28 に示します。図 2-28 と同じ波形を生成するには、EPWMxA について示したように、アクション選択サブモジュールを設定して信号を生成する必要があります。
- **モード 6：立ち上がりエッジ遅延をバイパス、モード 7：立ち下がりエッジ遅延をバイパス**  
表 2-13 の最後の 2 つは、立ち下がりエッジ遅延 (FED) または立ち上がりエッジ遅延 (RED) ブロックがバイパスされる組み合わせを示しています。

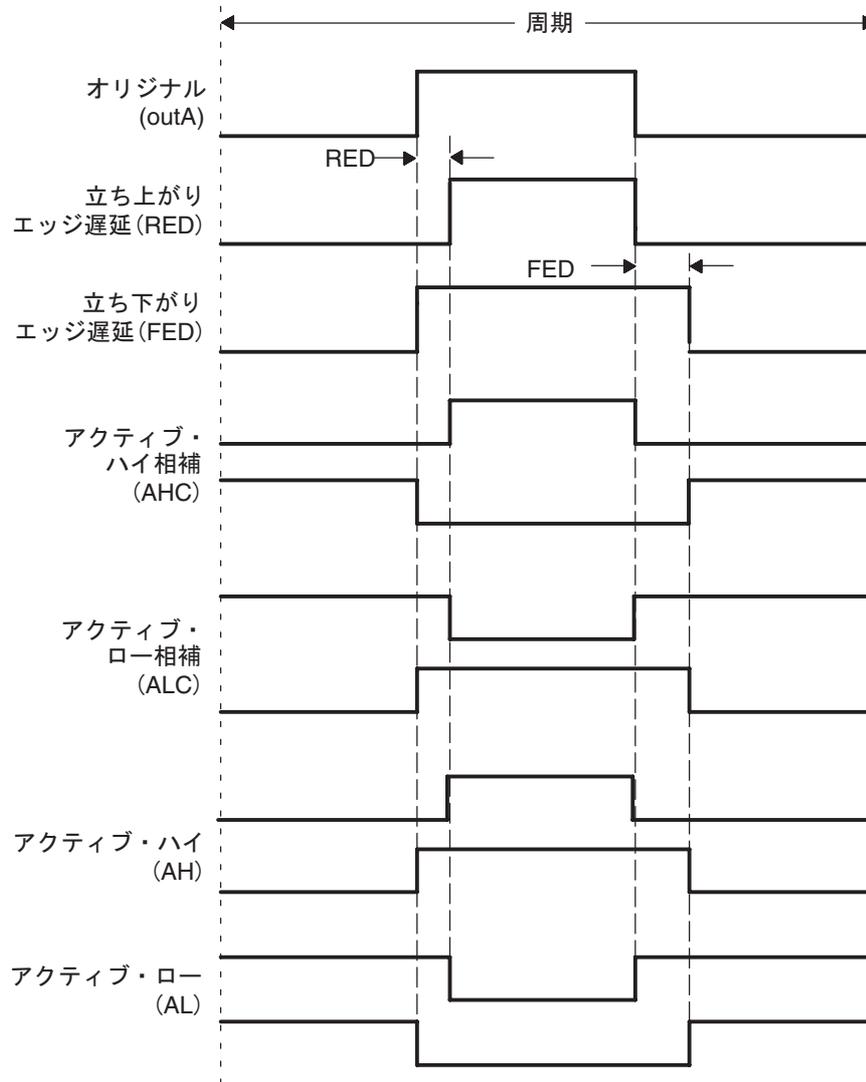
表 2-13 従来のデッドバンド動作モード

モード	モードの説明 (1)	DBCTL[POLSEL]		DBCTL[OUT_MODE]	
		S3	S2	S1	S0
1	EPWMxA と EPWMxB が渡される (遅延なし)	X	X	0	0
2	アクティブ・ハイ相補 (AHC)	1	0	1	1
3	アクティブ・ロー相補 (ALC)	0	1	1	1
4	アクティブ・ハイ (AH)	0	0	1	1
5	アクティブ・ロー (AL)	1	1	1	1
6	EPWMxA Out=EPWMxA In (遅延なし) EPWMxB Out=EPWMxA In (立ち下がりエッジ遅延あり)	0 または 1	0 または 1	0	1
7	EPWMxA Out=EPWMxA In (立ち上がりエッジ遅延あり) EPWMxB Out=EPWMxB In (遅延なし)	0 または 1	0 または 1	1	0

(1) これらは従来からよく用いられるデッドバンド・モードであり、DBCTL[IN\_MODE]=0,0 であることを前提としています。つまり、EPWMxA in は立ち下がりエッジ遅延と立ち上がりエッジ遅延の両方に対するソースになります。IN\_MODE 設定を変更すると、高度な従来にはないモードを実現できます。

図 2-28 は、 $0\% < \text{デューティ} < 100\%$  である一般的なケースの波形を示しています。

図 2-28 一般的なケースのデッドバンド波形 ( $0\% < \text{デューティ} < 100\%$ )



デッドバンド・サブモジュールは、立ち上がりエッジ (RED) 遅延と立ち下がりエッジ (FED) 遅延について独立した値をサポートしています。遅延の量は、DBRED および DBFED レジスタを使用してプログラムされます。これらは 10 ビットのレジスタであり、その値は信号エッジの遅延量となるタイム・ベース・クロック (TBCLK) 周期の数を表します。たとえば、立ち下がりエッジ遅延と立ち上がりエッジ遅延を計算する式は以下のようになります。

$$FED = DBFED \times T_{TBCLK}$$

$$RED = DBRED \times T_{TBCLK}$$

TBCLK は TBCLK (SYSCLKOUT がプリスケールされたクロック) の周期です。

便宜上、さまざまな TBCLK オプションの遅延値を表 2-14 に示します。

表 2-14 DBFED および DBRED の機能としてのデッドバンド遅延値 (μS 単位)

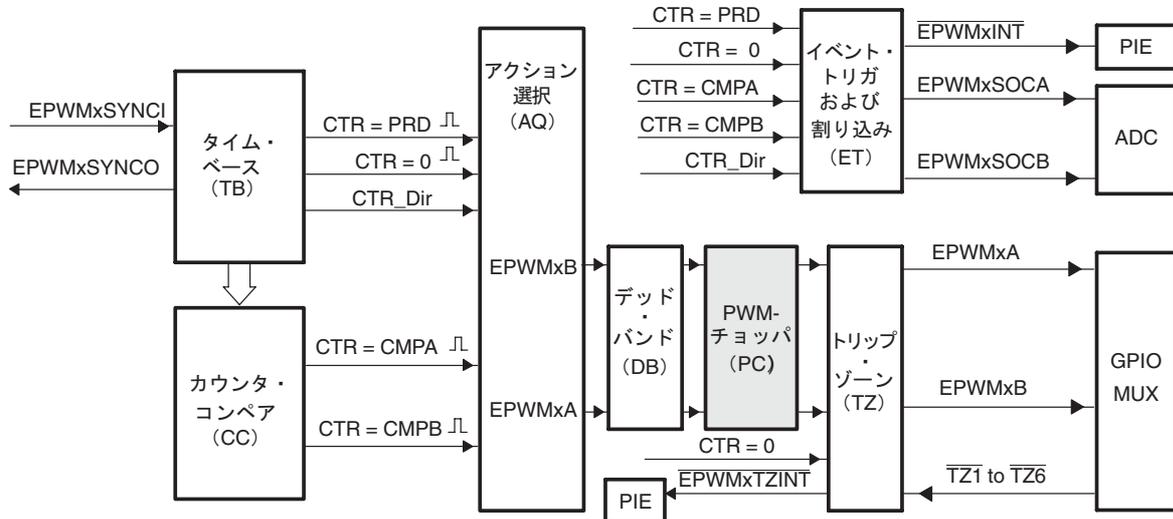
デッドバンド値	デッドバンド遅延 (μS) <sup>(1)</sup>			
	DBFED、DBRED	TBCLK=SYSCLKOUT/1	TBCLK=SYSCLKOUT/2	TBCLK=SYSCLKOUT/4
1		0.01μS	0.02μS	0.04μS
5		0.05μS	0.10μS	0.20μS
10		0.10μS	0.20μS	0.40μS
100		1.00μS	2.00μS	4.00μS
200		2.00μS	4.00μS	8.00μS
300		3.00μS	6.00μS	12.00μS
400		4.00μS	8.00μS	16.00μS
500		5.00μS	10.00μS	20.00μS
600		6.00μS	12.00μS	24.00μS
700		7.00μS	14.00μS	28.00μS
800		8.00μS	16.00μS	32.00μS
900		9.00μS	18.00μS	36.00μS
1000		10.00μS	20.00μS	40.00μS

(1) 表の値は、SYSCLKOUT=100MHz に基づいて計算されています。

## 2.6 PWM チョッパ (PC) サブモジュール

図 2-29 は、ePWM モジュール内の PWM チョッパ (PC) サブモジュールを表します。

図 2-29 PWM チョッパ・サブモジュール



PWM チョッパ・サブモジュールでは、アクション選択およびデッドバンド・サブモジュールによって生成された PWM 波形を高周波キャリア信号で変調することができます。パワー・スイッチング部品を制御するためにパルス・トランスフォーマーベースのゲート・ドライバが必要な場合は、この機能が重要になります。

### 2.6.1 PWM チョッパ・サブモジュールの目的

PWM チョッパ・サブモジュールの主要機能は以下のとおりです。

- プログラマブルなチョッピング（キャリア）周波数
- 最初のパルスのプログラマブル・パルス幅
- 2 番目以降のパルスのプログラマブル・デューティ・サイクル
- 不要な場合は完全にバイパス可能

### 2.6.2 PWM チョッパ・サブモジュールの制御

PWM チョッパ・サブモジュールの動作は、表 2-15 のレジスタによって制御されます。

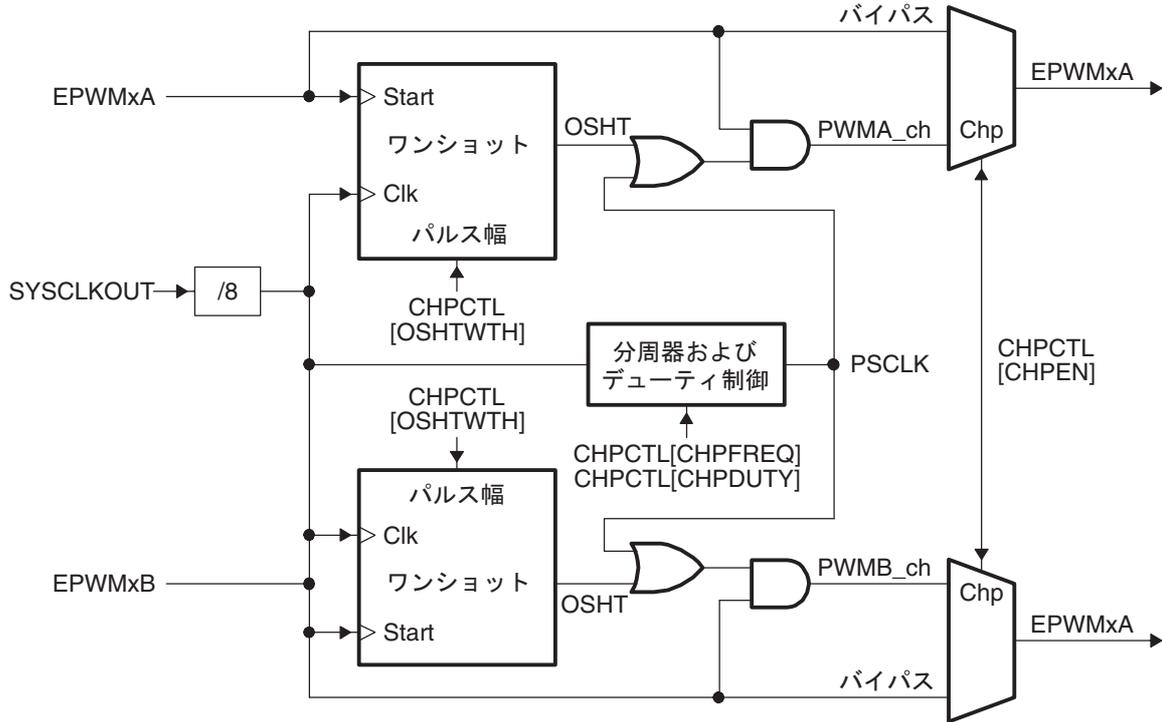
表 2-15 PWM チョッパ・サブモジュールのレジスタ

ニーモニック	アドレス・オフセット	シャドウ化	説明
PCCTL	0x001E	無	PWM チョッパ制御レジスタ

### 2.6.3 PWM チョッパ・サブモジュールの動作上の重要点

図 2-30 は、PWM チョッパ・サブモジュールの動作詳細を示しています。キャリア・クロックは、SYSCLKOUT から派生します。周波数とデューティ・サイクルは、CHPCTL レジスタの CHPFREQ および CHPDUTY ビットによって制御されます。ワンショット・ブロックは、パワー・スイッチが確実にオンになるように最初のパルスに高エネルギーを供給する機能になり、後続のパルスはパルスを維持しつつ、パワー・スイッチはオンのままになります。ワンショット幅は、OSHTWTH ビットでプログラムされます。CHPEN ビットを使用すると、PWM チョッパ・サブモジュールを完全にディスエーブルにする（バイパスする）ことができます。

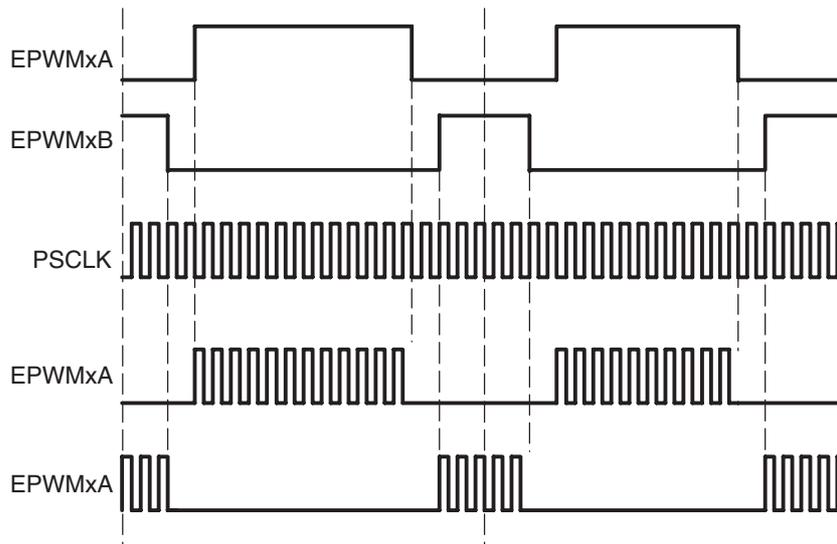
図 2-30 PWM チョッパ・サブモジュールの動作詳細



#### 2.6.4 波形

図 2-31 では、チョッピング・アクションだけの簡略波形を示しており、ワンショットとデューティ・サイクルの制御は示していません。ワンショットおよびデューティ・サイクル制御の詳細については、後続のセクションで説明します。

図 2-31 チョッピング・アクションだけを示している単純な PWM チョッパ・サブモジュール波形



### 2.6.4.1 ワンショット・パルス

最初のパルスの幅は、16 のパルス幅値のいずれかにプログラムすることができます。最初のパルスの幅または周期は、以下の式で求められます。

$$T_{1stpulse} = T_{SYSCLKOUT} \times 8 \times OSHTWTH$$

ここで、 $T_{SYSCLKOUT}$  はシステム・クロック (SYSCLKOUT) の周期、OSHTWTH は 4 つの制御ビット (値は 1 ~ 16) です。

図 2-32 は、最初のパルスとその後に続くパルスを示しています。表 2-16 は、SYSCLKOUT=100MHz の場合の可能なパルス幅値を示しています。

図 2-32 最初のパルスとその後に続くパルスを示している PWM チョップパ・サブモジュール波形

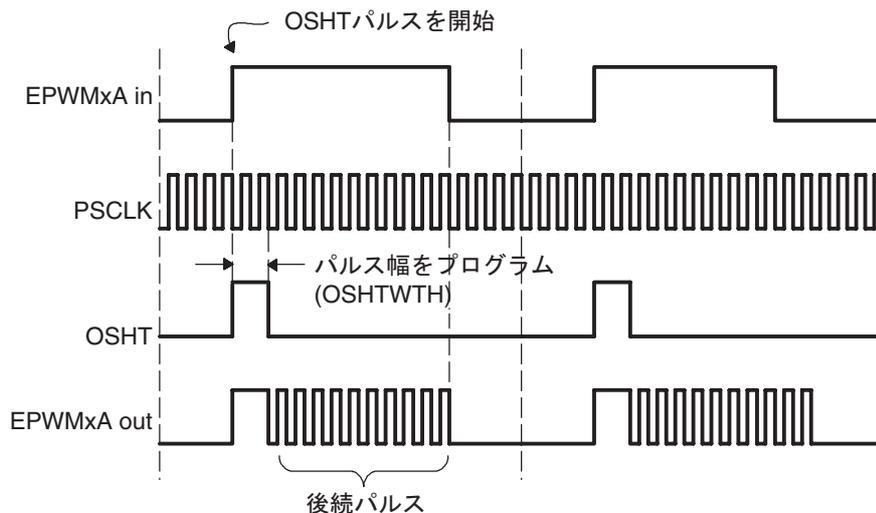


表 2-16 SYSCLKOUT=100 MHz の場合の可能なパルス幅値

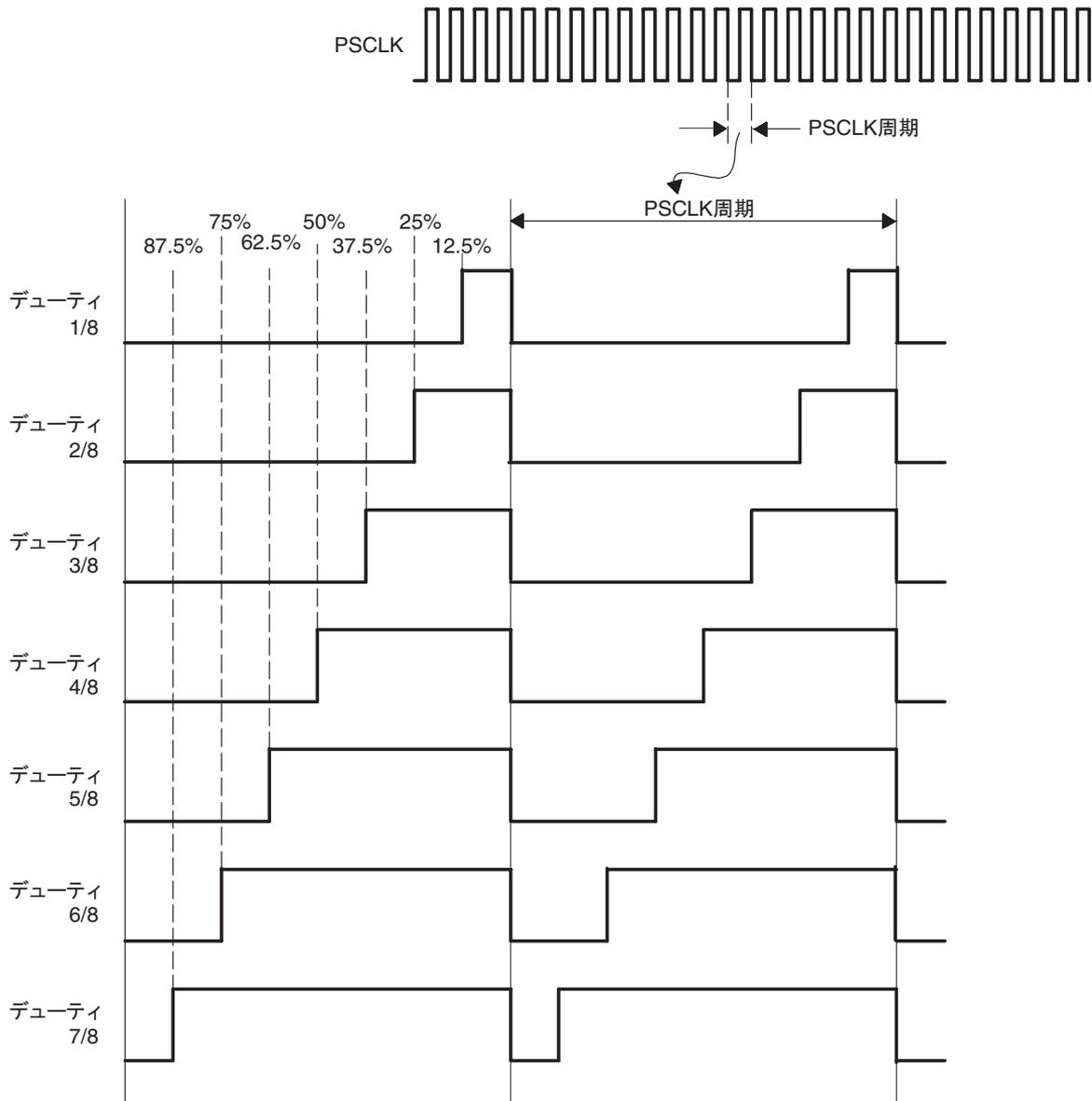
OSHTWTHz (hex)	パルス幅 (nS)
0	80
1	160
2	240
3	320
4	400
5	480
6	560
7	640
8	720
9	800
A	880
B	960
C	1040
D	1120
E	1200
F	1280

### 2.6.4.2 デューティ・サイクル制御

パルス・トランスフォーマベースのゲート・ドライブ設計では、トランスフォーマと関連回路の磁気特性を理解する必要があります。飽和は、そのような考慮事項の 1 つです。ゲート・ドライブ設計者を支援するために、2 番目以降のパルスのデューティ・サイクルはプログラムできるようになっています。これらの後続パルスにより、オン周期中は正しいドライブの強さと極性がパワー・スイッチ・ゲートで維持されます。したがって、プログラマブル・デューティ・サイクルは、ソフトウェア制御による設計の調整や最適化を可能にします。

図 2-33 は、CHPDUTY ビットのプログラミングによって実現できるデューティ・サイクル制御を示しています。7 つのデューティ比率 (12.5 ~ 87.5%) のいずれかを選択できます。

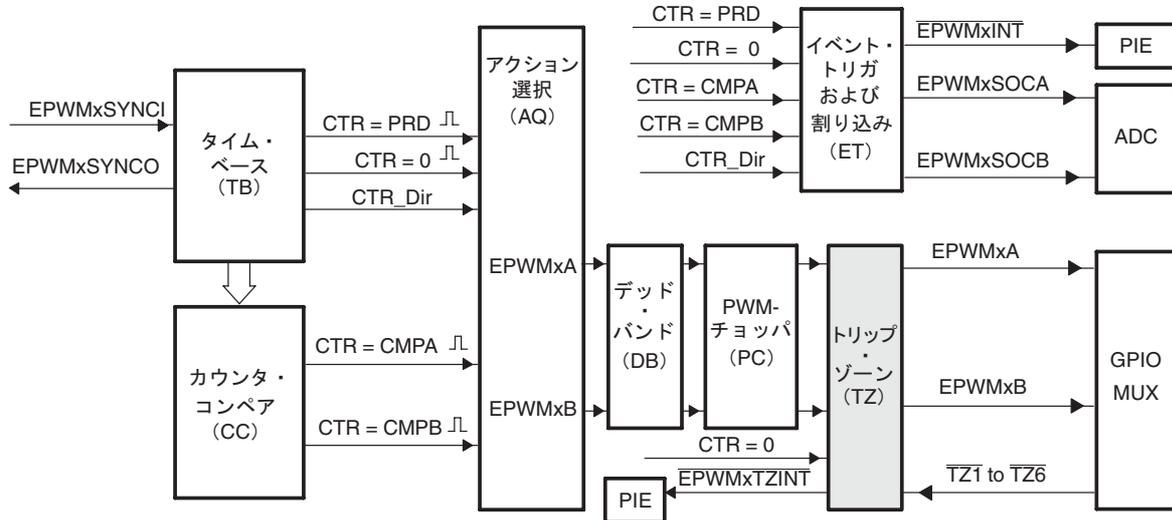
図 2-33 後続パルスのパルス幅 (デューティ・サイクル) 制御を示している  
 PWM チョップパ・サブモジュール波形



## 2.7 トリップ・ゾーン (TZ) サブモジュール

図 2-34 は、トリップ・ゾーン (TZ) サブモジュールがどのように ePWM モジュール内で動作するのかを示しています。

図 2-34 トリップ・ゾーン・サブモジュール



各 ePWM モジュールは、GPIO MUX から供給される 6 つの  $\overline{TZn}$  信号 ( $\overline{TZ1} \sim \overline{TZ6}$ ) に接続されます。これらの信号は、外部フェールまたはトリップ状態を示します。ePWM 出力は、それに応じてフェール発生時に応答するようにプログラムすることができます。

### 2.7.1 トリップ・ゾーン・サブモジュールの目的

トリップ・ゾーン・サブモジュールの主要機能は以下のとおりです。

- トリップ入力  $\overline{TZ1} \sim \overline{TZ6}$  は、任意の ePWM モジュールに柔軟にマッピングすることができます。
- フェール状態の発生時には、EPWMxA および EPWMxB 出力を強制的に以下のいずれかにすることができます。
  - High
  - Low
  - ハイ・インピーダンス
  - アクションなし
- 重大な短絡または過電流状態に対するワンショット・トリップ (OSHT) をサポートします。
- 電流リミット動作に対するサイクル・バイ・サイクル・トリップ (CBC) をサポートします。
- 各トリップ・ゾーン入力ピンは、ワンショットまたはサイクル・バイ・サイクル動作に割り当てることができます。
- 割り込み生成は、どのトリップ・ゾーン・ピンでも可能です。
- ソフトウェア強制トリップもサポートしています。
- 不要な場合は、トリップ・ゾーン・サブモジュールを完全にバイパスできます。

## 2.7.2 トリップ・ゾーン・サブモジュールの制御と監視

トリップ・ゾーン・サブモジュールの動作は、以下のレジスタによって制御および監視されます。

表 2-17 トリップ・ゾーン・サブモジュールのレジスタ

レジスタ名	アドレス・オフセット	シャドウ化	説明
TZSEL	0x0012	無	トリップ・ゾーン・セレクト・レジスタ
予約	0x0013		
TZCTL	0x0014	無	トリップ・ゾーン制御レジスタ
TZEINT	0x0015	無	トリップ・ゾーン・イネーブル割り込みレジスタ
TZFLG	0x0016	無	トリップ・ゾーン・フラグ・レジスタ
TZCLR	0x0017	無	トリップ・ゾーン・クリア・レジスタ
TZFRC	0x0018	無	トリップ・ゾーン強制レジスタ

(1) トリップ・ゾーン・レジスタはすべて EALLOW 保護されており、EALLOW 命令の実行後のみ変更できます。詳細については、『TMS320x280x, 2801x, 2804x System Control and Interrupts Reference Guide』(SPRU712) を参照してください。

## 2.7.3 トリップ・ゾーン・サブモジュールの動作上の重要点

以下の各セクションでは、トリップ・ゾーン・サブモジュールの動作上の重要点と設定オプションについて説明します。

$\overline{TZ1}$  ~  $\overline{TZ6}$  ピン (まとめてに  $\overline{TZn}$  とも呼ばれる) のトリップ・ゾーン信号は、アクティブ・ロー入力信号です。

これらのピンのいずれかが Low になった場合、それはトリップ・イベントが発生したことを意味します。各 ePWM モジュールは、各トリップ・ゾーン・ピンを無視または使用するよう個別に設定できます。どのトリップ・ゾーン・ピンが特定の ePWM モジュールによって使用されるのかは、その特定の ePWM モジュールの TZSEL レジスタによって決定されます。トリップ・ゾーン信号は、システム・クロック (SYSCLKOUT) と同期化され、GPIO MUX ブロック内でデジタル的にフィルタリングされることもあります。ePWM モジュール内のフェール状態をトリガするには、 $\overline{TZn}$  入力の SYSCLKOUT Low パルスが最低 1 つあれば十分です。非同期トリップを使用すると、何らかの理由でクロックが欠けている場合でも GPIO が適切に設定されていれば、 $\overline{TZn}$  入力に有効なイベントが起これば引き続き出力をトリップすることができます。『TMS320x280x, 2801x, 2804x System Control and Interrupts Reference Guide』の GPIO のセクションを参照してください。

各  $\overline{TZn}$  入力は、サイクル・バイ・サイクルまたはワンショット・トリップ・イベントを ePWM モジュールに供給するよう個別に設定できます。

この設定は、TZSEL[CBCn] および TZSEL[OSHTn] 制御ビット (n はトリップ・ピンに対応する) によって決まります。

- **サイクル・バイ・サイクル (CBC) :**

サイクル・バイ・サイクル・トリップ・イベントが発生すると、TZCTL レジスタで指定したアクションが EPWMxA または EPWMxB 出力 (あるいはその両方) ですぐ実行されます。表 2-18 に可能なアクションを示します。また、サイクル・バイ・サイクル・トリップ・イベント・フラグ (TZFLG[CBC]) がセットされ、EPWMx\_TZINT 割り込みが生成されます (TZEINT レジスタと PIE ペリフェラルでイネーブルになっている場合)。

トリップ・イベントが存在しなくなった場合は、ePWM タイム・ベース・カウンタがゼロに達すると (TBCTR=0x0000)、指定したピン状態が自動的にクリアされます。したがってこのモードでは、PWM サイクルごとにトリップ・イベントがクリアまたはリセットされます。TZFLG[CBC] フラグ・ビットは、TZCLR[CBC] ビットへのライトによって手動でクリアされるまではセットされたままになります。TZFLG[CBC] ビットをクリアしたときにサイクル・バイ・サイクル・トリップ・イベントがまだ存在する場合は、そのビットがもう一度すぐにセットされます。

- **ワンショット (OSHT) :**

ワンショット・トリップ・イベントが発生すると、TZCTL レジスタで指定したアクションが EPWMxA または EPWMxB 出力 (あるいはその両方) ですぐ実行されます。表 2-18 に可能なアクションを示します。また、ワンショット・トリップ・イベント・フラグ (TZFLG[OST]) がセットされ、EPWMx\_TZINT 割り込みが生成されます (TZEINT レジスタと PIE ペリフェラルでイネーブルになっ

ている場合)。ワンショット・トリップ状態は、TZCLR[OST] ビットへのライトによって手動でクリアする必要があります。  
トリップ・イベントの発生時にとられるアクションは、TZCTL[TZA] および TZCTL[TZB] レジスタ・ビットで ePWM 出力ピンごとに設定できます。トリップ・イベントでは、可能な 4 つのアクション (表 2-18 を参照) のいずれかを実行できます。

表 2-18 トリップ・イベントでの可能なアクション

TZCTL[TZA] および/または TZCTL[TZB]	EPWM xA および/または EPWMxB	コメント
0,0	ハイ・インピーダンス	トリップされる
0,1	強制的に High 状態になる	トリップされる
1,0	強制的に Low 状態になる	トリップされる
1,1	変化なし	何もしない 出力は変更されない

### 例 2-8 トリップ・ゾーン設定

#### シナリオ A :

TZ1 のワンショット・トリップ・イベントは EPWM1A と EPWM1B の両方を Low にし、EPWM2A と EPWM2B を強制的に High にします。

- ePWM1 レジスタを以下のように設定します。
  - TZSEL[OSHT1]=1: TZ1 を ePWM1 のワンショット・イベント・ソースとしてイネーブルにします。
  - TZCTL[TZA]=2: EPWM1A はトリップ・イベントで強制的に Low にされます。
  - TZCTL[TZB]=2: EPWM1B はトリップ・イベントで強制的に Low にされます。
- ePWM2 レジスタを以下のように設定します。
  - TZSEL[OSHT1]=1: TZ1 を ePWM2 のワンショット・イベント・ソースとしてイネーブルにします。
  - TZCTL[TZA]=1: EPWM2A はトリップ・イベントで強制的に High にされます。
  - TZCTL[TZB]=1: EPWM2B はトリップ・イベントで強制的に High にされます。

#### シナリオ B :

TZ5 のサイクル・バイ・サイクル・イベントは、EPWM1A と EPWM1B の両方を Low にします。TZ1 または TZ6 のワンショット・イベントは、EPWM2A をハイ・インピーダンス状態にします。

- ePWM1 レジスタを以下のように設定します。
  - TZSEL[CBC5]=1: TZ5 を ePWM1 のサイクル・バイ・サイクル・イベント・ソースとしてイネーブルにします。
  - TZCTL[TZA]=2: EPWM1A はトリップ・イベントで強制的に Low にされます。
  - TZCTL[TZB]=2: EPWM1B はトリップ・イベントで強制的に Low にされます。
- ePWM2 レジスタを以下のように設定します。
  - TZSEL[OSHT1]=1: TZ1 を ePWM2 のワンショット・イベント・ソースとしてイネーブルにします。
  - TZSEL[OSHT6]=1: TZ6 を ePWM1 のワンショット・イベント・ソースとしてイネーブルにします。
  - TZCTL[TZA]=0: EPWM1A は、トリップ・イベントでハイ・インピーダンス状態になります。
  - TZCTL[TZB]=3: EPWM1B はトリップ・イベントを無視します。

2.7.4 トリップ・イベント割り込みの生成

図 2-35 と 図 2-36 は、それぞれトリップ・ゾーン・サブモジュール制御と割り込みロジックを表しています。

図 2-35 トリップ・ゾーン・サブモジュール・モード制御ロジック

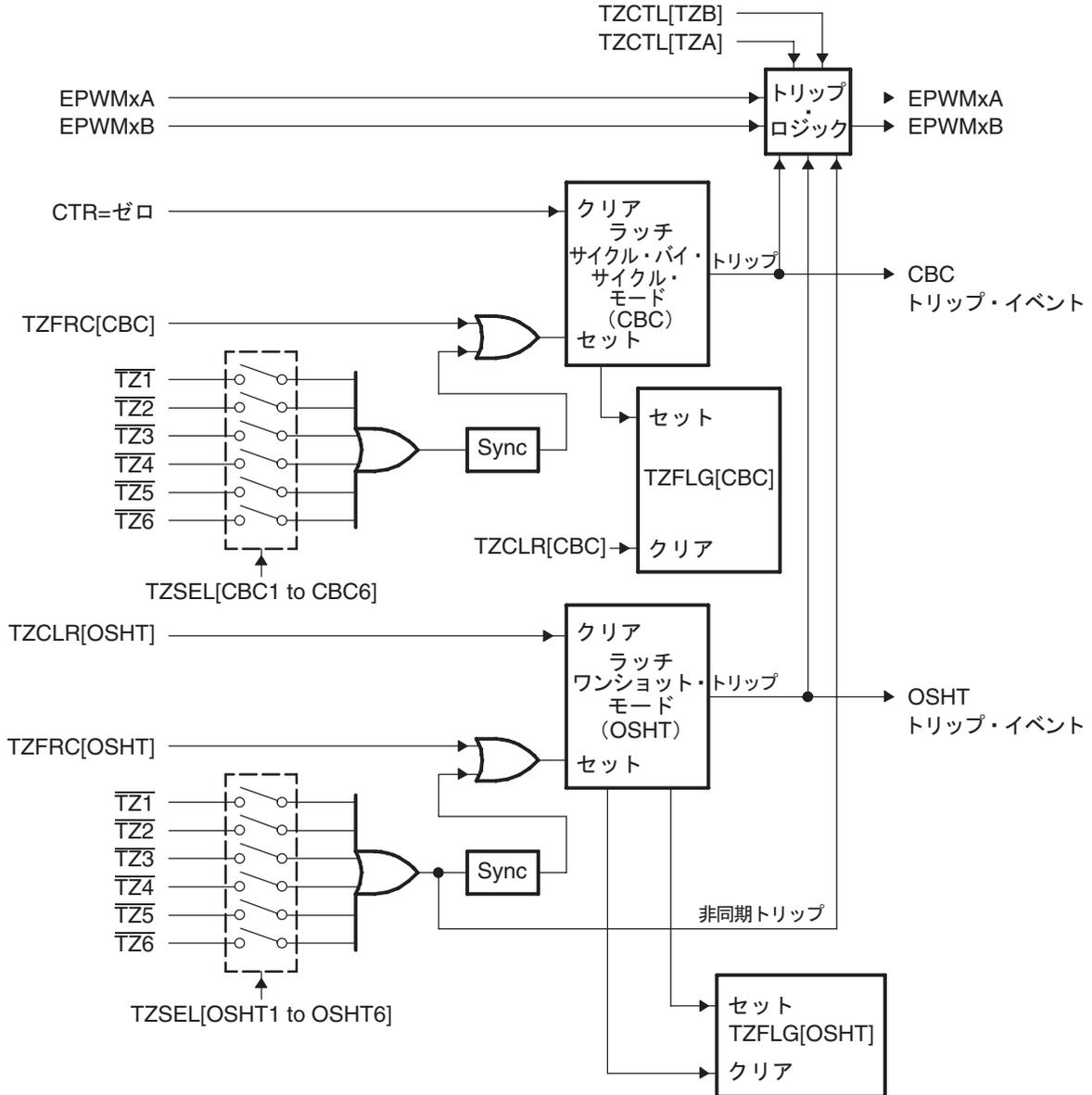
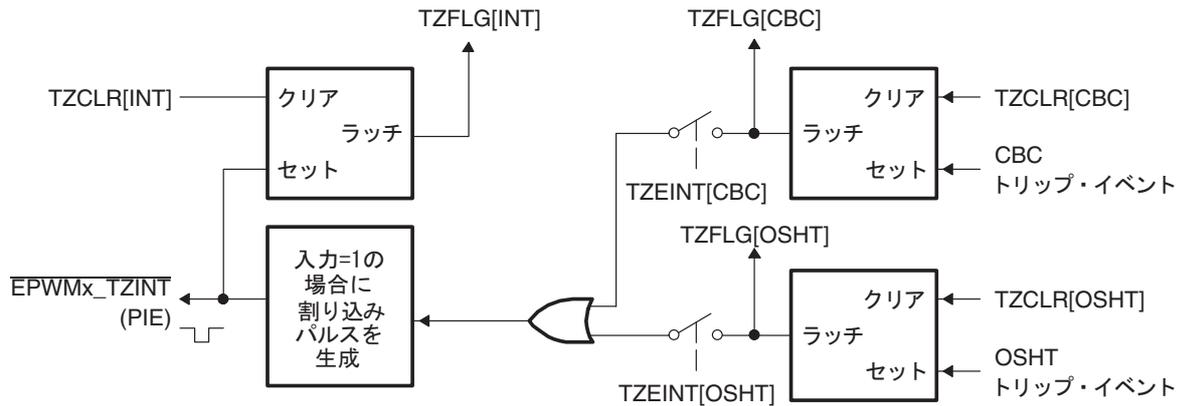


図 2-36 トリップ・ゾーン・サブモジュール割り込みロジック



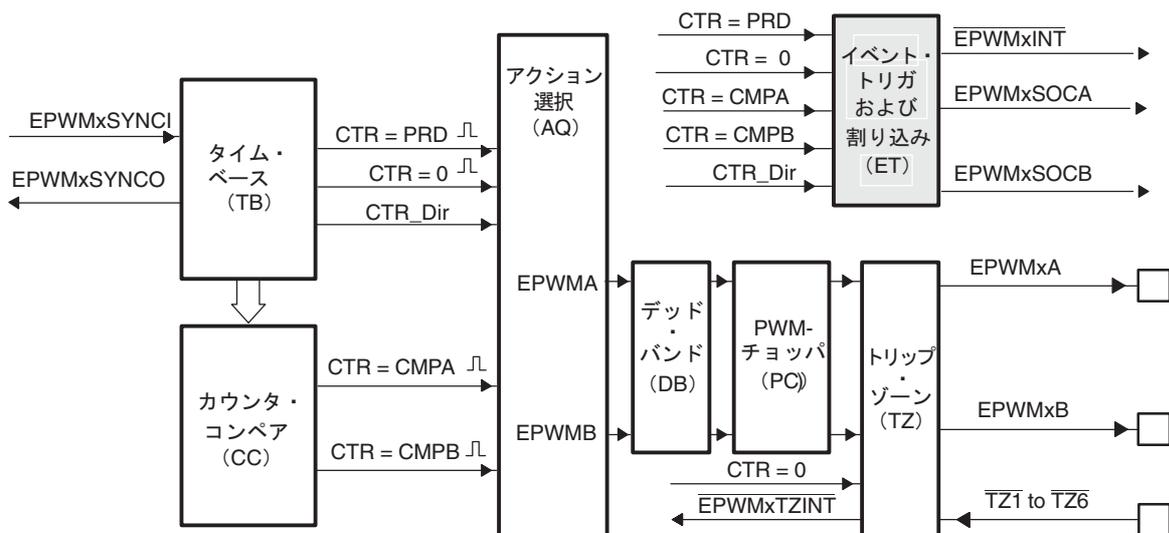
## 2.8 イベント・トリガ (ET) サブモジュール

イベント・トリガ・サブモジュールの主要機能は以下のとおりです。

- タイム・ベース・サブモジュールとカウンタ・コンペア・サブモジュールによって生成されたイベント入力を受信する
- アップ/ダウン・イベント判断にタイム・ベース方向情報を使用する
- プリスケール・ロジックを使用して、以下のタイミングで割り込みリクエストと ADC 変換開始を発行する
  - イベントごと
  - 2回のイベントごと
  - 3回のイベントごと
- イベント・カウンタおよびフラグによってイベント生成の完全な可視性を提供する
- ソフトウェアによる強制的な割り込みと ADC 変換開始を可能にする

イベント・トリガ・サブモジュールは、タイム・ベース・サブモジュールとカウンタ・コンペア・サブモジュールによって生成されたイベントを管理し、選択されたイベントの発生時に CPU への割り込み、または ADC への変換開始パルス (あるいはその両方) を生成します。図 2-37 は、ePWM システム内のどこにイベント・トリガ・サブモジュールが存在するのかを表しています。

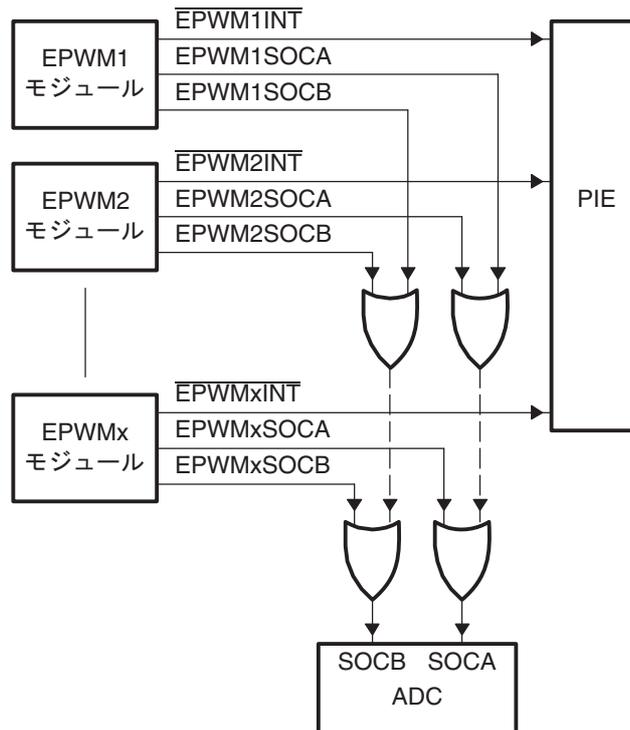
図 2-37 イベント・トリガ・サブモジュール



### 2.8.1 イベント・トリガ・サブモジュールの動作概要

以下の各セクションでは、イベント・トリガ・サブモジュールの動作上の重要点について説明します。各 ePWM モジュールには、PIE に接続された 1 つの割り込みリクエスト・ラインと、ADC モジュールに接続された 2 つの変換開始信号（シーケンサごとに 1 つ）があります。図 2-38 に示すように、すべての ePWM モジュールの ADC 変換開始が論理和演算されます。したがって、複数のモジュールで ADC 変換開始を起動できます。1 つの変換開始ラインで 2 つのリクエストが発生した場合は、1 つのリクエストだけが ADC によって認識されます。

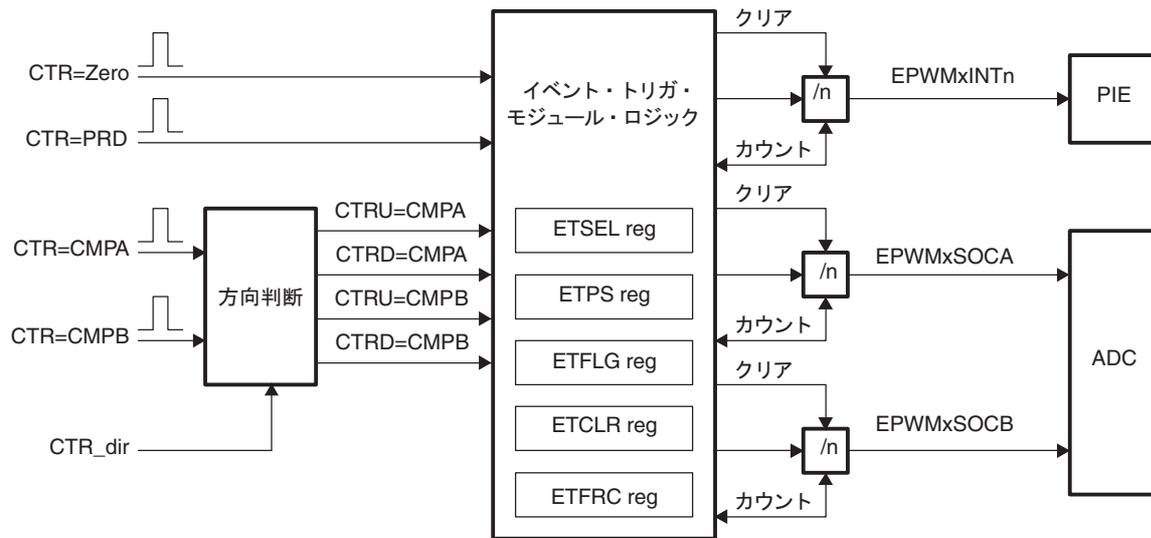
図 2-38 ADC 変換開始および割り込み信号のイベント・トリガ・サブモジュール相互接続性



イベント・トリガ・サブモジュールは、さまざまなイベント状態（図 2-39 に示したイベント・トリガ・サブモジュールに対する左側の入力）を監視します。また、これらのイベントをプリスケールしてから割り込みリクエストや ADC 変換開始を発行するように設定できます。イベント・トリガ・プリスケールリング・ロジックでは、以下のタイミングで割り込みリクエストと ADC 変換開始を発行できます。

- イベントごと
- 2 回のイベントごと
- 3 回のイベントごと

図 2-39 イベント入力とプリスケール出力を示しているイベント・トリガ・サブモジュール



イベント・トリガ・サブモジュールの設定に使用される主なレジスタを表 2-19 に示します。

表 2-19 イベント・トリガ・サブモジュールのレジスタ

レジスタ名	アドレス・オフセット	シャドウ化	説明
ETSEL	0x0019	無	イベント・トリガ選択レジスタ
ETPS	0x001A	無	イベント・トリガ・プリスケール・レジスタ
ETFLG	0x001B	無	イベント・トリガ・フラグ・レジスタ
ETCLR	0x001C	無	イベント・トリガ・クリア・レジスタ
ETFRC	0x001D	無	イベント・トリガ強制レジスタ

- ETSEL — このレジスタでは、割り込みをトリガするイベントやADC変換を開始するイベントを選択します。
- ETPS — このレジスタでは、上記のイベント・プリスケール・オプションをプログラムします。
- ETFLG — これらは、選択およびプリスケールされたイベントの状態を示すフラグ・ビットです。
- ETCLR — これらのビットを使用すると、ETFLG レジスタのフラグ・ビットをソフトウェアでクリアすることができます。
- ETFRC — これらのビットを使用すると、ソフトウェアによるイベントの強制が可能になります。デバッグやソフトウェア介入の場合に役立ちます。

各種レジスタ・ビットと割り込みおよび ADC 変換開始ロジックがどのように相互作用するのか、[図 2-40](#)、[図 2-41](#)、[図 2-42](#) に詳細を示します。

[図 2-40](#) は、イベント・トリガの割り込み生成ロジックを示しています。割り込み周期 (ETPS[INTPRD]) ビットでは、割り込みパルスを生成するのに必要なイベントの数が指定されます。用意されている選択肢は以下のとおりです。

- 割り込みを生成しない
- イベントごとに割り込みを生成する
- 2回のイベントごとに割り込みを生成する
- 3回のイベントごとに割り込みを生成する

どのイベントが割り込みを発生させるのかは、割り込み選択 (ETSEL[INTSEL]) ビットによって設定されます。イベントは、以下のいずれかにすることができます。

- タイム・ベース・カウンタはゼロに等しい (TBCTR=0x0000)
- タイム・ベース・カウンタは周期に等しい (TBCTR=TBPRD)
- タイマがインクリメントされているときにタイム・ベース・カウンタがコンペア A レジスタ (CMPA) に等しい
- タイマがデクリメントされているときにタイム・ベース・カウンタがコンペア A レジスタ (CMPA) に等しい
- タイマがインクリメントされているときにタイム・ベース・カウンタがコンペア B レジスタ (CMPB) に等しい
- タイマがデクリメントされているときにタイム・ベース・カウンタがコンペア B レジスタ (CMPB) に等しい

発生したイベントの数は、割り込みイベント・カウンタ (ETPS[INTCNT]) レジスタ・ビットからリードできます。つまり、指定したイベントが発生すると、ETPS[INTPRD] で指定した値に達するまで ETPS[INTCNT] ビットがインクリメントされます。ETPS[INTCNT]=ETPS[INTPRD] の場合は、カウンタが停止し、出力がセットされます。

- 割り込みをイネーブル (ETSEL[INTEN]=1) にして、割り込みフラグをクリア (ETFLG[INT]=0) した場合は、割り込みパルスが生成され、割り込みフラグがセット (ETFLG[INT]=1) されて、イベント・カウンタがクリア (ETPS[INTCNT]=0) されます。
- 割り込みをディスエーブル (ETSEL[INTEN]=0) にするか、または割り込みフラグをセット (ETFLG[INT]=1) した場合は、カウンタが周期値に達すると (ETPS[INTCNT]=ETPS[INTPRD])、イベントのカウンタが停止されます。

INTPRD ビットにライトすると、自動的にカウンタがクリアされ (INTCNT=0)、カウンタ出力がリセットされます (したがって、割り込みは生成されません)。

ETFRC[INT] ビットに 1 をライトすると、イベント・カウンタ INTCNT がインクリメントされます。INTCNT=INTPRD の場合、カウンタは上記のように動作します。

INTPRD=0 の場合は、カウンタがディスエーブルになるため、イベントは検出されず、ETFRC[INT] ビットも無視されます。

上記の定義は、イベントごと、2 回のイベントごと、または 3 回のイベントごとに割り込みを生成できることを意味します。4 回以上のイベントごとに割り込みを生成することはできません。

図 2-40 イベント・トリガ割り込みジェネレータ

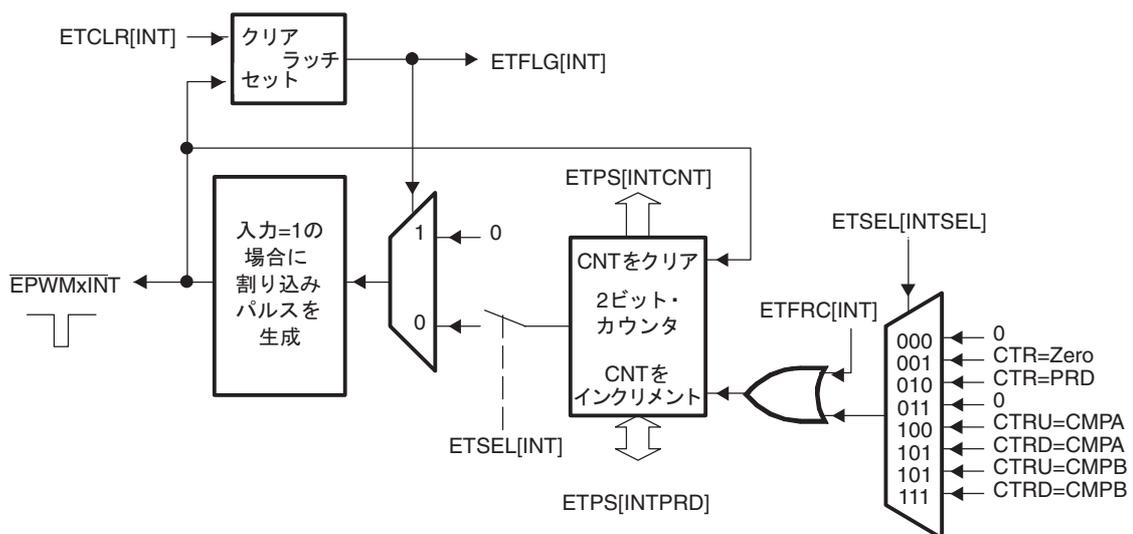


図 2-41 は、イベント・トリガの変換開始 A (SOCA) パルス・ジェネレータの動作を示しています。ETPS[SOCACNT] カウンタと ETPS[SOCAPRD] 周期値は、パルスが継続的に生成されることを除き、割り込みジェネレータと同じように動作します。つまり、パルスが生成されるとパルス・フラグ ETFLG[SOCA] がラッチされますが、後続のパルス生成は停止されません。イネーブル / ディスエーブル・ビット ETPS[ESOCA] を使用するとパルス生成が停止されますが、割り込み生成ロジックの場合と同様に、周期値に達するまでは入力イベントを引き続きカウントできます。SOCA および SOCB パルスをトリガするイベントは、ETSEL[SOCASEL] および ETSEL[SOCSSEL] ビットで個別に設定できます。可能なイベントは、割り込み生成ロジックの場合に指定できるイベントと同じです。

図 2-41 イベント・トリガ SOCA パルス・ジェネレータ

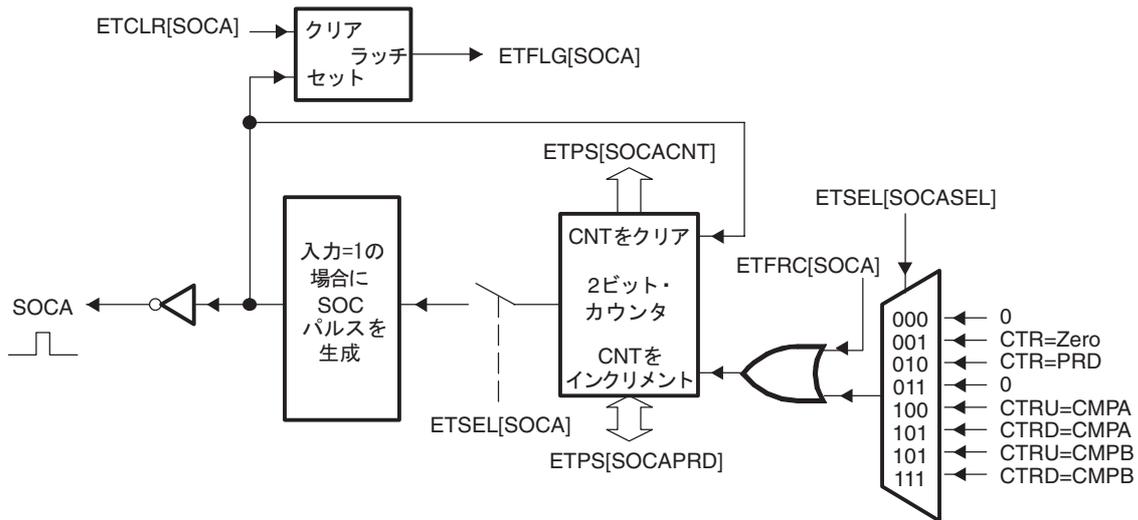
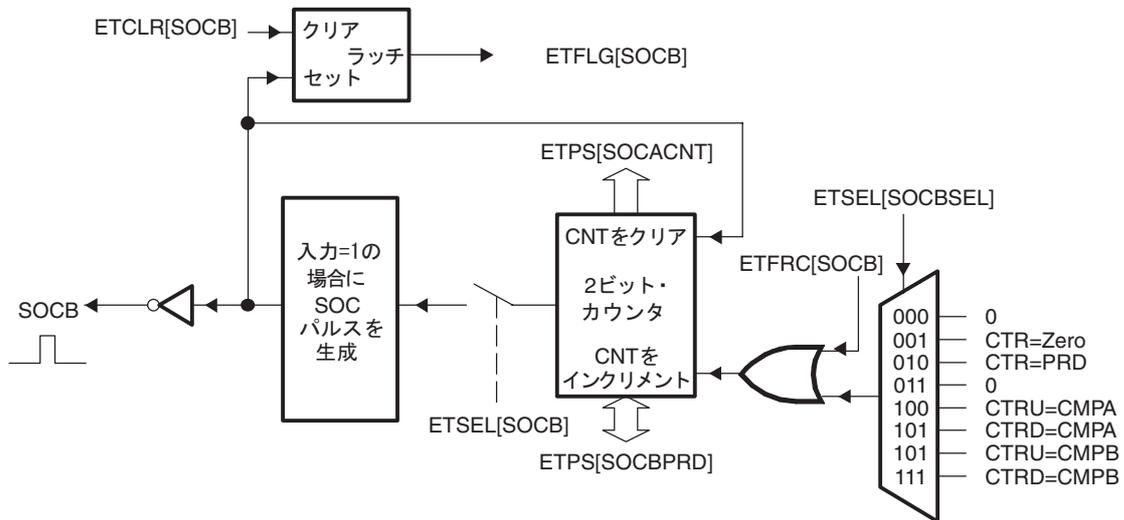


図 2-42 は、イベント・トリガの変換開始 B (SOCB) パルス・ジェネレータの動作を示しています。イベント・トリガの SOCB パルス・ジェネレータは、SOCA と同じように動作します。

図 2-42 イベント・トリガ SOCB パルス・ジェネレータ



## パワー・トポロジへの応用

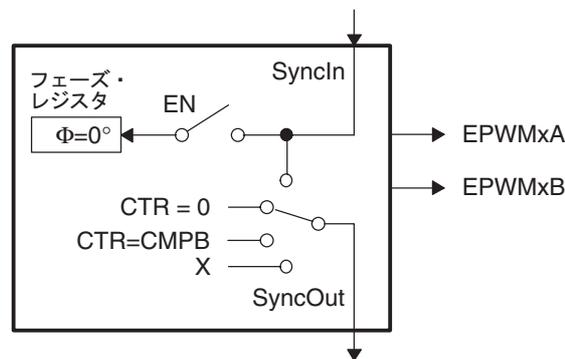
ePWM モジュールは、完全なスタンドアロン・モジュールとしての動作や他の同一 ePWM モジュールと同期した動作に必要なローカル・リソースをすべて備えています。

項目	ページ
3.1 複数モジュールの概要 .....	72
3.2 重要な設定機能 .....	72
3.3 独立した周波数による複数のバック・コンバータの制御 .....	73
3.4 同じ周波数による複数のバック・コンバータの制御 .....	77
3.5 複数のハーフ H ブリッジ (HHB) コンバータの制御 .....	80
3.6 モータ (ACI および PMSM) に対する デュアル 3 相インバータの制御 .....	82
3.7 PWM モジュール間のフェーズ制御を使用した実際の応用 .....	86
3.8 3 相インターリーブ型 DC/DC コンバータの制御 .....	87
3.9 ZVSFB (Zero Voltage Switched Full Bridge : ゼロ電圧スイッチング・フル・ブリッジ) コンバータの制御 .....	91

### 3.1 複数モジュールの概要

このユーザーズ・ガイドでこれまで説明してきた動作は、すべて単一モジュールの動作です。システムで協調動作する複数のモジュールを理解しやすくするために、リファレンスで説明する ePWM モジュールは図 3-1 に示した比較的単純なブロック図で表されています。この簡略化された ePWM ブロックは、協調動作している複数の ePWM モジュールでマルチスイッチ・パワー・トポロジを制御する方法の説明に必要な主要リソースだけを示しています。

図 3-1 簡略化された ePWM モジュール



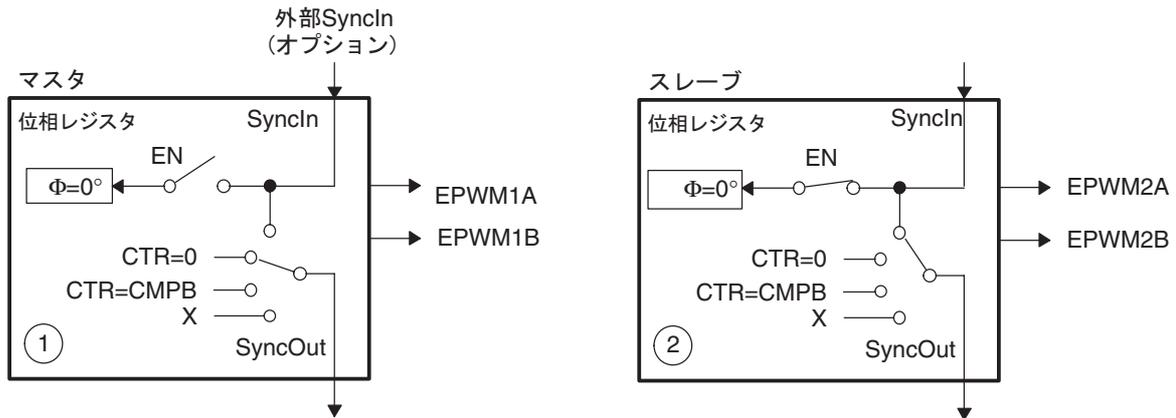
### 3.2 重要な設定機能

各モジュールに用意されている重要な設定の選択肢は以下のとおりです。

- SyncIn のオプション
  - 入ってくる同期ストロブで固有のカウンタに位相レジスタをロードする—イネーブル (EN) スイッチが閉じている状態
  - 何もしない、または入ってくる同期ストロブを無視する—イネーブル・スイッチが開いている状態
  - 同期のスルー - SyncOut が SyncIn に接続される
  - マスタ・モードとなり、PWM の周期境界で同期を提供する—SyncOut が CTR=PRD に接続される
  - マスタ・モードとなり、プログラム可能な任意の時点で同期を提供する—SyncOut が CTR=CMPB に接続される
  - モジュールがスタンダアロン・モードになり、他のモジュールには同期を提供しない—SyncOut が X (ディセーブル) に接続される
- SyncOut のオプション
  - 同期のスルー - SyncOut が SyncIn に接続される
  - マスタ・モードとなり、PWM の周期境界で同期を提供する—SyncOut が CTR=PRD に接続される
  - マスタ・モードとなり、プログラム可能な任意の時点で同期を提供する—SyncOut が CTR=CMPB に接続される
  - モジュールがスタンダアロン・モードになり、他のモジュールには同期を提供しない—SyncOut が X (ディセーブル) に接続される

SyncOut の各選択肢については、イネーブル・スイッチにより、SyncIn ストロブ入力時に独自のカウンタに新しい位相値をロードすること、あるいはその入力を見捨てることをモジュールで選択することもできます。さまざまな組み合わせが考えられますが、その中で最も一般的な 2 つ (マスタ・モジュール・モードとスレーブ・モジュール・モード) を図 3-2 に示します。

図 3-2 一般的なマスタとして設定された EPWM1、スレーブとして設定された EPWM2



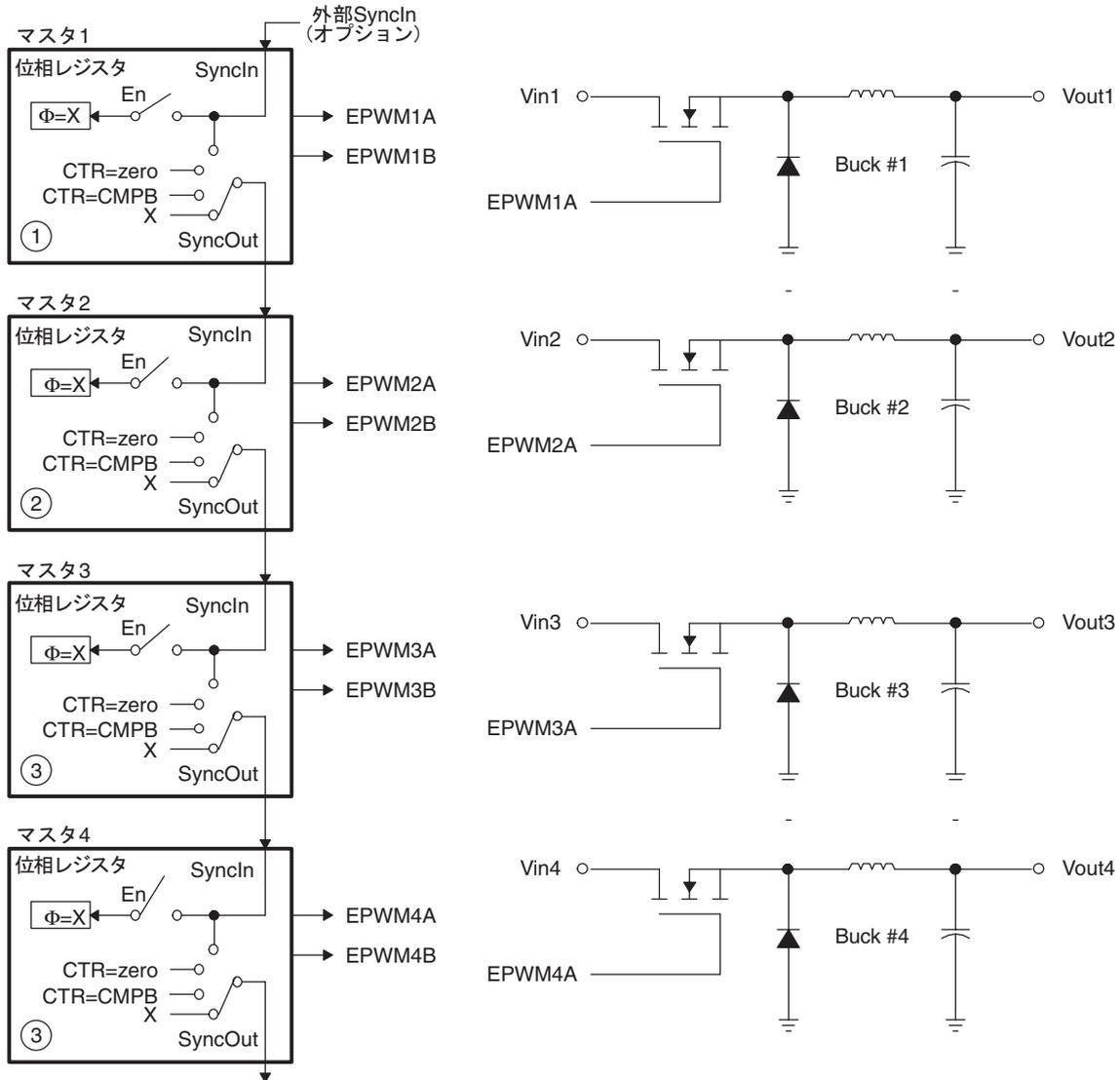
### 3.3 独立した周波数による複数のバック・コンバータの制御

最も単純なパワー・コンバータ・トポロジの1つはバックです。

マスタとして設定された単一 ePWM モジュールは、同じ PWM 周波数で 2 つのバック・ステージを制御できます。バック・コンバータごとに独立した周波数制御が必要な場合は、コンバータ・ステージごとに 1 つの ePWM モジュールを割り当てる必要があります。図 3-3 は、それぞれ独立した周波数で動作している 4 つのバック・ステージを示しています。この場合、4 つの ePWM モジュールは、すべてマスタとして設定されており、同期は使用していません。

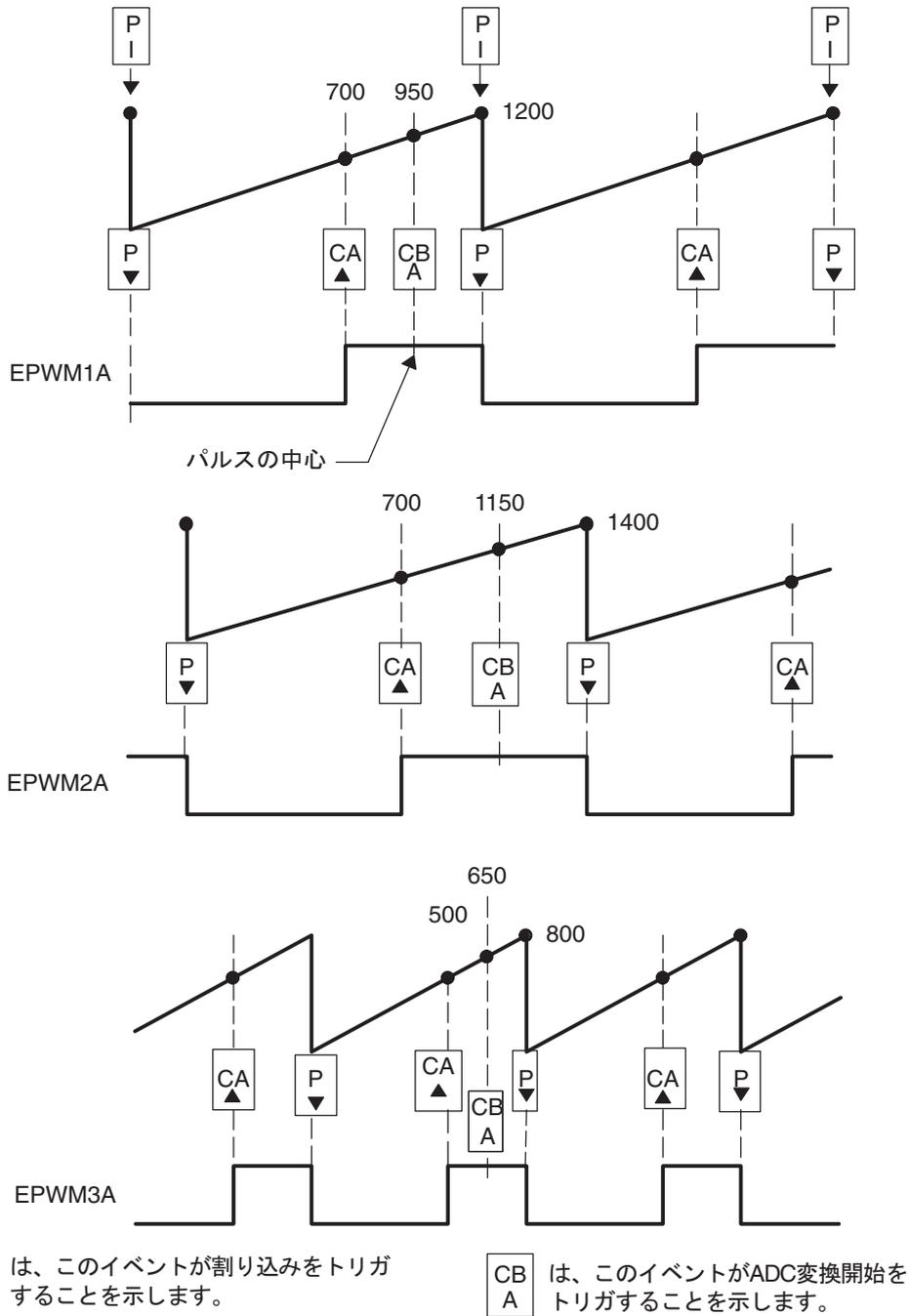
図 3-4 は、図 3-3 に示したセットアップによって生成される波形です。ステージは 4 つありますが、3 つの波形だけが示されていることに注意してください。

図 3-3 4つのバック・ステージの制御 ( $F_{PWM1} \neq F_{PWM2} \neq F_{PWM3} \neq F_{PWM4}$ )



注：  $\Phi=X$  は、位相レジスタの値が "don't care" であることを意味します。

図 3-4 図 3-3 のバック波形 (注: ここでは、3つのバックだけを示しています)



**例 3-1 図 3-4 に示した例の設定**

```

//=====
// (Note: code for only 3 modules shown)
// Initialization Time
//=====
// EPWM Module 1 config
EPwm1Regs.TBPRD = 1200; // Period = 1201 TBCLK counts
EPwm1Regs.TBPHS = 0; // Set Phase register to zero
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP; // Asymmetrical mode
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Phase loading disabled
EPwm1Regs.TBCTL.bit.PRDLN = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCSEL = TB_SYNC_DISABLE;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm1Regs.AQCTLA.bit.PRDLN = AQ_CLEAR;
EPwm1Regs.AQCTLA.bit.CAU = AQ_SET;
// EPWM Module 2 config
EPwm2Regs.TBPRD = 1400; // Period = 1401 TBCLK counts
EPwm2Regs.TBPHS = 0; // Set Phase register to zero
EPwm2Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP; // Asymmetrical mode
EPwm2Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Phase loading disabled
EPwm2Regs.TBCTL.bit.PRDLN = TB_SHADOW;
EPwm2Regs.TBCTL.bit.SYNCSEL = TB_SYNC_DISABLE;
EPwm2Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm2Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm2Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm2Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm2Regs.AQCTLA.bit.PRDLN = AQ_CLEAR;
EPwm2Regs.AQCTLA.bit.CAU = AQ_SET;
// EPWM Module 3 config
EPwm3Regs.TBPRD = 800; // Period = 801 TBCLK counts
EPwm3Regs.TBPHS = 0; // Set Phase register to zero
EPwm3Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP;
EPwm3Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Phase loading disabled
EPwm3Regs.TBCTL.bit.PRDLN = TB_SHADOW;
EPwm3Regs.TBCTL.bit.SYNCSEL = TB_SYNC_DISABLE;
EPwm3Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm3Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm3Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm3Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm3Regs.AQCTLA.bit.PRDLN = AQ_CLEAR;
EPwm3Regs.AQCTLA.bit.CAU = AQ_SET;
//
// Run Time (Note: Example execution of one run-time instant)
//=====
EPwm1Regs.CMPA.half.CMPA = 700; // adjust duty for output EPWM1A
EPwm2Regs.CMPA.half.CMPA = 700; // adjust duty for output EPWM2A
EPwm3Regs.CMPA.half.CMPA = 500; // adjust duty for output EPWM3A
    
```

### 3.4 同じ周波数による複数のバック・コンバータの制御

同期が必要な場合は、ePWM モジュール 2 をスレーブとして設定し、モジュール 1 の整数倍 (N) の周波数で動作させることができます。マスタからスレーブへの同期信号により、これらのモジュールはロックされたままになります。図 3-5 は、そのような設定を示しています。図 3-6 は、その設定によって生成される波形を示しています。

図 3-5 4 つのバック・ステージの制御 (注:  $F_{PWM2} = N \times F_{PWM1}$ )

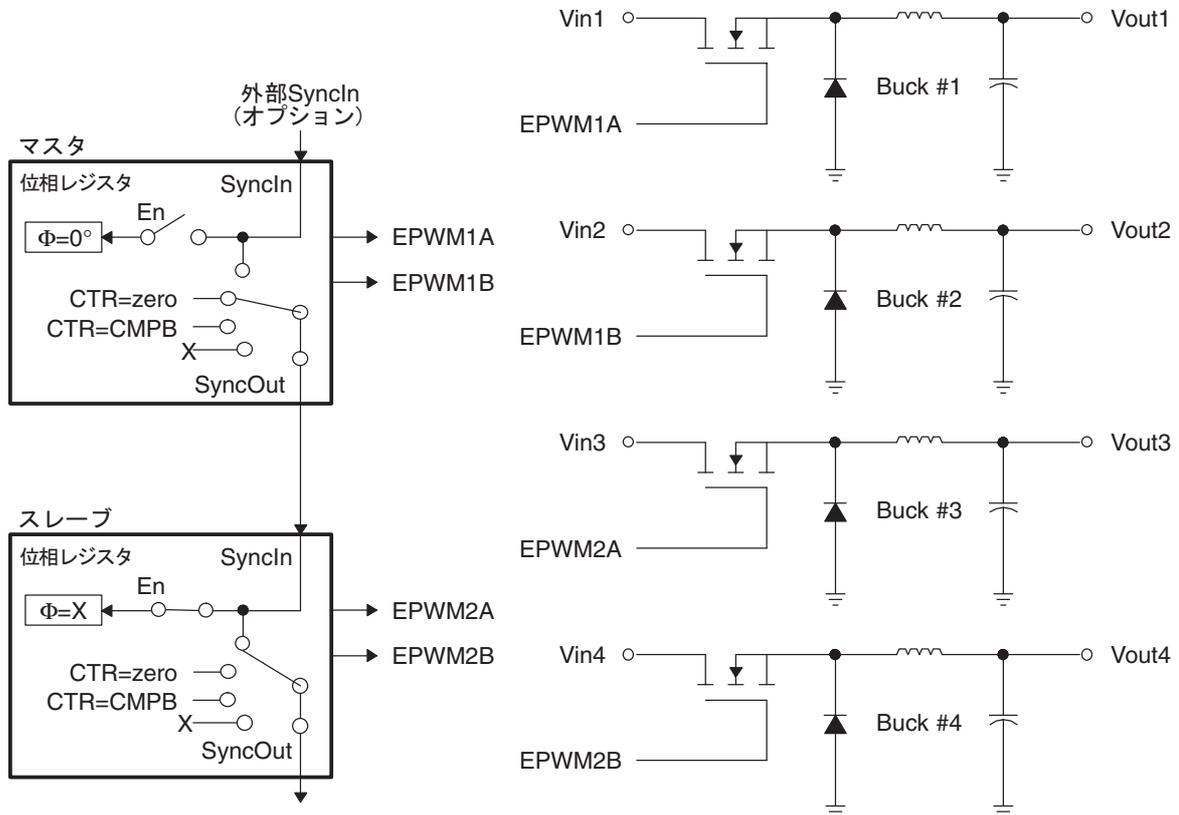
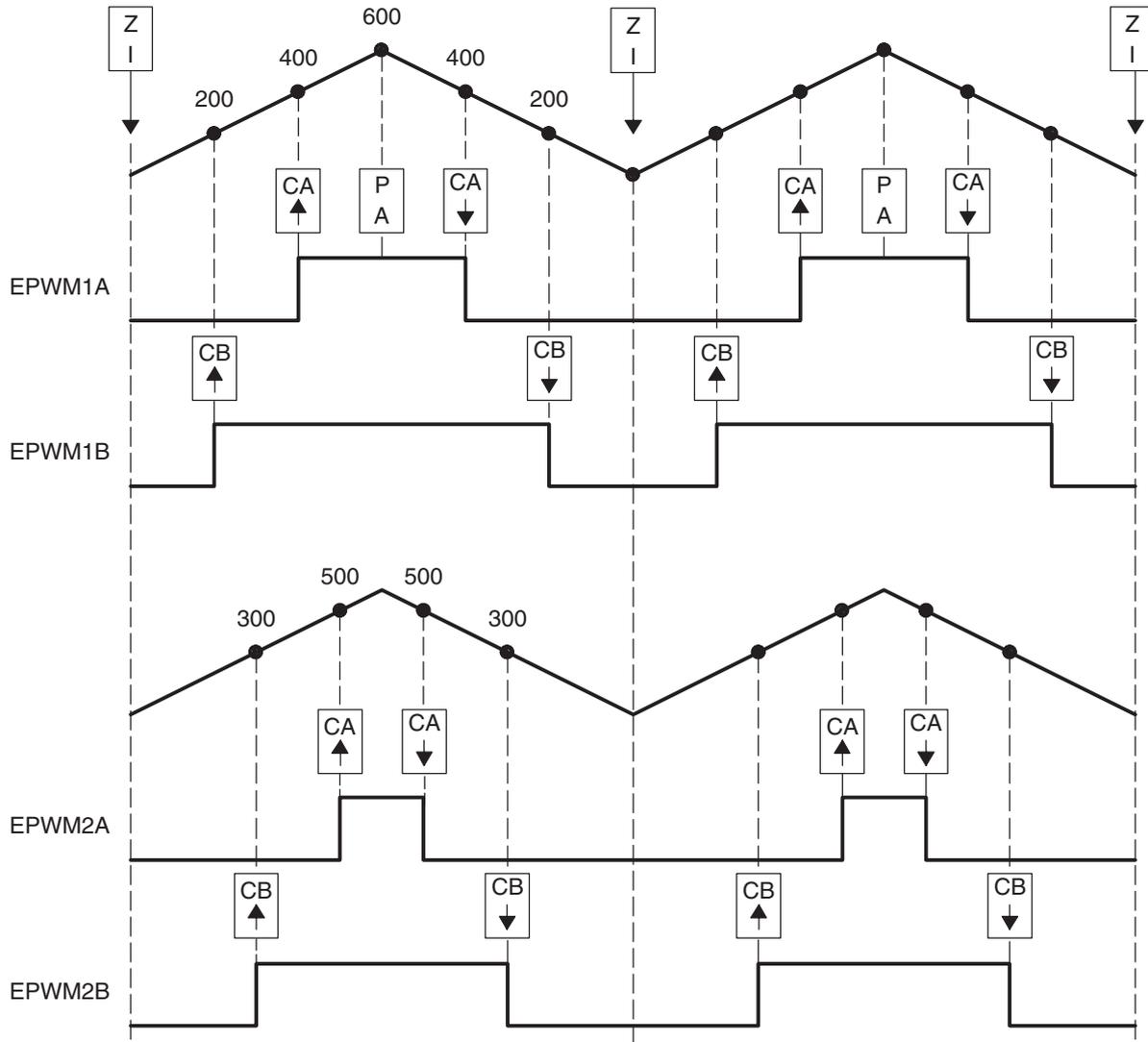


図 3-6 図 3-5 のバック波形 (注:  $F_{PWM2}=F_{PWM1}$ )



例 3-2 図 3-5 に示した設定コード

```

//=====
// Config
//=====
// Initialization Time
//=====
// EPWM Module 1 config
EPwm1Regs.TBPRD = 600; // Period = 1200 TBCLK counts
EPwm1Regs.TBPHS = 0; // Set Phase register to zero
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetrical mode
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Master module
EPwm1Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCSEL = TB_CTR_ZERO; // Sync down-stream module
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm1Regs.AQCTLA.bit.CAU = AQ_SET; // set actions for EPWM1A
EPwm1Regs.AQCTLA.bit.CAD = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.CBU = AQ_SET; // set actions for EPWM1B
EPwm1Regs.AQCTLB.bit.CBD = AQ_CLEAR;
// EPWM Module 2 config
EPwm2Regs.TBPRD = 600; // Period = 1200 TBCLK counts
EPwm2Regs.TBPHS = 0; // Set Phase register to zero
EPwm2Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetrical mode
EPwm2Regs.TBCTL.bit.PHSEN = TB_ENABLE; // Slave module
EPwm2Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm2Regs.TBCTL.bit.SYNCSEL = TB_SYNC_IN; // sync flow-through
EPwm2Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm2Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm2Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm2Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm2Regs.AQCTLA.bit.CAU = AQ_SET; // set actions for EPWM2A
EPwm2Regs.AQCTLA.bit.CAD = AQ_CLEAR;
EPwm2Regs.AQCTLB.bit.CBU = AQ_SET; // set actions for EPWM2B
EPwm2Regs.AQCTLB.bit.CBD = AQ_CLEAR;
//
// Run Time (Note: Example execution of one run-time instance)
//=====
EPwm1Regs.CMPA.half.CMPA = 400; // adjust duty for output EPWM1A
EPwm1Regs.CMPB = 200; // adjust duty for output EPWM1B
EPwm2Regs.CMPA.half.CMPA = 500; // adjust duty for output EPWM2A
EPwm2Regs.CMPB = 300; // adjust duty for output EPWM2B
  
```

### 3.5 複数のハーフHブリッジ (HHB) コンバータの制御

複数のスイッチング・エレメントの制御を必要とするトポロジは、これらの同一 ePWM モジュールで扱うこともできます。ハーフHブリッジ・ステージは、単一の ePWM モジュールで制御できます。この制御は、複数のステージに拡張可能です。図 3-7 は、2つの同期したハーフHブリッジ・ステージの制御を示しています。ここで、ステージ2はステージ1の整数倍 (N) の周波数で動作できます。

図 3-8 は、図 3-7 に示した設定によって生成される波形です。

モジュール2 (スレーブ) は、同期のスルー用に設定されます。必要な場合、この設定では3番目のハーフHブリッジをPWMモジュール3によって制御でき、また最も重要なこととして、マスタ・モジュール1と同期した状態のままにすることができます。

図 3-7 2つのハーフ・ブリッジ・ステージの制御 ( $F_{PWM2}=N \times F_{PWM1}$ )

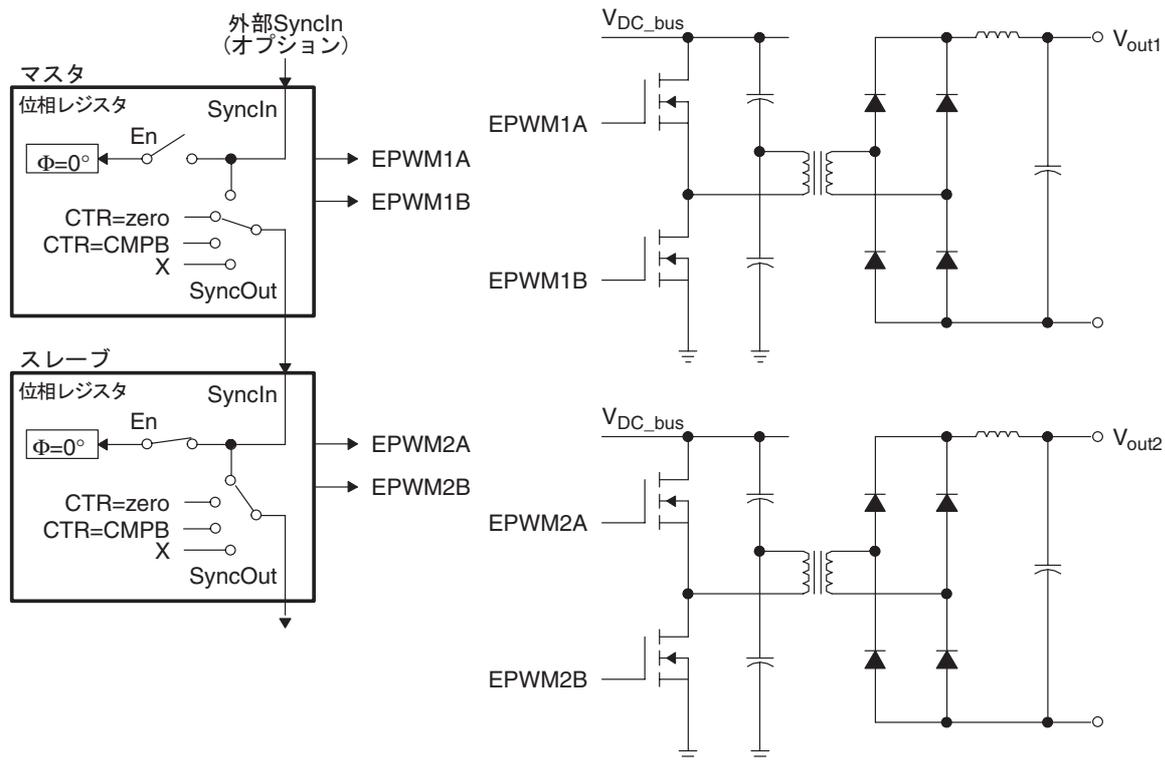
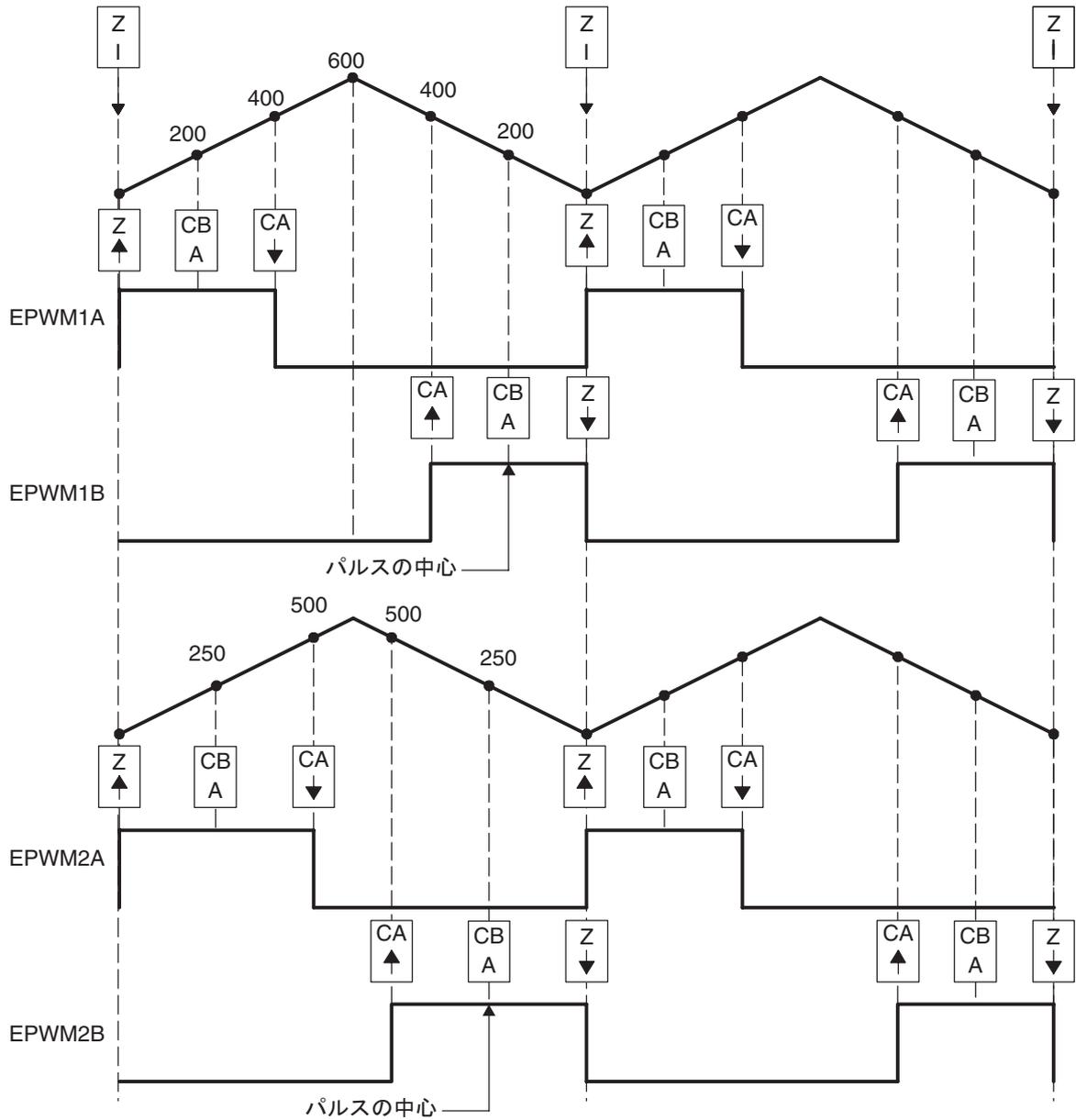


図 3-8 図 3-7 のハーフブリッジ波形 (注:  $F_{PWM2}=F_{PWM1}$ )



例 3-3 図 3-7 に示した設定コード

```
//=====
// Config
//=====
// Initialization Time
//=====
// EPWM Module 1 config
EPwm1Regs.TBPRD = 600; // Period = 1200 TBCLK counts
EPwm1Regs.TBPHS = 0; // Set Phase register to zero
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetrical mode
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Master module
EPwm1Regs.TBCTL.bit.PRDLN = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCSEL = TB_CTR_ZERO; // Sync down-stream module
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm1Regs.AQCTLA.bit.ZRO = AQ_SET; // set actions for EPWM1A
EPwm1Regs.AQCTLA.bit.CAU = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.ZRO = AQ_CLEAR; // set actions for EPWM1B
EPwm1Regs.AQCTLB.bit.CAD = AQ_SET;
// EPWM Module 2 config
EPwm2Regs.TBPRD = 600; // Period = 1200 TBCLK counts
EPwm2Regs.TBPHS = 0; // Set Phase register to zero
EPwm2Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetrical mode
EPwm2Regs.TBCTL.bit.PHSEN = TB_ENABLE; // Slave module
EPwm2Regs.TBCTL.bit.PRDLN = TB_SHADOW;
EPwm2Regs.TBCTL.bit.SYNCSEL = TB_SYNC_IN; // sync flow-through
EPwm2Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm2Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm2Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm2Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm2Regs.AQCTLA.bit.ZRO = AQ_SET; // set actions for EPWM1A
EPwm2Regs.AQCTLA.bit.CAU = AQ_CLEAR;
EPwm2Regs.AQCTLB.bit.ZRO = AQ_CLEAR; // set actions for EPWM1B
EPwm2Regs.AQCTLB.bit.CAD = AQ_SET;
//=====
EPwm1Regs.CMPA.half.CMPA = 400; // adjust duty for output EPWM1A & EPWM1B
EPwm1Regs.CMPB = 200; // adjust point-in-time for ADCSOC trigger
EPwm2Regs.CMPA.half.CMPA = 500; // adjust duty for output EPWM2A & EPWM2B
EPwm2Regs.CMPB = 250; // adjust point-in-time for ADCSOC trigger
```

### 3.6 モータ (ACI および PMSM) 用デュアル 3 相インバータの制御

複数のモジュールで単一のパワー・ステージを制御するという考え方は、3 相インバータのケースに拡張できます。その場合、3 つの PWM モジュール (インバータのアームごとに 1 つ) を使用して 6 つのスイッチング・エレメントを制御できます。各アームは同じ周波数で切り替わる必要があります、また、すべてのアームが同期している必要があります。1 つのマスタと 2 つのスレーブからなる構成では、この要件に容易に対処できます。図 3-9 は、6 つの PWM モジュールが 2 つの独立した 3 相インバータ (それぞれモータを動かしている) をどのように制御するのかを示しています。

前のセクションで示したケースと同様に、異なる周波数で各インバータを動作させるか (図 3-9 のようにモジュール 1 とモジュール 4 がマスタになる)、または 1 つのマスタ (モジュール 1) と 5 つのスレーブを使用して両方のインバータを同期させることができます。この場合、モジュール 4、5、6 の周波数 (すべて同じ) は、モジュール 1、2、3 の周波数 (すべて同じ) の整数倍にすることができます。

図 3-9 モータ制御でよく用いられるデュアル 3 相インバータ・ステージの制御

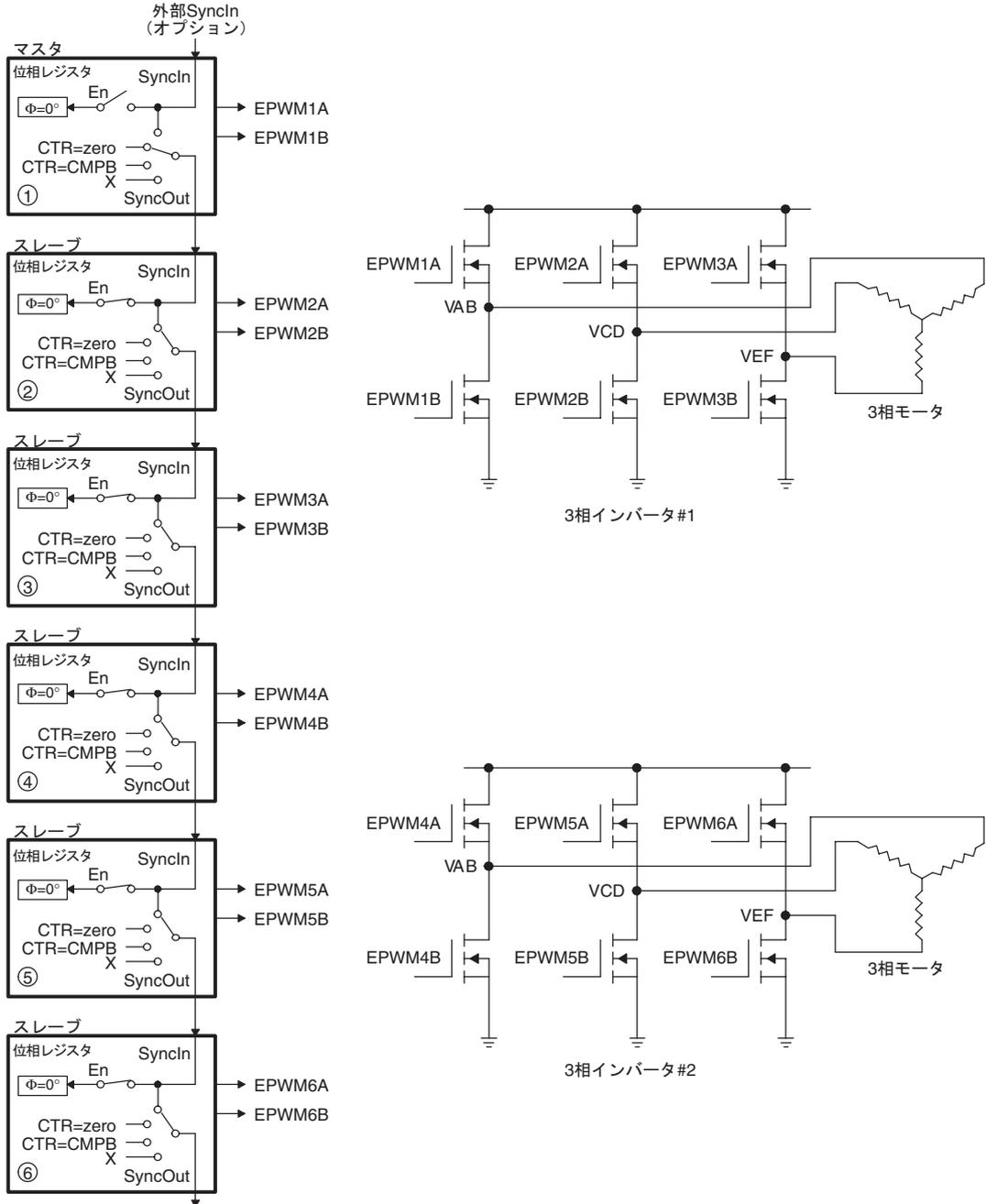
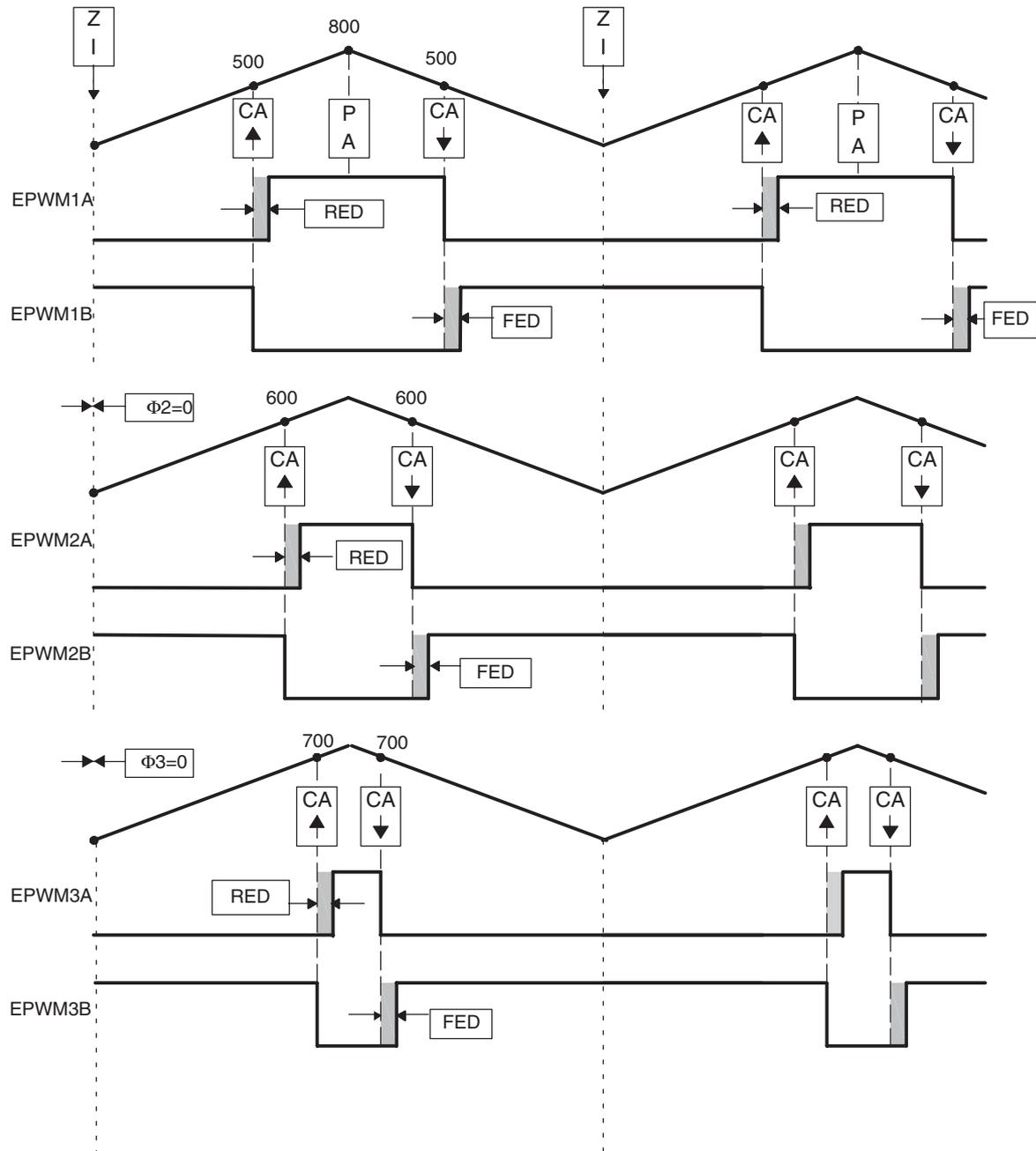


図 3-10 図 3-9 の 3 相インバータ波形 (1 つのインバータだけを示す)



例 3-4 図 3-9 に示した設定コード

```

//=====
// Configuration
//=====
// Initialization Time
//=====// EPWM Module 1 config
    EPwm1Regs.TBPRD = 800; // Period = 1600 TBCLK counts
    EPwm1Regs.TBPHS = 0; // Set Phase register to zero
    EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetrical mode
    EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Master module
    EPwm1Regs.TBCTL.bit.PRDLN = TB_SHADOW;
    EPwm1Regs.TBCTL.bit.SYNCSEL = TB_CTR_ZERO; // Sync down-stream module
    EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
    EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
    EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
    EPwm1Regs.CMPCTL.bit.LOADMODE = CC_CTR_ZERO; // load on CTR=Zero
    EPwm1Regs.AQCTLA.bit.CAU = AQ_SET; // set actions for EPWM1A
    EPwm1Regs.AQCTLA.bit.CAD = AQ_CLEAR;
    EPwm1Regs.DBCTL.bit.MODE = DB_FULL_ENABLE; // enable Dead-band module
    EPwm1Regs.DBCTL.bit.POLSEL = DB_ACTV_HIC; // Active Hi complementary
    EPwm1Regs.DBFED = 50; // FED = 50 TBCLKs
    EPwm1Regs.DBRED = 50; // RED = 50 TBCLKs
// EPWM Module 2 config
    EPwm2Regs.TBPRD = 800; // Period = 1600 TBCLK counts
    EPwm2Regs.TBPHS = 0; // Set Phase register to zero
    EPwm2Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetrical mode
    EPwm2Regs.TBCTL.bit.PHSEN = TB_ENABLE; // Slave module
    EPwm2Regs.TBCTL.bit.PRDLN = TB_SHADOW;
    EPwm2Regs.TBCTL.bit.SYNCSEL = TB_SYNC_IN; // sync flow-through
    EPwm2Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
    EPwm2Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
    EPwm2Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
    EPwm2Regs.CMPCTL.bit.LOADMODE = CC_CTR_ZERO; // load on CTR=Zero
    EPwm2Regs.AQCTLA.bit.CAU = AQ_SET; // set actions for EPWM2A
    EPwm2Regs.AQCTLA.bit.CAD = AQ_CLEAR;
    EPwm2Regs.DBCTL.bit.MODE = DB_FULL_ENABLE; // enable Dead-band module
    EPwm2Regs.DBCTL.bit.POLSEL = DB_ACTV_HIC; // Active Hi complementary
    EPwm2Regs.DBFED = 50; // FED = 50 TBCLKs
    EPwm2Regs.DBRED = 50; // RED = 50 TBCLKs
// EPWM Module 3 config
    EPwm3Regs.TBPRD = 800; // Period = 1600 TBCLK counts
    EPwm3Regs.TBPHS = 0; // Set Phase register to zero
    EPwm3Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetrical mode
    EPwm3Regs.TBCTL.bit.PHSEN = TB_ENABLE; // Slave module
    EPwm3Regs.TBCTL.bit.PRDLN = TB_SHADOW;
    EPwm3Regs.TBCTL.bit.SYNCSEL = TB_SYNC_IN; // sync flow-through
    EPwm3Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
    EPwm3Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
    EPwm3Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
    EPwm3Regs.CMPCTL.bit.LOADMODE = CC_CTR_ZERO; // load on CTR=Zero
    EPwm3Regs.AQCTLA.bit.CAU = AQ_SET; // set actions for EPWM3A
    EPwm3Regs.AQCTLA.bit.CAD = AQ_CLEAR;
    EPwm3Regs.DBCTL.bit.MODE = DB_FULL_ENABLE; // enable Dead-band module
    EPwm3Regs.DBCTL.bit.POLSEL = DB_ACTV_HIC; // Active Hi complementary
    EPwm3Regs.DBFED = 50; // FED = 50 TBCLKs
    EPwm3Regs.DBRED = 50; // RED = 50 TBCLKs
// Run Time (Note: Example execution of one run-time instant)
//=====
EPwm1Regs.CMPA.half.CMPA = 500; // adjust duty for output EPWM1A
EPwm2Regs.CMPA.half.CMPA = 600; // adjust duty for output EPWM2A
EPwm3Regs.CMPA.half.CMPA = 700; // adjust duty for output EPWM3A
  
```

### 3.7 PWM モジュール間の位相制御を使用した実際の応用

これまで、どの例でも位相レジスタ (TBPHS) を使用していませんでした。位相レジスタはゼロに設定されているか、または値が無視されていました。ただし、TBPHS への適切な値のプログラミングにより、複数の PWM モジュールを用いて、各アーム (またはステージ) 間の位相関係に依存する別のクラスのパワー・トポロジを扱うことができます。TB モジュールのセクションで説明したように、PWM モジュールは、Syncln パルスによって TBPHS レジスタを TBCTR レジスタにロードさせるように設定できます。この概念を分かりやすく説明するために、フェーズ関係が  $120^\circ$  のマスタおよびスレーブ・モジュール (つまり、スレーブがマスタに先行する) を図 3-11 に示します。

図 3-11 フェーズ制御用の 2 つの PWM モジュールの設定

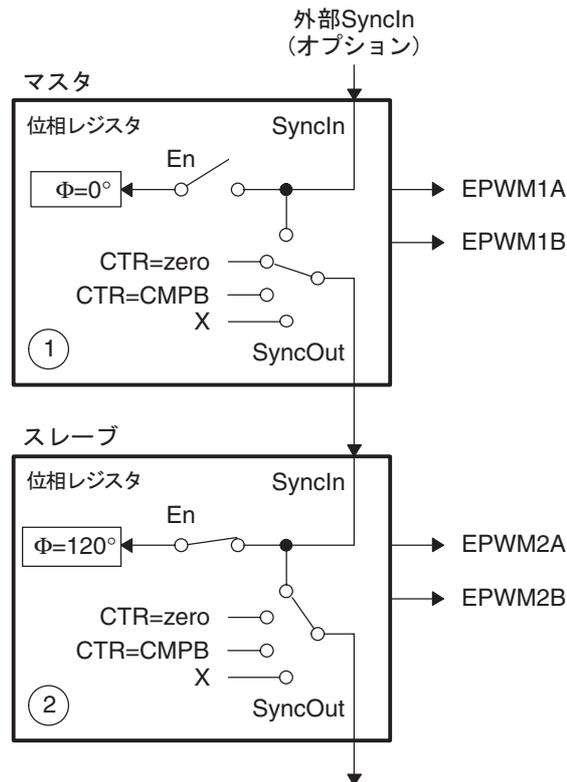
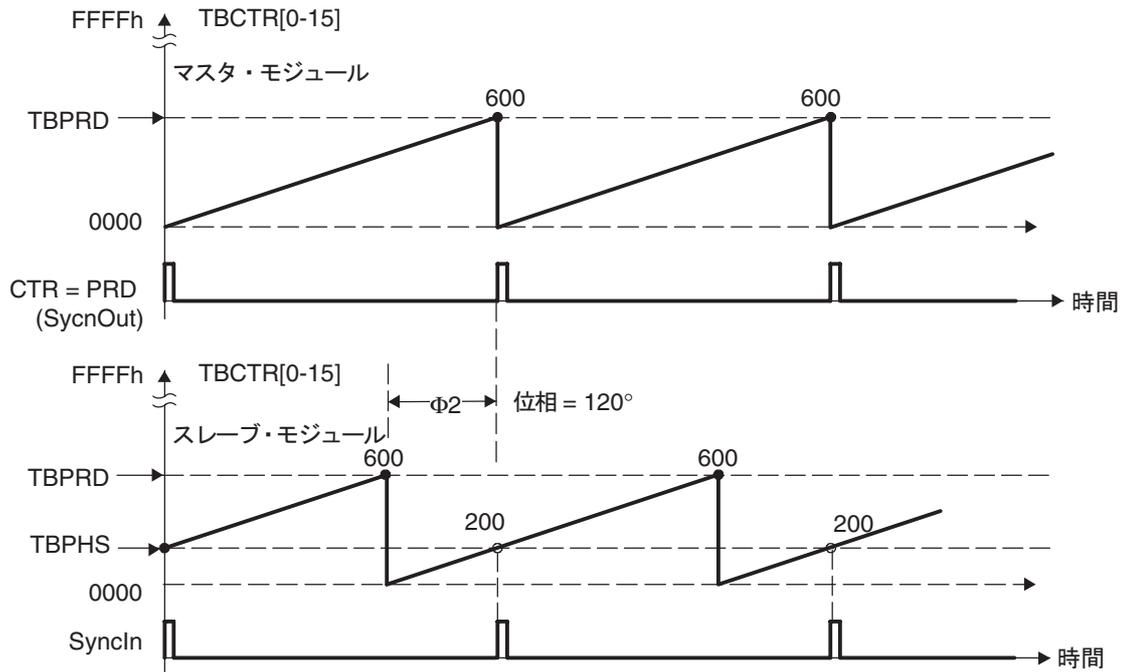


図 3-12 は、この設定に関連するタイミング波形を示しています。ここで、マスタとスレーブの TBPRD は、どちらも 600 です。スレーブの TBPHS は 200 (つまり、 $200/600 \times 360^\circ = 120^\circ$ ) です。マスタが Syncln パルスを生成するたびに (CTR=PRD)、TBPHS 値 =200 がスレーブの TBCTR レジスタにロードされます。そのため、スレーブのタイマ・カウンタは常に  $120^\circ$  だけマスタのタイマ・カウンタに先行します。

図 3-12 2つのモジュール間の位相制御に関連するタイミング波形



### 3.8 3相インターリーブ型DC/DCコンバータの制御

モジュール間の位相オフセットを使用する一般的なパワー・トポロジを図 3-13 に示します。このシステムでは 3 つの PWM モジュールを使用しており、モジュール 1 がマスタとして設定されています。動作するためには、隣接するモジュール間の位相関係が  $F=120^\circ$  でなければなりません。これを実現するには、周期値の  $1/3$  および  $2/3$  の値を使用してスレーブ TBPHS レジスタ 2 および 3 をそれぞれ設定します。たとえば、周期レジスタに 600 カウントの値がロードされた場合は、TBPHS (スレーブ 2) = 200、TBPHS (スレーブ 3) = 400 となります。どちらのスレーブ・モジュールもマスタ 1 モジュールと同期化されます。

この概念は、TBPHS 値の適切な設定によって 4 以上の位相に拡張できます。以下の式では、N フェーズ (位相) の TBPHS 値が算出されます。

$$TBPHS(N,M) = (TBPRD/N) \times (M-1)$$

以下に説明します。

N = 位相の数

M = PWM モジュール番号

たとえば、3 相 (N=3)、TBPRD=600 の場合は以下ようになります。

$$TBPHS(3,2) = (600/3) \times (2-1) = 200 \text{ (スレーブ・モジュール 2 の位相値)}$$

$$TBPHS(3,3) = 400 \text{ (スレーブ・モジュール 3 の位相値)}$$

図 3-14 は、図 3-13 の設定に対する波形です。

図 3-13 3 相インターリーブ型 DC/DC コンバータの制御

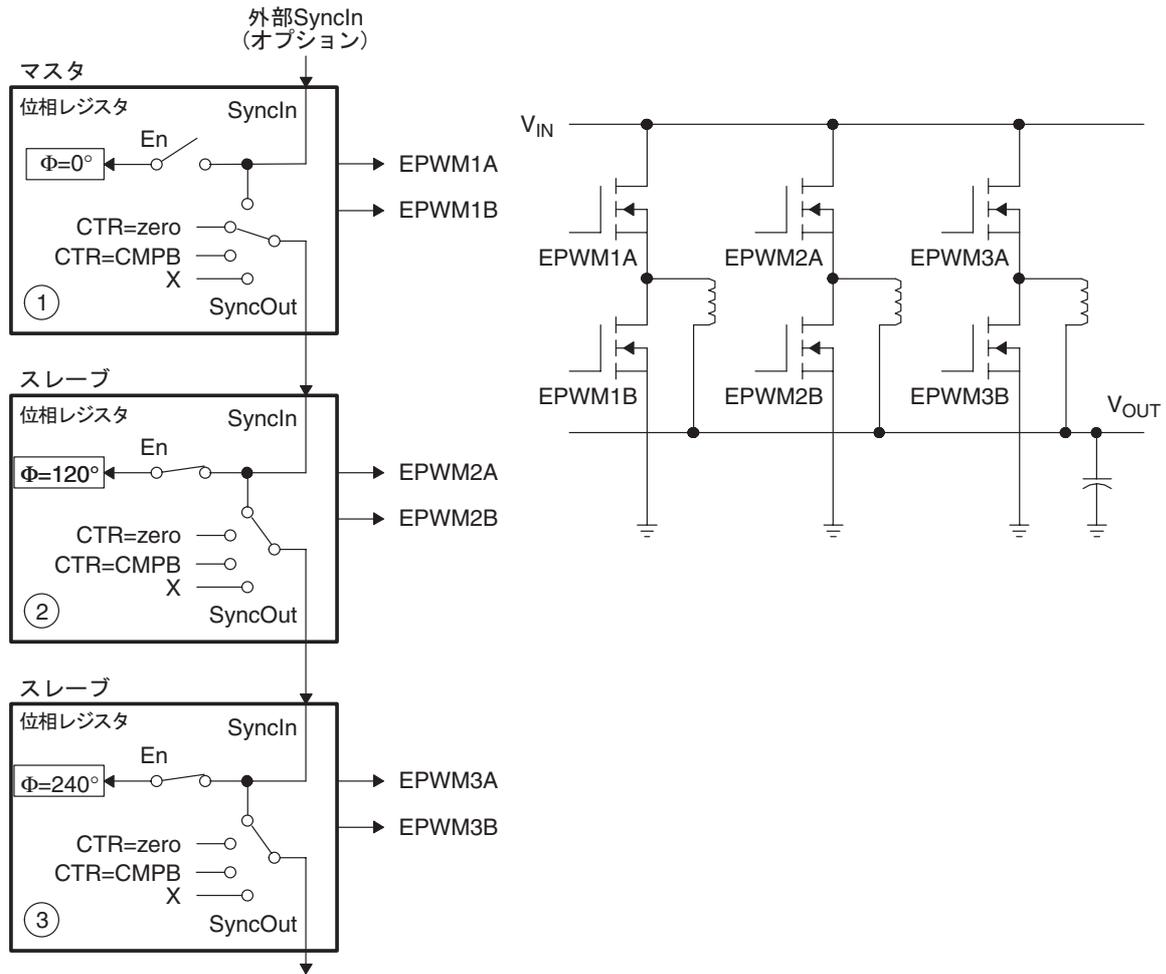
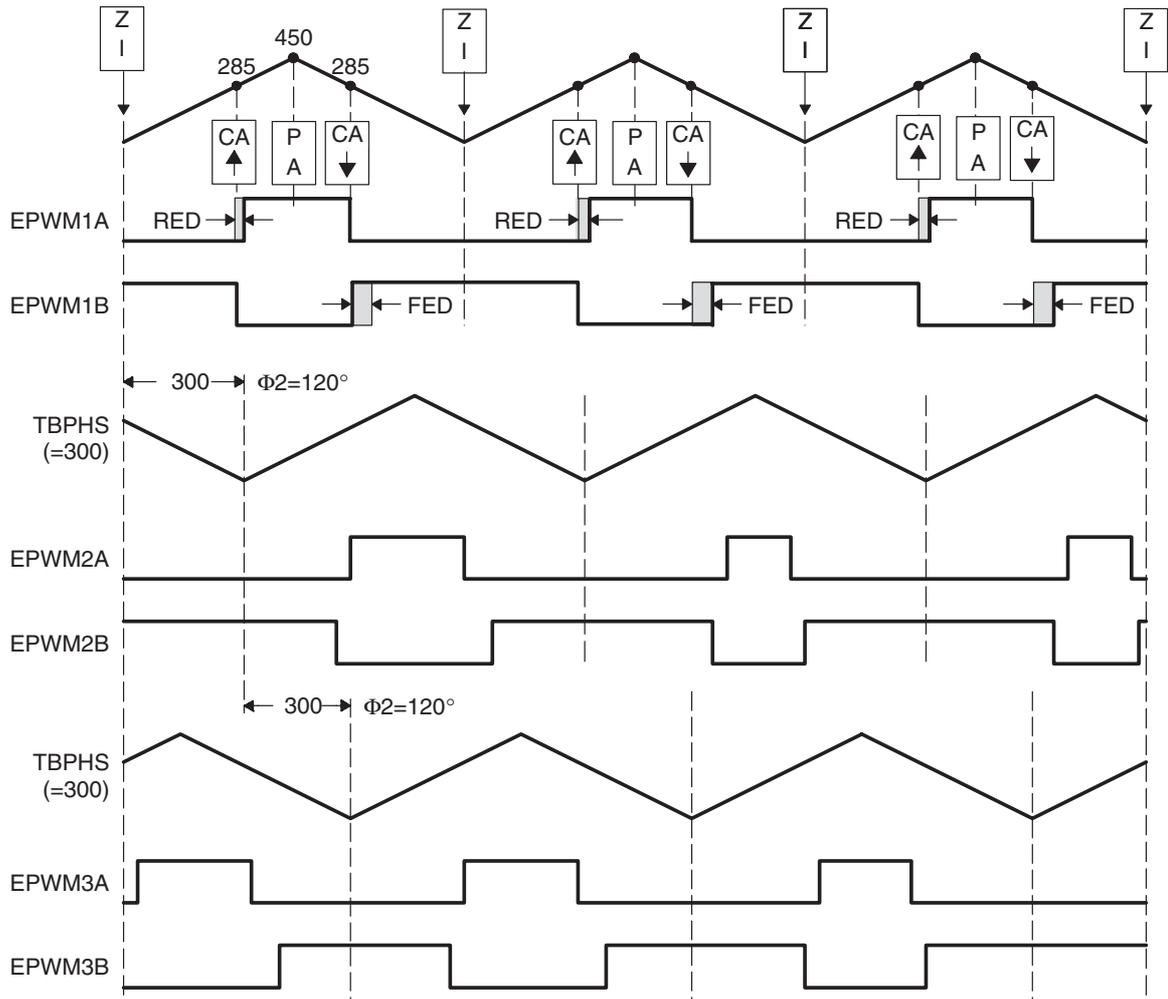


図 3-14 図 3-13 の 3 相インターリーブ型 DC/DC コンバータ 波形



例 3-5 図 3-13 に示した設定コード

```

//=====
// Config
// Initialization Time
//=====
// EPWM Module 1 config
EPwm1Regs.TBPRD = 450; // Period = 900 TBCLK counts
EPwm1Regs.TBPHS = 0; // Set Phase register to zero
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetrical mode
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Master module
EPwm1Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCSEL = TB_CTR_ZERO; // Sync down-stream module
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm1Regs.AQCTLA.bit.CAU = AQ_SET; // set actions for EPWM1A
EPwm1Regs.AQCTLA.bit.CAD = AQ_CLEAR;
EPwm1Regs.DBCTL.bit.MODE = DB_FULL_ENABLE; // enable Dead-band module
EPwm1Regs.DBCTL.bit.POLSEL = DB_ACTV_HIC; // Active Hi complementary
EPwm1Regs.DBFED = 20; // FED = 20 TBCLKs
EPwm1Regs.DBRED = 20; // RED = 20 TBCLKs
// EPWM Module 2 config
EPwm2Regs.TBPRD = 450; // Period = 900 TBCLK counts
EPwm2Regs.TBPHS = 300; // Phase = 300/900 * 360 = 120 deg
EPwm2Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetrical mode
EPwm2Regs.TBCTL.bit.PHSEN = TB_ENABLE; // Slave module
EPwm2Regs.TBCTL.bit.PHSDIR = TB_DOWN; // Count DOWN on sync (=120 deg)
EPwm2Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm2Regs.TBCTL.bit.SYNCSEL = TB_SYNC_IN; // sync flow-through
EPwm2Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm2Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm2Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm2Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm2Regs.AQCTLA.bit.CAU = AQ_SET; // set actions for EPWM2A
EPwm2Regs.AQCTLA.bit.CAD = AQ_CLEAR;
EPwm2Regs.DBCTL.bit.MODE = DB_FULL_ENABLE; // enable Dead-band module
EPwm2Regs.DBCTL.bit.POLSEL = DB_ACTV_HIC; // Active Hi Complementary
EPwm2Regs.DBFED = 20; // FED = 20 TBCLKs
EPwm2Regs.DBRED = 20; // RED = 20 TBCLKs
// EPWM Module 3 config
EPwm3Regs.TBPRD = 450; // Period = 900 TBCLK counts
EPwm3Regs.TBPHS = 300; // Phase = 300/900 * 360 = 120 deg
EPwm3Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetrical mode
EPwm3Regs.TBCTL.bit.PHSEN = TB_ENABLE; // Slave module
EPwm2Regs.TBCTL.bit.PHSDIR = TB_UP; // Count UP on sync (=240 deg)
EPwm3Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm3Regs.TBCTL.bit.SYNCSEL = TB_SYNC_IN; // sync flow-through
EPwm3Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm3Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm3Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm3Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm3Regs.AQCTLA.bit.CAU = AQ_SET; // set actions for EPWM3Ai
EPwm3Regs.AQCTLA.bit.CAD = AQ_CLEAR;
EPwm3Regs.DBCTL.bit.MODE = DB_FULL_ENABLE; // enable Dead-band module
EPwm3Regs.DBCTL.bit.POLSEL = DB_ACTV_HIC; // Active Hi complementary
EPwm3Regs.DBFED = 20; // FED = 20 TBCLKs
EPwm3Regs.DBRED = 20; // RED = 20 TBCLKs
// Run Time (Note: Example execution of one run-time instant)
//=====
EPwm1Regs.CMPA.half.CMPA = 285; // adjust duty for output EPWM1A
EPwm2Regs.CMPA.half.CMPA = 285; // adjust duty for output EPWM2A
EPwm3Regs.CMPA.half.CMPA = 285; // adjust duty for output EPWM3A

```

### 3.9 ZVSFB (Zero Voltage Switched Full Bridge : ゼロ電圧スイッチング・フル・ブリッジ) コンバータの制御

図 3-15 に示した例は、アーム (モジュール) 間の静的または一定のフェーズ関係を前提としています。そのようなケースでは、デューティ・サイクルの変調によって制御が実現されます。また、サイクルごとに位相を動的に変更することもできます。この機能は、フェーズシフト・フル・ブリッジまたはゼロ電圧スイッチング・フル・ブリッジと呼ばれるパワー・トポロジの制御に適しています。ここで制御されるパラメータは、デューティ・サイクルではなく (デューティ・サイクルは約 50% で固定される)、アーム間の位相関係です。このようなシステムは、2 つの PWM モジュールのリソースを割り当てて単一のパワー・ステージを制御することによって実装されます。また、そのパワー・ステージは 4 つのスイッチング素子の制御を必要とします。図 3-16 は、フル H ブリッジを制御するために同期化されたマスタ / スレーブ・モジュールの組み合わせを示しています。

この場合は、マスタ・モジュールとスレーブ・モジュールの両方を同じ PWM 周波数で切り替える必要があります。フェーズは、スレーブの位相レジスタ (TBPHS) で制御されます。マスタの位相レジスタは使用されないため、ゼロに初期化できます。

図 3-15 フル H ブリッジ・ステージの制御 ( $F_{PWM2}=F_{PWM1}$ )

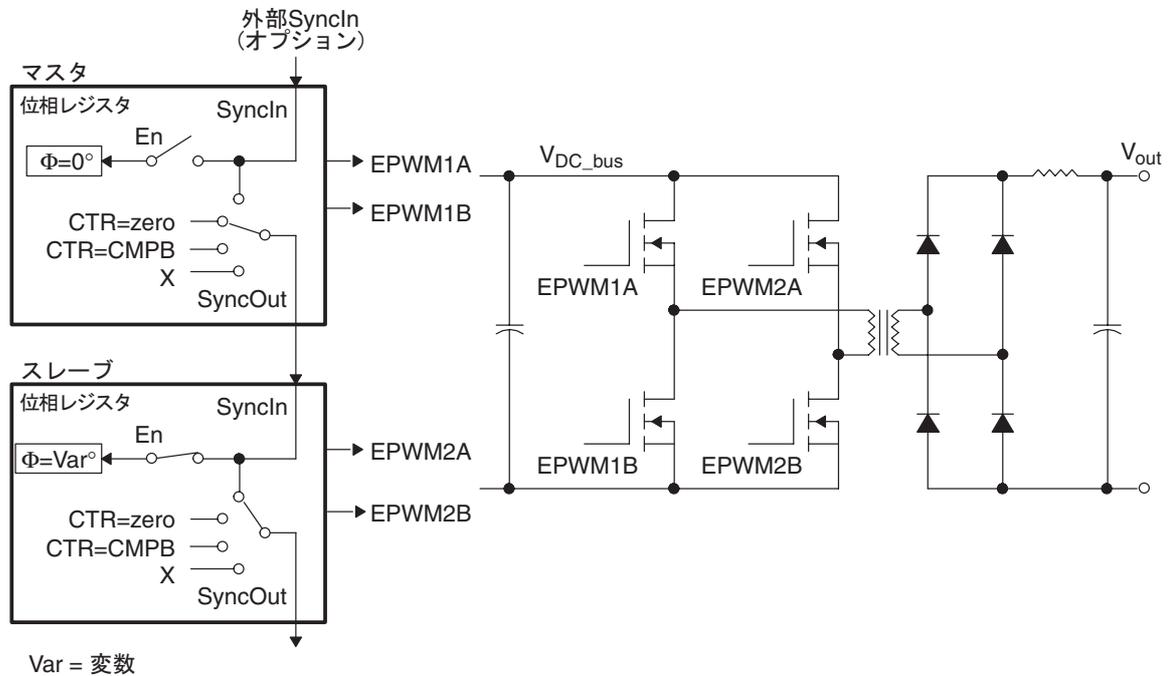
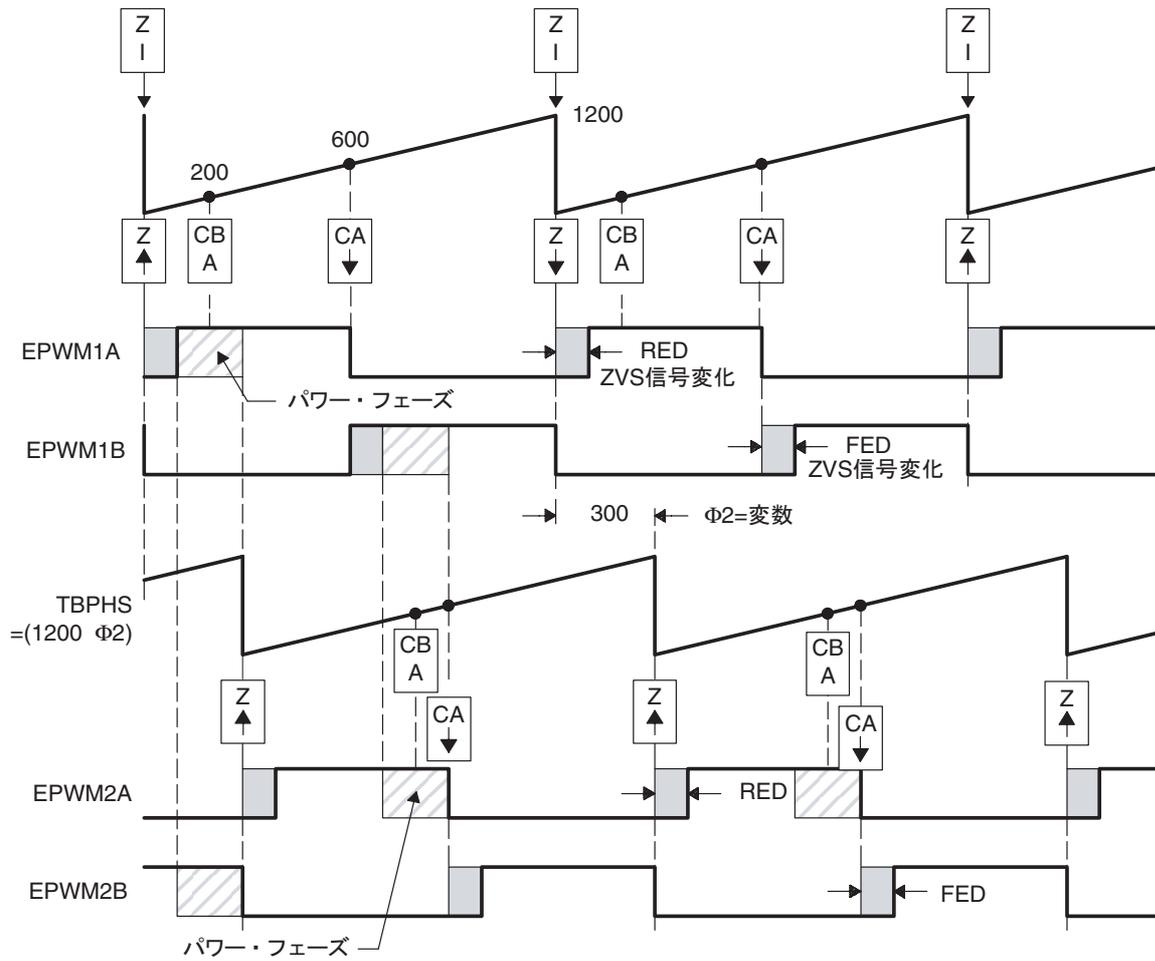


図 3-16 ZVS フル H ブリッジ波形



例 3-6 図 3-15 に示した設定コード

```

//=====
// Config
//=====
// Initialization Time
//=====
// EPWM Module 1 config
EPwm1Regs.TBPRD = 1200; // Period = 1201 TBCLK counts
EPwm1Regs.CMPA = 600; // Set 50% fixed duty for EPWM1A
EPwm1Regs.TBPHS = 0; // Set Phase register to zero
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP; // Asymmetrical mode
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Master module
EPwm1Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCSEL = TB_CTR_ZERO; // Sync down-stream module
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm1Regs.AQCTLA.bit.ZRO = AQ_SET; // set actions for EPWM1A
EPwm1Regs.AQCTLA.bit.CAU = AQ_CLEAR;
EPwm1Regs.DBCTL.bit.MODE = DB_FULL_ENABLE; // enable Dead-band module
EPwm1Regs.DBCTL.bit.POLSEL = DB_ACTV_HIC; // Active Hi complementary
EPwm1Regs.DBFED = 50; // FED = 50 TBCLKs initially
EPwm1Regs.DBRED = 70; // RED = 70 TBCLKs initially
// EPWM Module 2 config
EPwm2Regs.TBPRD = 1200; // Period = 1201 TBCLK counts
EPwm2Regs.CMPA.half.CMPA = 600; // Set 50% fixed duty EPWM2A
EPwm2Regs.TBPHS = 0; // Set Phase register to zero initially
EPwm2Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP; // Asymmetrical mode
EPwm2Regs.TBCTL.bit.PHSEN = TB_ENABLE; // Slave module
EPwm2Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm2Regs.TBCTL.bit.SYNCSEL = TB_SYNC_IN; // sync flow-through
EPwm2Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm2Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm2Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm2Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm2Regs.AQCTLA.bit.ZRO = AQ_SET; // set actions for EPWM2A
EPwm2Regs.AQCTLA.bit.CAU = AQ_CLEAR;
EPwm2Regs.DBCTL.bit.MODE = DB_FULL_ENABLE; // enable Dead-band module
EPwm2Regs.DBCTL.bit.POLSEL = DB_ACTV_HIC; // Active Hi complementary
EPwm2Regs.DBFED = 30; // FED = 30 TBCLKs initially
EPwm2Regs.DBRED = 40; // RED = 40 TBCLKs initially

// Run Time (Note: Example execution of one run-time instant)
//=====
EPwm2Regs.TBPHS = 1200-300; // Set Phase reg to 300/1200 * 360 = 90 deg
EPwm1Regs.DBFED = FED1_NewValue; // Update ZVS transition interval
EPwm1Regs.DBRED = RED1_NewValue; // Update ZVS transition interval
EPwm2Regs.DBFED = FED2_NewValue; // Update ZVS transition interval
EPwm2Regs.DBRED = RED2_NewValue; // Update ZVS transition interval
EPwm1Regs.CMPB = 200; // adjust point-in-time for ADCSOC trigger
  
```



## レジスタ

この章には、サブモジュールのレジスタ・レイアウトとビット説明が含まれています。

項目	ページ
4.1 タイム・ベース・サブモジュール・レジスタ .....	96
4.2 カウンタ・コンペア・サブモジュール・レジスタ .....	99
4.3 アクション選択サブモジュール・レジスタ .....	102
4.4 デッドバンド・サブモジュール・レジスタ .....	106
4.5 PWM チョップ・サブモジュール制御レジスタ .....	108
4.6 トリップ・ゾーン・サブモジュール制御 およびステータス・レジスタ .....	109
4.7 イベント・トリガ・サブモジュール・レジスタ .....	114
4.8 正しい割り込み初期化手順 .....	118

#### 4.1 タイム・ベース・サブモジュール・レジスタ

図 4-1 ~ 図 4-5 と表 4-1 ~ 表 4-5 にタイム・ベース・レジスタの定義を示します。

図 4-1 タイム・ベース周期レジスタ (TBPRD)

15	0
TBPRD	
R/W-0	

凡例：R/W= リード/ライト、R= リードのみ、-n= リセット後の値

表 4-1 タイム・ベース周期レジスタ (TBPRD) のフィールドの説明

ビット	名前	値	説明
15-0	TBPRD	0000-FFFF	<p>これらのビットでは、タイム・ベース・カウンタの周期が決定されます。これにより、PWM 周波数が設定されます。</p> <p>このレジスタのシャドウ化は、TBCTL[PRDL] ビットによってイネーブルまたはディスエーブルできます。</p> <p>デフォルトでは、このレジスタはシャドウ化されます。</p> <ul style="list-style-type: none"> <li>• TBCTL[PRDL]=0 の場合、シャドウは有効になり、ライトやリードは自動的にシャドウ・レジスタに対して実行されます。この場合、タイム・ベース・カウンタがゼロに等しくなると、アクティブ・レジスタがシャドウ・レジスタからロードされます。</li> <li>• TBCTL[PRDL]=1 の場合、シャドウはディスエーブルになり、ライトやリードはアクティブ・レジスタ（つまり、ハードウェアをアクティブに制御しているレジスタ）に対して直接実行されます。</li> <li>• アクティブ・レジスタとシャドウ・レジスタは同じメモリ・マップ・アドレスを共有します。</li> </ul>

図 4-2 タイム・ベース位相レジスタ (TBPHS)

15	0
TBPHS	
R/W-0	

凡例：R/W= リード/ライト、R= リードのみ、-n= リセット後の値

表 4-2 タイム・ベース位相レジスタ (TBPHS) のフィールドの説明

ビット	名前	値	説明
15-0	TBPHS	0000-FFFF	<p>これらのビットでは、同期入力信号を供給しているタイム・ベースを基準として、選択した ePWM のタイム・ベース・カウンタの位相が設定されます。</p> <ul style="list-style-type: none"> <li>• TBCTL[PHSEN]=0 の場合は、同期イベントが発生すると、タイム・ベース・カウンタ (TBCTR) に位相 (TBPHS) がロードされます。同期イベントは、入力同期信号 (EPWMxSYNCl) またはソフトウェア強制同期によって起動できます。</li> <li>• TBCTL[PHSEN]=1 の場合は、同期イベントが無視され、タイム・ベース・カウンタには位相がロードされません。</li> </ul>

図 4-3 タイム・ベース・カウンタ・レジスタ (TBCTR)

15	0
TBCTR	
R/W-0	

凡例：R/W= リード/ライト、R= リードのみ、-n= リセット後の値

表 4-3 タイム・ベース・カウンタ・レジスタ (TBCTR) のフィールドの説明

ビット	名前	値	説明
15-0	TBCTR	0000-FFFF	<p>これらのビットをリードすると、現在のタイム・ベース・カウンタ値が分かります。これらのビットにライトすると、現在のタイム・ベース・カウンタ値が設定されます。ライトが行うと更新がすぐに行われます。ライトはタイム・ベース・クロック (TBCLK) に対して同期されず、レジスタはシャドウ化されません。</p>

図 4-4 タイム・ベース制御レジスタ (TBCTL)

15	14	13	12	11	10	9	8
FREE, SOFT		PHSDIR	CLKDIV			HSPCLKDIV	
R/W-0		R/W-0	R/W-0			R/W-0,0,1	
7	6	5	4	3	2	1	0
HSPCLKDIV	SWFSYNC	SYNCOSSEL		PRDL	PHSEN	CTRMODE	
R/W-0,0,1	R/W-0	R/W-0		R/W-0	R/W-0	R/W-0	

凡例：R/W= リード/ライト、R= リードのみ、-n= リセット後の値

表 4-4 タイム・ベース制御レジスタ (TBCTL) のフィールドの説明

ビット	フィールド	値	説明
15:14	FREE、SOFT	00 01 1X	<p>エミュレーション・モード・ビット。これらのビットでは、エミュレーション・イベント時の ePWM タイム・ベース・カウンタの動作が選択されます。</p> <p>次のタイム・ベース・カウンタのインクリメントまたはデクリメント後に停止します。</p> <p>カウンタが全サイクルを完了したら停止します。</p> <ul style="list-style-type: none"> <li>・ アップ・カウント・モード：タイム・ベース・カウンタ = 周期 (TBCTR=TBPRD) の場合に停止</li> <li>・ ダウン・カウント・モード：タイム・ベース・カウンタ = 0x0000 (TBCTR=0x0000) の場合に停止</li> <li>・ アップ・ダウン・カウント・モード：タイム・ベース・カウンタ = 0x0000 (TBCTR=0x0000) の場合に停止</li> </ul> <p>フリー・ラン</p>
13	PHSDIR	0 1	<p>位相方向ビット</p> <p>このビットは、タイム・ベース・カウンタがアップ・ダウン・カウント・モードで設定されている場合にのみ使用します。</p> <p>PHSDIR ビットは、同期イベント発生後にタイム・ベース・カウンタ (TBCTR) がカウントする方向を示します。新しい位相値はフェーズ (TBPHS) レジスタからロードされます。これは、同期イベント以前のカウンタの方向とは無関係です。</p> <p>アップ・カウント・モードとダウン・カウント・モードでは、このビットは無視されます。</p> <p>0 同期イベント後にカウント・ダウンします。</p> <p>1 同期イベント後にカウント・アップします。</p>
12:10	CLKDIV	000 001 010 011 100 101 110 111	<p>タイム・ベース・クロック・プリスケール・ビット</p> <p>これらのビットでは、タイム・ベース・クロック・プリスケール値の一部が決定されます。</p> <p><math>TBCLK = SYSCLKOUT / (HSPCLKDIV \times CLKDIV)</math></p> <p>/1 (リセット時のデフォルト)</p> <p>/2</p> <p>/4</p> <p>/8</p> <p>/16</p> <p>/32</p> <p>/64</p> <p>/128</p>
9:7	HSPCLKDIV	000 001 010 011 100	<p>高速タイム・ベース・クロック・プリスケール・ビット</p> <p>これらのビットでは、タイム・ベース・クロック・プリスケール値の一部が決定されます。</p> <p><math>TBCLK = SYSCLKOUT / (HSPCLKDIV \times CLKDIV)</math></p> <p>この除数は、TMS320x281x システムのイベント・マネージャ (EV) ペリフェラルで使用される HSPCLK をエミュレートしています。</p> <p>/1</p> <p>/2 (リセット時のデフォルト)</p> <p>/4</p> <p>/6</p> <p>/8</p>

表 4-4 タイム・ベース制御レジスタ (TBCTL) のフィールドの説明 (続き)

ビット	フィールド	値	説明
		101	/10
		110	/12
		111	/14
6	SWFSYNC	0 1	ソフトウェア強制同期パルス 0 をライトしても何の影響もなく、リードは常に 0 を返します。 1 をライトすると、ワンタイム同期パルスが生成されます。 このイベントは、ePWM モジュールの EPWMxSYNCl 入力と論理和演算されます。 SWFSYNC は、SYNCOSEL=00 によって EPWMxSYNCl が選択されている場合にのみ有効です (機能します)。
5:4	SYNCOSEL	00 01 10 11	同期出力セレクト。これらのビットでは、EPWMxSYNCO 信号のソースが選択されます。 EPWMxSYNCl: 00 CTR=ゼロ: タイム・ベース・カウンタはゼロに等しい (TBCTR=0x0000) 01 CTR=CMPB: タイム・ベース・カウンタはカウンタ・コンペア A に等しい (TBCTR=CMPA) 10 EPWMxSYNCO 信号をディスエーブルにします。
3	PRDL	0 1	シャドウ・レジスタからのアクティブ周期レジスタ・ロードのセレクト 0 タイム・ベース・カウンタ (TBCTR) がゼロに等しい場合に、周期レジスタ (TBPRD) がシャドウ・レジスタからロードされます。 TBPRD レジスタのライトまたはリードを行うと、シャドウ・レジスタがアクセスされます。 1 シャドウ・レジスタを使わずに TBPRD レジスタをすぐにロードします。 TBPRD レジスタのライトまたはリードを行うと、アクティブ・レジスタが直接アクセスされます。
2	PHSEN	0 1	位相レジスタからのカウンタ・レジスタ・ロードのイネーブル 0 タイム・ベース・カウンタ (TBCTR) をタイム・ベース位相レジスタ (TBPHS) からロードしません。 1 EPWMxSYNCl 入力信号が発生した場合やソフトウェア同期が SWFSYNC ビットによって発生した場合は、タイム・ベース・カウンタに位相レジスタをロードします。
1:0	CTRMODE	00 01 10 11	カウンタ・モード 一般に、タイム・ベース・カウンタ・モードは 1 回設定され、通常動作中は変更されません。カウンタのモードを変更した場合、その変更は次の TBCLK エッジで有効になり、現在のカウンタ値はモード変更前の値からインクリメントまたはデクリメントされます。 これらのビットでは、以下のようにタイム・ベース・カウンタの動作モードが設定されます。 00 アップ・カウント・モード 01 ダウン・カウント・モード 10 アップ・ダウン・カウント・モード 11 カウンタ動作を停止 (リセット時のデフォルト)

図 4-5 タイム・ベース・ステータス・レジスタ (TBSTS)



凡例：R/W= リード/ライト、R= リードのみ、R/W1C=1 をリード/ライトしてクリア、-n= リセット後の値

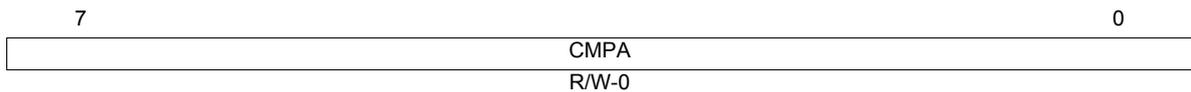
表 4-5 タイム・ベース・ステータス・レジスタ (TBSTS) のフィールドの説明

ビット	フィールド	値	説明
15:3	予約		予約
2	CTRMAX	0 1	タイム・ベース・カウンタの最大ラッチ・ステータス・ビット 0 のリードは、タイム・ベースがこの値に達していないことを意味します。0 をライトしても影響はありません。 1 のリードは、タイム・ベース・カウンタが最大値の 0xFFFF に達したことを意味します。このビットに 1 をライトすると、ラッチされたイベントがクリアされます。
1	SYNCI	0 1	入力同期ラッチ・ステータス・ビット 0 をライトしても影響はありません。0 のリードは、外部同期イベントが発生していないことを意味します。 1 のリードは、外部同期イベントが発生したことを意味します (EPWMxSYNCI)。このビットに 1 をライトすると、ラッチされたイベントがクリアされます。
0	CTRDIR	0 1	タイム・ベース・カウンタの方向ステータス・ビット 0 タイム・ベース・カウンタのカウント・ダウン中です。 1 タイム・ベース・カウンタのカウント・アップ中です。

## 4.2 カウンタ・コンペア・サブモジュール・レジスタ

図 4-6 ~ 図 4-8 と表 4-6 ~ 表 4-8 は、カウンタ・コンペア・サブモジュールの制御およびステータス・レジスタを示しています。

図 4-6 カウンタ・コンペア A レジスタ (CMPA) のフィールドの説明



凡例：R/W= リード/ライト、R= リードのみ、-n= リセット後の値

**表 4-6 カウンタ・コンペア A レジスタ (CMPA) のフィールドの説明**

ビット	名前	説明
15-0	CMPA	<p>アクティブ・CMPA レジスタの値は、タイム・ベース・カウンタ (TBCTR) と常にコンペアされます。値が等しい場合は、カウンタ・コンペア・モジュールが「タイム・ベース・カウンタはカウンタ・コンペア A に等しい」というイベントを生成します。このイベントは、アクション選択モジュールに送られ、そこで 1 つ以上のアクションに変換されます。これらのアクションは、AQCTLA および AQCTLB レジスタの設定に応じて EPWMxA または EPWMxB 出力に適用できます。AQCTLA および AQCTLB レジスタで定義できるアクションは以下のとおりです。</p> <ul style="list-style-type: none"> <li>• 何もしない：イベントは無視される</li> <li>• クリア：EPWMxA または EPWMxB 信号（あるいはその両方）を Low にする</li> <li>• セット：EPWMxA または EPWMxB 信号（あるいはその両方）を High にする</li> <li>• EPWMxA または EPWMxB 信号（あるいはその両方）をトグルする</li> </ul> <p>このレジスタのシャドウ化は、CMPCTL[SHDWAMODE] ビットによってイネーブルまたはディスエーブルにできます。デフォルトでは、このレジスタはシャドウ化されます。</p> <ul style="list-style-type: none"> <li>• CMPCTL[SHDWAMODE]=0 の場合、シャドウは有効になり、ライトやリードは自動的にシャドウ・レジスタに対して実行されます。この場合、CMPCTL[LOADAMODE] ビット・フィールドでは、シャドウ・レジスタからアクティブ・レジスタをロードするイベントが決定されます。</li> <li>• ライトの前には、CMPCTL[SHDWAFULL] ビットをリードして、シャドウ・レジスタが現在フルかどうかを確認できます。</li> <li>• CMPCTL[SHDWAMODE]=1 の場合、シャドウはディスエーブルになり、ライトやリードはアクティブ・レジスタ（つまり、ハードウェアをアクティブに制御しているレジスタ）に対して直接実行されます。</li> <li>• どちらのモードでも、アクティブ・レジスタとシャドウ・レジスタは同じメモリ・マップ・アドレスを共有します。</li> </ul>

**図 4-7 カウンタ・コンペア B レジスタ (CMPB)**

15	CMPB	0
R/W-0		

凡例：R/W= リード/ライト、R= リードのみ、-n= リセット後の値

表 4-7 カウンタ・コンペア B レジスタ (CMPB) のフィールドの説明

ビット	名前	説明
15-0	CMPB	<p>アクティブ・CMPB レジスタの値は、タイム・ベース・カウンタ (TBCTR) と常にコンペアされます。値が等しい場合は、カウンタ・コンペア・モジュールが「タイム・ベース・カウンタはカウンタ・コンペア A に等しい」というイベントを生成します。このイベントは、アクション選択モジュールに送られ、そこで 1 つ以上のアクションに変換されます。これらのアクションは、AQCTLA および AQCTLB レジスタの設定に応じて EPWMxA または EPWMxB 出力に適用できます。AQCTLA および AQCTLB レジスタで定義できるアクションは以下のとおりです。</p> <ul style="list-style-type: none"> <li>・ 何もしない：イベントは無視される</li> <li>・ クリア：EPWMxA または EPWMxB 信号（あるいはその両方）を Low にする</li> <li>・ セット：EPWMxA または EPWMxB 信号（あるいはその両方）を High にする</li> <li>・ EPWMxA または EPWMxB 信号（あるいはその両方）をトグルする</li> </ul> <p>コンペア B レジスタ・セット</p> <p>このレジスタのシャドウ化は、CMPCTL[SHDWBMODE] ビットによってイネーブルまたはディスエーブルにできます。デフォルトでは、このレジスタはシャドウ化されます。</p> <ul style="list-style-type: none"> <li>・ CMPCTL[SHDWBMODE]=0 の場合、シャドウは有効になり、ライトやリードは自動的にシャドウ・レジスタに対して実行されます。この場合、CMPCTL[LOADBMODE] ビット・フィールドでは、シャドウ・レジスタからアクティブ・レジスタをロードするイベントが決定されます。</li> <li>・ ライトの前には、CMPCTL[SHDWBFULL] ビットをリードして、シャドウ・レジスタが現在フルかどうかを確認できます。</li> <li>・ CMPCTL[SHDWBMODE]=1 の場合、シャドウはディスエーブルになり、ライトやリードはアクティブ・レジスタ（つまり、ハードウェアをアクティブに制御しているレジスタ）に直接実行されます。</li> <li>・ どちらのモードでも、アクティブ・レジスタとシャドウ・レジスタは同じメモリ・マップ・アドレスを共有します。</li> </ul>

図 4-8 カウンタ・コンペア制御レジスタ (CMPCTL)

15				10				9	8						
予約								SHDWBFULL	SHDWAFULL						
R-0								R-0	R-0						
7		6		5		4		3		2		1		0	
予約		SHDWBMODE		予約		SHDWAMODE		LOADBMODE				LOADAMODE			
R-0		R/W-0		R-0		R/W-0		R/W-0				R/W-0			

凡例：R/W= リード/ライト、R= リードのみ、-n= リセット後の値

表 4-8 カウンタ・コンペア制御レジスタ (CMPCTL) のフィールドの説明

ビット	名前	値	説明
15-10	予約		予約
9	SHDWBFULL	0 1	<p>カウンタ・コンペア B (CMPB) シャドウ・レジスタのフル・ステータス・フラグ このビットは、ロードが発生すると自動的にクリアされます。</p> <p>0 CMPB シャドウ FIFO はまだフルになっていません。</p> <p>1 CMPB シャドウ FIFO がフルであることを意味します。CPU ライトによって現在のシャドウ値が上書きされます。</p>
8	SHDWAFULL	0 1	<p>カウンタ・コンペア A (CMPA) シャドウ・レジスタのフル・ステータス・フラグ CMPA:CMPAHR レジスタへの 32 ビット・ライト、または CMPA レジスタへの 16 ビット・ライトを行うと、フラグ・ビットがセットされます。CMPAHR レジスタへの 16 ビット・ライトはフラグに影響しません。</p> <p>このビットは、ロードが発生すると自動的にクリアされます。</p> <p>0 CMPA シャドウ FIFO はまだフルになっていません。</p> <p>1 CMPA シャドウ FIFO がフルであることを意味します。CPU ライトによって現在のシャドウ値が上書きされます。</p>
7	予約		予約

ビット	名前	値	説明
6	SHDWBMODE	0	カウンタ・コンペア B (CMPB) レジスタの動作モード シャドウ・モード。ダブル・バッファとして動作します。CPU によるライトでは、シャドウ・レジスタがアクセスされます。
		1	即時モード。アクティブ・コンペア B レジスタだけが使用されます。ライトおよびリードでは、即時コンペア・アクションのためにアクティブ・レジスタが直接アクセスされます。
5	予約		予約
4	SHDWAMODE	0	カウンタ・コンペア A (CMPA) レジスタの動作モード シャドウ・モード。ダブル・バッファとして動作します。CPU によるライトでは、シャドウ・レジスタがアクセスされます。
		1	即時モード。アクティブ・コンペア・レジスタだけが使用されます。ライトおよびリードでは、即時コンペア・アクションのためにアクティブ・レジスタが直接アクセスされます。
3-2	LOADBMODE		シャドウからのアクティブ・カウンタ・コンペア B (CMPB) ロードのセレクト・モード 即時モード (CMPCTL[SHDWBMODE]=1) では、このビットは無効です。
		00	CTR=Zero でロード：タイム・ベース・カウンタはゼロに等しい (TBCTR=0x0000)
		01	CTR=PRD でロード：タイム・ベース・カウンタは周期に等しい (TBCTR=TBPRD)
		10	CTR=Zero または CTR=PRD でロード
	11	フリーズ (ロード不可)	
1-0	LOADAMODE		シャドウからのアクティブ・カウンタ・コンペア A (CMPA) ロードのセレクト・モード 即時モード (CMPCTL[SHDWAMODE]=1) では、このビットは無効です。
		00	CTR=Zero でロード：タイム・ベース・カウンタはゼロに等しい (TBCTR=0x0000)
		01	CTR=PRD でロード：タイム・ベース・カウンタは周期に等しい (TBCTR=TBPRD)
		10	CTR=Zero または CTR=PRD でロード
	11	フリーズ (ロード不可)	

### 4.3 アクション選択サブモジュール・レジスタ

図 4-9 ~ 図 4-12 と表 4-9 ~ 表 4-12 にアクション選択サブモジュールのレジスタの定義を示します。

図 4-9 アクション選択出力 A 制御レジスタ (AQCTLA)

15		12	11	10	9	8
予約 R-0			CBD R/W-0		CBU R/W-0	
7	6	5	4	3	2	1 0
CAD R/W-0		CAU R/W-0		PRD R/W-0		ZRO R/W-0

凡例：R/W= リード/ライト、R= リードのみ、-n= リセット後の値

表 4-9 アクション選択出力 A 制御レジスタ (AQCTLA) のフィールドの説明

ビット	名前	値	説明
15-12	予約		予約
11-10	CBD	00	デクリメント中のタイム・ベース・カウンタがアクティブ CMPB レジスタに等しい場合のアクション 何もしない (アクションがディスエーブルになる) にします。
		01	クリア：EPWMxA 出力を Low にします。
		10	セット：EPWMxA 出力を High にします。
		11	EPWMxA 出力をトグル：Low 出力信号を High にし、High 信号を Low にします。

ビット	名前	値	説明
9-8	CBU	00 01 10 11	インクリメント中のカウンタがアクティブ CMPB レジスタに等しい場合のアクション 何もしない（アクションがディスエーブルになる）にします。 クリア：EPWMxA 出力を Low にします。 セット：EPWMxA 出力を High にします。 EPWMxA 出力をトグル：Low 出力信号を High にし、High 信号を Low にします。
7-6	CAD	00 01 10 11	デクリメント中のカウンタがアクティブ CMPA レジスタに等しい場合のアクション 何もしない（アクションがディスエーブルになる）にします。 クリア：EPWMxA 出力を Low にします。 セット：EPWMxA 出力を High にします。 EPWMxA 出力をトグル：Low 出力信号を High にし、High 信号を Low にします。
5-4	CAU	00 01 10 11	インクリメント中のカウンタがアクティブ CMPA レジスタに等しい場合のアクション 何もしない（アクションがディスエーブルになる）にします。 クリア：EPWMxA 出力を Low にします。 セット：EPWMxA 出力を High にします。 EPWMxA 出力をトグル：Low 出力信号を High にし、High 信号を Low にします。
3-2	PRD	00 01 10 11	カウンタが周期に等しい場合のアクション 注：カウント・アップ・ダウン・モードでカウンタが周期に等しい場合、方向は 0 またはカウント・ダウンとして定義されます。 何もしない（アクションがディスエーブルになる）にします。 クリア：EPWMxA 出力を Low にします。 セット：EPWMxA 出力を High にします。 EPWMxA 出力をトグル：Low 出力信号を High にし、High 信号を Low にします。
1-0	ZRO	00 01 10 11	カウンタがゼロに等しい場合のアクション 注：カウント・アップ・ダウン・モードでカウンタが 0 に等しい場合、方向は 1 またはカウント・アップとして定義されます。 何もしない（アクションがディスエーブルになる）にします。 クリア：EPWMxA 出力を Low にします。 セット：EPWMxA 出力を High にします。 EPWMxA 出力をトグル：Low 出力信号を High にし、High 信号を Low にします。

図 4-10 アクション選択出力 B 制御レジスタ (AQCTLB)

15	予約			12	11	10	9	8
	R-0			CBD		CBU		
	R/W-0			R/W-0		R/W-0		
7	6	5	4	3	2	1	0	
CAD		CAU		PRD		ZRO		
R/W-0		R/W-0		R/W-0		R/W-0		

凡例：R/W= リード/ライト、R= リードのみ、-n= リセット後の値

表 4-10 アクション選択出力 B 制御レジスタ (AQCTLB) のフィールドの説明

ビット	名前	値	説明
15-12	予約		予約
11-10	CBD	00 01 10 11	デクリメント中のカウンタがアクティブ CMPB レジスタに等しい場合のアクション 何もしない（アクションがディスエーブルになる）にします。 クリア：EPWMxB 出力を Low にします。 セット：EPWMxB 出力を High にします。 EPWMxB 出力をトグル：Low 出力信号を High にし、High 信号を Low にします。

ビット	名前	値	説明
9-8	CBU	00 01 10 11	インクリメント中のカウンタがアクティブCMPBレジスタに等しい場合のアクション 何もしない（アクションがディスエーブルになる）にします。 クリア：EPWMxB 出力を Low にします。 セット：EPWMxB 出力を High にします。 EPWMxB 出力をトグル：Low 出力信号を High にし、High 信号を Low にします。
7-6	CAD	00 01 10 11	デクリメント中のカウンタがアクティブCMPAレジスタに等しい場合のアクション 何もしない（アクションがディスエーブルになる）にします。 クリア：EPWMxB 出力を Low にします。 セット：EPWMxB 出力を High にします。 EPWMxB 出力をトグル：Low 出力信号を High にし、High 信号を Low にします。
5-4	CAU	00 01 10 11	インクリメント中のカウンタがアクティブCMPAレジスタに等しい場合のアクション 何もしない（アクションがディスエーブルになる）にします。 クリア：EPWMxB 出力を Low にします。 セット：EPWMxB 出力を High にします。 EPWMxB 出力をトグル：Low 出力信号を High にし、High 信号を Low にします。
3-2	PRD	00 01 10 11	カウンタが周期に等しい場合のアクション 注：カウント・アップ・ダウン・モードでカウンタが周期に等しい場合、方向は0またはカウント・ダウンとして定義されます。 何もしない（アクションがディスエーブルになる）にします。 クリア：EPWMxB 出力を Low にします。 セット：EPWMxB 出力を High にします。 EPWMxB 出力をトグル：Low 出力信号を High にし、High 信号を Low にします。
1-0	ZRO	00 01 10 11	カウンタがゼロに等しい場合のアクション 注：カウント・アップ・ダウン・モードでカウンタが0に等しい場合、方向は1またはカウント・アップとして定義されます。 何もしない（アクションがディスエーブルになる）にします。 クリア：EPWMxB 出力を Low にします。 セット：EPWMxB 出力を High にします。 EPWMxB 出力をトグル：Low 出力信号を High にし、High 信号を Low にします。

図 4-11 アクション選択ソフトウェア強制レジスタ (AQSFR)C

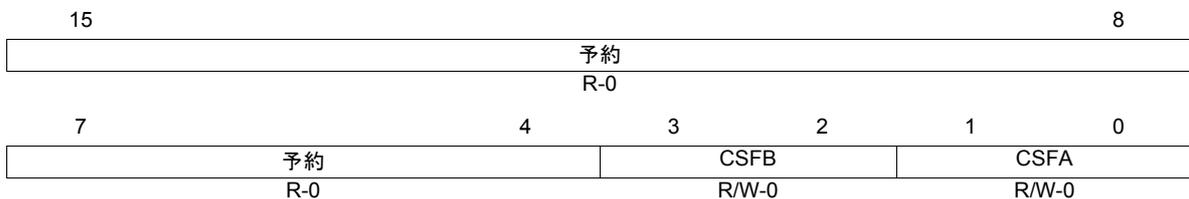


凡例：R/W= リード/ライト、R= リードのみ、-n= リセット後の値

表 4-11 アクション選択ソフトウェア強制レジスタ (AQSFRC) のフィールドの説明

ビット	名前	値	説明
15-8	予約		
7:6	RLDCSF	00 01 10 11	シャドウからの AQCSF アクティブ・レジスタ・リロードのオプション 「カウンタがゼロに等しい」 イベントでロードします。 「カウンタが周期に等しい」 イベントでロードします。 「カウンタはゼロに等しい」 または 「カウンタは周期に等しい」 イベントでロードします。 すぐにロードします (アクティブ・レジスタは CPU によって直接アクセスされ、シャドウ・レジスタからはロードされない)。
5	OTSFB	0 1	出力 B でのワнтаイム・ソフトウェア強制イベント 0 (ゼロ) をライトしても影響はありません。常に 0 をリード・バックします。 このビットは、このレジスタへのライトが完了すると (つまり、強制イベントが開始されると) 自動的にクリアされます。これはワンショット強制イベントであり、出力 B の別の後続イベントによってさらに起動することができます。 1 単発のソフトウェア強制イベントを起動します。
4:3	ACTSFB	00 01 10 11	ワнтаイム・ソフトウェア強制 B の起動時のアクション 何もしない (アクションがディスエーブルになる) にします。 クリア (Low) セット (High) トグル (Low → High、High → Low) 注: このアクションは、カウンタ方向 (CNT_dir) は関係ありません。
2	OTSFA	0 1	出力 A でのワнтаイム・ソフトウェア強制イベント 0 (ゼロ) をライトしても影響はありません。常に 0 をリード・バックします。 このビットは、このレジスタへのライトが完了すると (つまり、強制イベントが開始されると) 自動的にクリアされます。 1 単発のソフトウェア強制イベントを起動します。
1:0	ACTSFA	00 01 10 11	ワнтаイム・ソフトウェア強制 A の起動時のアクション 何もしない (アクションがディスエーブルになる) にします。 クリア (Low) セット (High) トグル (Low → High、High → Low) 注: このアクションは、カウンタ方向 (CNT_dir) は関係ありません。

図 4-12 アクション選択連続ソフトウェア強制レジスタ (AQCSFRC)



凡例: R/W= リード/ライト、R= リードのみ、-n= リセット後の値

表 4-12 アクション選択連続ソフトウェア強制レジスタ (AQCSFRC) のフィールドの説明

ビット	名前	値	説明
15-4	予約		予約
3-2	CSFB	00 01 10 11	出力 B での連続ソフトウェア強制 即時モードでは、次の TBCLK エッジで連続強制が有効になります。 シャドウ・モードでは、アクティブ・レジスタへのシャドウ・ロード後に次の TBCLK エッジで連続強制が有効になります。 00 強制がディスエーブルになります (つまり、影響を及ぼさない)。 01 出力 B で連続 Low を強制します。 10 出力 B で連続 High を強制します。 11 ソフトウェア強制はディスエーブルになり、何の影響も及ぼしません。
1-0	CSFA	00 01 10 11	出力 A での連続ソフトウェア強制 即時モードでは、次の TBCLK エッジで連続強制が有効になります。 シャドウ・モードでは、アクティブ・レジスタへのシャドウ・ロード後に次の TBCLK エッジで連続強制が有効になります。 00 強制がディスエーブルになります (つまり、影響を及ぼさない)。 01 出力 A で連続 Low を強制します。 10 出力 A で連続 High を強制します。 11 ソフトウェア強制はディスエーブルになり、何の影響も及ぼしません。

#### 4.4 デッドバンド・サブモジュール・レジスタ

図 4-13 ~ 図 4-15 と表 4-13 ~ 表 4-15 にレジスタの定義を示します。

図 4-13 デッドバンド・ジェネレータ制御レジスタ (DBCTL)

15	予約				8
	R-0				
7	5	4	3	2	1 0
予約	IN_MODE		POLSEL		OUT_MODE
R-0	R/W-0		R/W-0		R/W-0

凡例：R/W= リード/ライト、R= リードのみ、-n= リセット後の値

表 4-13 デッドバンド・ジェネレータ制御レジスタ (DBCTL) のフィールドの説明

ビット	名前	値	説明
15-6	予約		予約
5-4	OUT_MODE	S1 S0 00 01	デッドバンド出力モード制御 図 2-27 に示すように、ビット 5 は S1 スイッチ、ビット 4 は S0 スイッチを制御します。 これにより、立ち下がりがエッジおよび立ち上がりエッジ遅延についてのデッドバンド生成を、イネーブルまたはバイパスすることができます。 00 両方の出力信号についてデッドバンド生成がバイパスされます。このモードでは、アクション選択モジュールからの EPWMxA 出力信号と EPWMxB 出力信号がどちらも PWM チョップ・サブモジュールに直接渡されます。 このモードでは、POLSEL および IN_MODE ビットは何の影響も及ぼしません。 01 立ち上がりエッジ遅延をディスエーブルにします。アクション選択モジュールからの EPWMxA 信号は、PWM チョップ・サブモジュールの EPWMxA 入力に直接渡されます。立ち下がりがエッジ遅延信号が EPWMxB 出力で検出されます。遅延の入力信号は DBCTL[IN_MODE] によって決定されます。

表 4-13 デッドバンド・ジェネレータ制御レジスタ (DBCTL) のフィールドの説明 (続き)

ビット	名前	値	説明
		10	立ち上がりエッジ遅延信号が EPWMxA 出力で検出されます。遅延の入力信号は DBCTL[IN_MODE]によって決定されます。
		11	立ち下がりエッジ遅延をディスエーブルにします。アクション選択モジュールからの EPWMxA信号は、PWMチョッパ・サブモジュールのEWPMxA入力に直接渡されます。 EPWMxA 出力の立ち上がりエッジ遅延と EPWMxB 出力の立ち下がりエッジ遅延の両方についてデッドバンドが完全にイネーブルになります。遅延の入力信号は DBCTL[IN_MODE]によって決定されます。
3-2	POLSEL	S3 S2	極性選択制御 図 2-27 に示すように、ビット 3 は S3 スイッチ、ビット 2 は S2 スイッチを制御します。これにより、デッドバンド・サブモジュールから送出する前に、いずれかの遅延信号を反転することができます。 以下の説明は、デジタル・モータ制御インバータのアーム制御でよく用いられる上位 / 下位スイッチ制御に相当します。 これらは、DBCTL[OUT_MODE]=1,1、および DBCTL[IN_MODE]=0,0 であることを前提としています。他の高度なモードも可能ですが、一般的な使用モードとは考えられていません。
		00	アクティブ・ハイ (AH) モード。EPWMxA も EPWMxB も反転されません (デフォルト)。
		01	アクティブ・ロー相補 (ALC) モード。EPWMxA が反転されます。
		10	アクティブ・ハイ相補 (AHC)。EPWMxB が反転されます。
		11	アクティブ・ロー (AL) モード。EPWMxA と EPWMxB がどちらも反転されます。
1-0	IN_MODE	S5 S4	デッドバンド入力モード制御 図 2-27 に示すように、ビット 1 は S5 スイッチ、ビット 0 は S4 スイッチを制御します。これにより、立ち下がりエッジおよび立ち上がりエッジ遅延に対する入力ソースを選択できます。 従来からよく用いられる一般的なデッドバンド波形を生成する場合、デフォルトでは EPWMxA In が立ち下がりエッジ遅延と立ち上がりエッジ遅延の両方に対するソースとなります。
		00	EPWMxA In (アクション選択からの入力) は、立ち下がりエッジと立ち上がりエッジ遅延の両方に対するソースです。
		01	EPWMxB In (アクション選択からの入力) は、立ち上がりエッジ遅延信号のソースです。 EPWMxA In (アクション選択からの入力) は、立ち下がりエッジ遅延信号のソースです。
		10	EPWMxA In (アクション選択からの入力) は、立ち上がりエッジ遅延信号のソースです。 EPWMxB In (アクション選択からの入力) は、立ち下がりエッジ遅延信号のソースです。
		11	EPWMxB In (アクション選択からの入力) は、立ち上がりエッジ遅延信号と立ち下がりエッジ遅延信号の両方に対するソースです。

図 4-14 デッドバンド・ジェネレータ立ち上がりエッジ遅延レジスタ (DBRED)

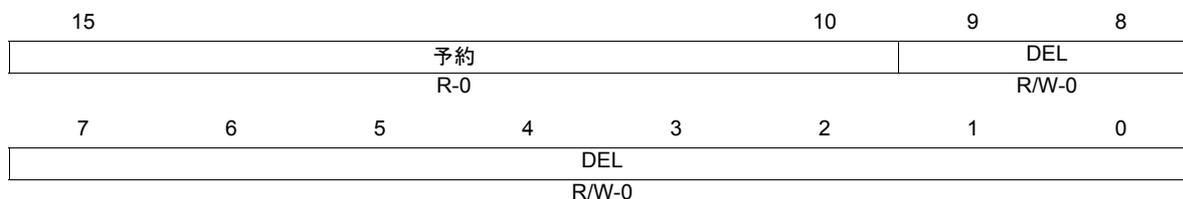
15						10	9	8
予約							DEL	
R-0							R/W-0	
7	6	5	4	3	2	1	0	
DEL								
R/W-0								

凡例 : R/W= リード / ライト、R= リードのみ、-n= リセット後の値

表 4-14 デッドバンド・ジェネレータ立ち上がりエッジ遅延レジスタ (DBRED) のフィールドの説明

ビット	名前	値	説明
15-4	予約		予約
9-0	DEL		立ち上がりエッジ遅延カウンタ、10 ビット・カウンタ

図 4-15 デッドバンド・ジェネレータ立ち下がりエッジ遅延レジスタ (DBFED)



凡例：R/W= リード/ライト、R= リードのみ、-n= リセット後の値

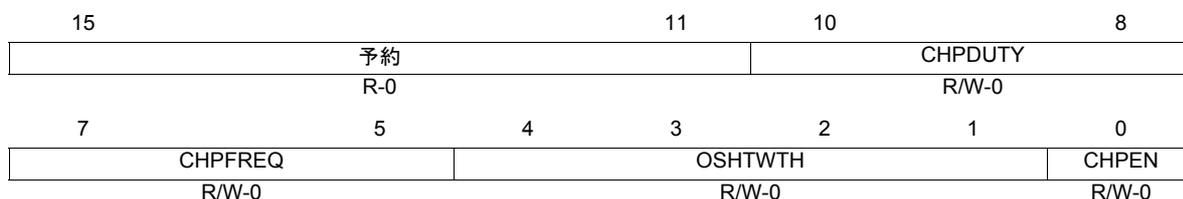
表 4-15 デッドバンド・ジェネレータ立ち下がりエッジ遅延レジスタ (DBFED) のフィールドの説明

ビット	名前	値	説明
15-4	予約		予約
9-0	DEL		立ち下がりエッジ遅延カウンタ、10 ビット・カウンタ

#### 4.5 PWM チョップパ・サブモジュール制御レジスタ

図 4-16 と表 4-16 に PWM チョップパ・サブモジュール制御レジスタの定義を示します。

図 4-16 PWM チョップパ制御レジスタ (PCCTL)



凡例：R/W= リード/ライト、R= リードのみ、-n= リセット後の値

表 4-16 PWM チョップパ制御レジスタ (PCCTL) のビットの説明

ビット	名前	値	説明
15-11	予約		予約
10-8	CHPDUTY	000	チョッピング・クロック・デューティ・サイクル デューティ = 1/8 (12.5%)
		001	デューティ = 2/8 (25.0%)
		010	デューティ = 3/8 (37.5%)
		011	デューティ = 4/8 (50.0%)
		100	デューティ = 5/8 (62.5%)
		101	デューティ = 6/8 (75.0%)
		110	デューティ = 7/8 (87.5%)
		111	予約
7:5	CHPFREQ	000	チョッピング・クロック周波数 1 分周 (プリスケールなし、=100MHz SYSCLKOUT で 12.5MHz)
		001	2 分周 (100MHz SYSCLKOUT で 6.25MHz)

表 4-16 PWM チョップ制御レジスタ (PCCTL) のビットの説明 (続き)

ビット	名前	値	説明
		010	3 分周 (100MHz SYSCLKOUT で 4.16MHz)
		011	4 分周 (100MHz SYSCLKOUT で 3.12MHz)
		100	5 分周 (100MHz SYSCLKOUT で 2.50MHz)
		101	6 分周 (100MHz SYSCLKOUT で 2.08MHz)
		110	7 分周 (100MHz SYSCLKOUT で 1.78MHz)
		111	8 分周 (100MHz SYSCLKOUT で 1.56MHz)
4:1	SHTWTH		ワンショット・パルス幅
		0000	1 × SYSCLKOUT/8 幅 (=100MHz SYSCLKOUT で 80nS)
		0001	2 × SYSCLKOUT/8 幅 (=100MHz SYSCLKOUT で 160nS)
		0010	3 × SYSCLKOUT/8 幅 (=100MHz SYSCLKOUT で 240nS)
		0011	4 × SYSCLKOUT/8 幅 (=100MHz SYSCLKOUT で 320nS)
		0100	5 × SYSCLKOUT/8 幅 (=100MHz SYSCLKOUT で 400nS)
		0101	6 × SYSCLKOUT/8 幅 (=100MHz SYSCLKOUT で 480nS)
		0110	7 × SYSCLKOUT/8 幅 (=100MHz SYSCLKOUT で 560nS)
		0111	8 × SYSCLKOUT/8 幅 (=100MHz SYSCLKOUT で 640nS)
		1000	9 × SYSCLKOUT/8 幅 (=100MHz SYSCLKOUT で 720nS)
		1001	10 × SYSCLKOUT/8 幅 (=100MHz SYSCLKOUT で 800nS)
		1010	11 × SYSCLKOUT/8 幅 (=100MHz SYSCLKOUT で 880nS)
		1011	12 × SYSCLKOUT/8 幅 (=100MHz SYSCLKOUT で 960nS)
		1100	13 × SYSCLKOUT/8 幅 (=100MHz SYSCLKOUT で 1040nS)
		1101	14 × SYSCLKOUT/8 幅 (=100MHz SYSCLKOUT で 1120nS)
		1110	15 × SYSCLKOUT/8 幅 (=100MHz SYSCLKOUT で 1200nS)
		1111	16 × SYSCLKOUT/8 幅 (=100MHz SYSCLKOUT で 1280nS)
0	CHPENPWM		チョッピング・イネーブル
		0	PWM チョッピング機能をディスエーブル (バイパス) します。
		1	チョッピング機能をイネーブルにします。

#### 4.6 トリップ・ゾーン・サブモジュール制御およびステータス・レジスタ

図 4-17 と表 4-17 にトリップ・ゾーン制御およびステータス・レジスタの定義を示します。

図 4-17 トリップ・ゾーン・セレクト・レジスタ (TZSEL)

15	14	13	12	11	10	9	8
予約		OSHT6	OSHT5	OSHT4	OSHT3	OSHT2	OSHT1
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
予約		CBC6	CBC5	CBC4	CBC3	CBC2	CBC1
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

凡例: R/W= リード/ライト、R= リードのみ、-n= リセット後の値

表 4-17 トリップ・ゾーン・サブモジュール・セレクト・レジスタ (TZSEL) のフィールドの説明

ビット	名前	値	説明
<p>ワンショット (OSHT) トリップ・ゾーンのイネーブル/ディスエーブル。イネーブルになっているピンのいずれかが Low になると、この ePWM モジュールに対してワンショット・トリップ・イベントが発生します。イベントが発生すると、EPWMxA および EPWMxB 出力では、TZCTL レジスタ (表 4-18) で定義されたアクションがとられます。ワンショット・トリップ状態は、ユーザが TZCLR レジスタ (表 4-21) でクリアするまでラッチされたままになります。</p>			
15:14	予約		予約
13	OSHT6		トリップ・ゾーン 6 ( $\overline{TZ6}$ ) セレクト
		0	この ePWM モジュールのワンショット・トリップ・ソースとして $\overline{TZ6}$ をディスエーブルにします。
12	OSHT5	1	この ePWM モジュールのワンショット・トリップ・ソースとして $\overline{TZ6}$ をイネーブルにします。
		0	トリップ・ゾーン 5 ( $\overline{TZ5}$ ) セレクト
11	OSHT4	0	この ePWM モジュールのワンショット・トリップ・ソースとして $\overline{TZ5}$ をディスエーブルにします。
		1	この ePWM モジュールのワンショット・トリップ・ソースとして $\overline{TZ5}$ をイネーブルにします。
10	OSHT3	0	トリップ・ゾーン 4 ( $\overline{TZ4}$ ) セレクト
		1	この ePWM モジュールのワンショット・トリップ・ソースとして $\overline{TZ4}$ をディスエーブルにします。
9	OSHT2	0	この ePWM モジュールのワンショット・トリップ・ソースとして $\overline{TZ4}$ をイネーブルにします。
		1	トリップ・ゾーン 3 ( $\overline{TZ3}$ ) セレクト
8	OSHT1	0	この ePWM モジュールのワンショット・トリップ・ソースとして $\overline{TZ3}$ をディスエーブルにします。
		1	この ePWM モジュールのワンショット・トリップ・ソースとして $\overline{TZ3}$ をイネーブルにします。
<p>サイクル・バイ・サイクル (CBC) トリップ・ゾーンのイネーブル/ディスエーブル。イネーブルになっているピンのいずれかが Low になると、この ePWM モジュールに対してサイクル・バイ・サイクル・トリップ・イベントが発生します。イベントが発生すると、EPWMxA および EPWMxB 出力では、TZCTL レジスタ (表 4-18) で定義されたアクションがとられます。サイクル・バイ・サイクル・トリップ状態は、タイム・ベース・カウンタがゼロに達すると自動的にクリアされます。</p>			
7:6	予約		予約
5	CBC6		トリップ・ゾーン 6 ( $\overline{TZ6}$ ) セレクト
		0	この ePWM モジュールの CBC トリップ・ソースとして $\overline{TZ6}$ をディスエーブルにします。
4	CBC5	1	この ePWM モジュールの CBC トリップ・ソースとして $\overline{TZ6}$ をイネーブルにします。
		0	トリップ・ゾーン 5 ( $\overline{TZ5}$ ) セレクト
3	CBC4	0	この ePWM モジュールの CBC トリップ・ソースとして $\overline{TZ5}$ をディスエーブルにします。
		1	この ePWM モジュールの CBC トリップ・ソースとして $\overline{TZ5}$ をイネーブルにします。
2	CBC3	0	トリップ・ゾーン 4 ( $\overline{TZ4}$ ) セレクト
		1	この ePWM モジュールの CBC トリップ・ソースとして $\overline{TZ4}$ をディスエーブルにします。
1	CBC2	0	この ePWM モジュールの CBC トリップ・ソースとして $\overline{TZ4}$ をイネーブルにします。
		1	トリップ・ゾーン 3 ( $\overline{TZ3}$ ) セレクト
		0	この ePWM モジュールの CBC トリップ・ソースとして $\overline{TZ3}$ をディスエーブルにします。
		1	この ePWM モジュールの CBC トリップ・ソースとして $\overline{TZ3}$ をイネーブルにします。
		0	トリップ・ゾーン 2 ( $\overline{TZ2}$ ) セレクト
		1	この ePWM モジュールの CBC トリップ・ソースとして $\overline{TZ2}$ をディスエーブルにします。
		0	この ePWM モジュールの CBC トリップ・ソースとして $\overline{TZ2}$ をイネーブルにします。
		1	この ePWM モジュールの CBC トリップ・ソースとして $\overline{TZ2}$ をイネーブルにします。

表 4-17 トリップ・ゾーン・サブモジュール・セレクト・レジスタ (TZSEL) のフィールドの説明 (続き)

ビット	名前	値	説明
0	CBC1	0	トリップ・ゾーン1 (TZ1) セレクト この ePWM モジュールの CBC トリップ・ソースとして $\overline{TZ1}$ をディスエーブルにします。
		1	この ePWM モジュールの CBC トリップ・ソースとして $\overline{TZ1}$ をイネーブルにします。

図 4-18 トリップ・ゾーン制御レジスタ (TZCTL)

15	予約			8	
R-0					
7	4	3	2	1	0
予約		TZB		TZA	
R-0		R/W-0		R/W-0	

凡例：R/W= リード/ライト、R= リードのみ、-n= リセット後の値

表 4-18 トリップ・ゾーン制御レジスタ (TZCTL) のフィールドの説明

ビット	名前	値	説明
15-4	予約		予約
3-2	TZB	00	トリップ・イベントが発生すると、EPWMxB 出力で以下のアクションがとられます。どのトリップ・ゾーン・ピンによってイベントが発生するのかは、TZSEL レジスタ (表 4-17) で定義されます。 ハイ・インピーダンス (EPWMxB= ハイ・インピーダンス状態)
		01	EPWMxB を High 状態にします。
		10	EPWMxB を Low 状態にします。
		11	何もしません。EPWMxB では何のアクションもとられません。
1-0	TZA	00	トリップ・イベントが発生すると、EPWMxA 出力で以下のアクションがとられます。どのトリップ・ゾーン・ピンによってイベントが発生するのかは、TZSEL レジスタ (表 4-17) で定義されます。 ハイ・インピーダンス (EPWMxA= ハイ・インピーダンス状態)
		01	EPWMxA を High 状態にします。
		10	EPWMxA を Low 状態にします。
		11	何もしません。EPWMxA では何のアクションもとられません。

図 4-19 トリップ・ゾーン・イネーブル割り込みレジスタ (TZEINT)

15	予約			8
R-0				
7	3	2	1	0
予約		OST	CBC	予約
R-0		R/W-0	R/W-0	R-0

凡例：R/W= リード/ライト、R= リードのみ、-n= リセット後の値

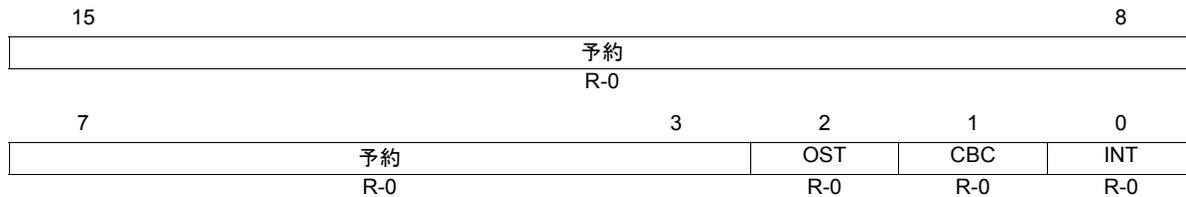
表 4-19 トリップ・ゾーン・イネーブル割り込みレジスタ (TZEINT) のフィールドの説明

ビット	名前	値	説明
15-3	予約		予約
2	OST	0	トリップ・ゾーン・ワンショット割り込みイネーブル ワンショット割り込み生成をディスエーブルにします。
		1	割り込み生成をイネーブルにします。ワンショット・トリップ・イベントによって EPWMx_ TZINT PIE 割り込みが発生します。(1)

ビット	名前	値	説明
1	CBC	0 1	トリップ・ゾーン・サイクル・バイ・サイクル割り込みイネーブル サイクル・バイ・サイクル割り込み生成をディスエーブルにします。 割り込み生成をイネーブルにします。サイクル・バイ・サイクル・トリップ・イベントによって EPWMx_TZINT PIE 割り込みが発生します。(1)
0	予約		予約

(1) ペリフェラル割り込み拡張 (PIE) モジュールの説明は、『TMS320x280x, 2801x, 2804x System Control and Interrupts Reference Guide』(SPRU712) に記載されています。

図 4-20 トリップ・ゾーン・フラグ・レジスタ (TZFLG)



凡例：R/W= リード/ライト、R= リードのみ、-n= リセット後の値

表 4-20 トリップ・ゾーン・フラグ・レジスタ (TZFLG) のフィールドの説明

ビット	名前	値	説明
15-3	予約		予約
2	OST	0 1	ワンショット・トリップ・イベント・ステータス・ラッチ・フラグ ワンショット・トリップ・イベントは発生しません。 ワンショット・トリップ・ソースとして選択されたピンでトリップ・イベントが発生したことを示します。 このビットは、TZCLR レジスタ(表 4-21)に適切な値をライトするとクリアされます。
1	CBC	0 1	サイクル・バイ・サイクル・トリップ・イベント・ステータス・ラッチ・フラグ サイクル・バイ・サイクル・トリップ・イベントは発生しません。 サイクル・バイ・サイクル・トリップ・ソースとして選択されたピンでトリップ・イベントが発生したことを示します。TZFLG[CBC] ビットは、ユーザが手動でクリアするまでセットされたままになります。CBC ビットをクリアしたときにサイクル・バイ・サイクル・トリップ・イベントがまだ存在する場合は、CBC がもう一度すぐにセットされます。トリップ状態が存在しなくなった場合は、ePWM タイム・ベース・カウンタがゼロに達すると (TBCTR=0x0000)、指定したピン状態が自動的にクリアされます。サイクルのどこで CBC フラグがクリアされようとも、ピンの状態は TBCTR=0x0000 の場合のみクリアされます。 このビットは、TZCLR レジスタ(表 4-21)に適切な値をライトするとクリアされます。
0	INT	0 1	トリップ割り込みステータス・ラッチ・フラグ 割り込みが生成されていないことを示します。 トリップ状態が原因で EPWMx_TZINT PIE 割り込みが生成されたことを示します。 このフラグがクリアされるまでは、後続の EPWMx_TZINT PIE 割り込みは生成されません。CBC または OST がセットされているときに割り込みフラグがクリアされた場合は、別の割り込みパルスが生成されます。すべてのフラグ・ビットをクリアすると、後続の割り込みが防止されます。 このビットは、TZCLR レジスタ(表 4-21)に適切な値をライトするとクリアされます。

図 4-21 トリップ・ゾーン・クリア・レジスタ (TZCLR)

15	予約				8
R-0					
7	3	2	1	0	
予約		OST	CBC	INT	
R-0		R/W-0	R/W-0	R/W-0	

凡例：R/W= リード/ライト、R= リードのみ、-n= リセット後の値

表 4-21 トリップ・ゾーン・クリア・レジスタ (TZCLR) のフィールドの説明

ビット	名前	値	説明
15-3	予約		予約
2	OST	0 1	ワンショット・トリップ (OST) ラッチのクリア・フラグ 何の影響も及ぼしません。常に 0 をリード・バックします。 このトリップ (セット) 状態をクリアします。
1	CBC	0 1	サイクル・バイ・サイクル (CBC) トリップ・ラッチのクリア・フラグ 何の影響も及ぼしません。常に 0 をリード・バックします。 このトリップ (セット) 状態をクリアします。
0	INT	0 1	グローバル割り込みラッチのクリア・フラグ 何の影響も及ぼしません。常に 0 をリード・バックします。 この ePWM モジュールに対するトリップ割り込みフラグ (TZFLG[INT]) をクリアします。 <b>注：</b> このフラグがクリアされるまでは、後続の EPWMx_TZINT PIE 割り込みは生成されません。TZFLG[INT] ビットがクリアされ、他のいずれかのフラグ・ビットがセットされると、別の割り込みパルスが生成されます。すべてのフラグ・ビットをクリアすると、後続の割り込みが防止されます。

図 4-22 トリップ・ゾーン強制レジスタ (TZFRC)

15	予約				8
R-0					
7	3	2	1	0	
予約		OST	CBC	予約	
R-0		R/W-0	R/W-0	R-0	

凡例：R/W= リード/ライト、R= リードのみ、-n= リセット後の値

表 4-22 トリップ・ゾーン強制レジスタ (TZFRC) のフィールドの説明

ビット	名前	値	説明
15-3	予約		予約
2	OST	0 1	ソフトウェアによるワンショット・トリップ・イベントの強制 0 のライトは無視されます。常に 0 をリード・バックします。 ワンショット・トリップ・イベントを強制し、TZFLG[OST] ビットをセットします。
1	CBC	0 1	ソフトウェアによるサイクル・バイ・サイクル・トリップ・イベントの強制 0 のライトは無視されます。常に 0 をリード・バックします。 サイクル・バイ・サイクル・トリップ・イベントを強制し、TZFLG[CBC] ビットをセットします。
0	予約		予約

#### 4.7 イベント・トリガ・サブモジュール・レジスタ

図 4-23 ~ 図 4-27 と表 4-23 ~ 表 4-27 では、イベント・トリガ・サブモジュールのレジスタを説明します。

図 4-23 イベント・トリガ選択レジスタ (ETSEL)

15	14	12	11	10	8
SOCBEN	SOCBSEL	SOCAEN	SOCASEL		
R/W-0	R/W-0	R/W-0	R/W-0		
7	4	3	2	0	
予約		INTEN	INTSEL		
R-0		R/W-0	R/W-0		

凡例：R/W= リード/ライト、R= リードのみ、-n= リセット後の値

表 4-23 イベント・トリガ選択レジスタ (ETSEL) のフィールドの説明

ビット	名前	値	説明
15	SOCBEN	0 1	ADC 変換開始 B (EPWMxSOCB) パルスのイネーブル EPWMxSOCB をディスエーブルにします。 EPWMxSOCB パルスをイネーブルにします。
14-12	SOCBSEL	000 001 010 011 100 101 110 111	EPWMxSOCB 選択オプション これらのビットでは、EPWMxSOCB パルスをいつ生成するのが決定されます。 予約 「タイム・ベース・カウンタはゼロに等しい (TBCTR=0x0000)」 イベントをイネーブルにします。 「タイム・ベース・カウンタは周期に等しい (TBCTR=TBPRD)」 イベントをイネーブルにします。 予約 タイマがインクリメントされている場合は、「タイム・ベース・カウンタは CMPA に等しい」 イベントをイネーブルにします。 タイマがデクリメントされている場合は、「タイム・ベース・カウンタは CMPA に等しい」 イベントをイネーブルにします。 タイマがインクリメントされている場合は、「タイム・ベース・カウンタは CMPB に等しい」 イベントをイネーブルにします。 タイマがデクリメントされている場合は、「タイム・ベース・カウンタは CMPB に等しい」 イベントをイネーブルにします。
11	SOCAEN	0 1	ADC 変換開始 A (EPWMxSOCA) パルスのイネーブル EPWMxSOCA をディスエーブルにします。 EPWMxSOCA パルスをイネーブルにします。
10-8	SOCASEL	000 001 010 011 100 101 110 111	EPWMxSOCA 選択オプション これらのビットでは、EPWMxSOCA パルスをいつ生成するのが決定されます。 予約 「タイム・ベース・カウンタはゼロに等しい (TBCTR=0x0000)」 イベントをイネーブルにします。 「タイム・ベース・カウンタは周期に等しい (TBCTR=TBPRD)」 イベントをイネーブルにします。 予約 タイマがインクリメントされている場合は、「タイム・ベース・カウンタは CMPA に等しい」 イベントをイネーブルにします。 タイマがデクリメントされている場合は、「タイム・ベース・カウンタは CMPA に等しい」 イベントをイネーブルにします。 タイマがインクリメントされている場合は、「タイム・ベース・カウンタは CMPB に等しい」 イベントをイネーブルにします。 タイマがデクリメントされている場合は、「タイム・ベース・カウンタは CMPB に等しい」 イベントをイネーブルにします。

表 4-23 イベント・トリガ選択レジスタ (ETSEL) のフィールドの説明 (続き)

ビット	名前	値	説明
7-4	予約		予約
3	INTEN	0 1	ePWM 割り込み (EPWMx_INT) 生成のイネーブル EPWMx_INT 生成をディスエーブルにします。 EPWMx_INT 生成をイネーブルにします。
2-0	INTSEL	000 001 010 011 100 101 110 111	ePWM 割り込み (EPWMx_INT) 選択オプション 予約 「タイム・ベース・カウンタはゼロに等しい (TBCTR=0x0000)」 イベントをイネーブルにします。 「タイム・ベース・カウンタは周期に等しい (TBCTR=TBPRD)」 イベントをイネーブルにします。 予約 タイマがインクリメントされている場合は、「タイム・ベース・カウンタは CMPA に等しい」 イベントをイネーブルにします。 タイマがデクリメントされている場合は、「タイム・ベース・カウンタは CMPA に等しい」 イベントをイネーブルにします。 タイマがインクリメントされている場合は、「タイム・ベース・カウンタは CMPB に等しい」 イベントをイネーブルにします。 タイマがデクリメントされている場合は、「タイム・ベース・カウンタは CMPB に等しい」 イベントをイネーブルにします。

図 4-24 イベント・トリガ・プリスケール・レジスタ (ETPS)

15	14	13	12	11	10	9	8
SOCBCNT		SOCBPRD		SOCACNT		SOCAPRD	
R-0		R/W-0		R-0		R/W-0	
7		4	3	2	1		0
予約				INTCNT		INTPRD	
R-0				R-0		R/W-0	

凡例: R/W= リード/ライト、R= リードのみ、-n= リセット後の値

表 4-24 イベント・トリガ・プリスケール・レジスタ (ETPS) のフィールドの説明

ビット	名前	値	説明
15-14	SOCBCNT	00 01 10 11	ePWM ADC 変換開始 B イベント (EPWMxSOCB) カウンタ・レジスタ これらのビットは、選択した ETSEL[SOCBSEL] イベントがいくつ発生したのかを示します。 イベントは発生していません。 1つのイベントが発生しました。 2つのイベントが発生しました。 3つのイベントが発生しました。
13-12	SOCBPRD	00 01 10 11	ePWM ADC 変換開始 B イベント (EPWMxSOCB) 周期セレクト これらのビットでは、選択した ETSEL[SOCBSEL] イベントが EPWMxSOCB パルスの生成前にいくつ発生する必要があるのが決定されます。パルスを生成するには、そのパルスをイネーブルにする必要があります (ETSEL[SOCBEN]=1)。ステータス・フラグが前回の変換開始からセットされていても (ETFLG[SOCB]=1)、SOCB パルスは生成されず、SOCB パルスが生成されると、ETPS[SOCBCNT] ビットが自動的にクリアされます。 00 SOCB イベント・カウンタをディスエーブルにします。EPWMxSOCB パルスは生成されません。 01 最初のイベント (ETPS[SOCBCNT]=0,1) で EPWMxSOCB パルスを生成します。 10 2番目のイベント (ETPS[SOCBCNT]=1,0) で EPWMxSOCB パルスを生成します。 11 3番目のイベント (ETPS[SOCBCNT]=1,1) で EPWMxSOCB パルスを生成します。

表 4-24 イベント・トリガ・プリスケール・レジスタ (ETPS) のフィールドの説明 (続き)

ビット	名前	値	説明
11-10	SOCACNT	00 01 10 11	ePWM ADC 変換開始 A イベント (EPWMxSOCA) カウンタ・レジスタ これらのビットは、選択した ETSEL[SOCASEL] イベントがいくつ発生したのかを示します。 00 イベントは発生していません。 01 1つのイベントが発生しました。 10 2つのイベントが発生しました。 11 3つのイベントが発生しました。
9-8	SOCAPRD	00 01 10 11	ePWM ADC 変換開始 A イベント (EPWMxSOCA) 周期セレクト これらのビットでは、選択した ETSEL[SOCASEL] イベントが EPWMxSOCA パルスの生成前にいくつ発生する必要があるのかが決定されます。パルスを生成するには、そのパルスをイネーブルにする必要があります (ETSEL[SOCASEN]=1)。ステータス・フラグが前回の変換開始からセットされていても (ETFLG[SOCAL]=1)、SOCA パルスは生成されます SOCA パルスが生成されると、ETPS[SOCACNT] ビットが自動的にクリアされます。 00 SOCA イベント・カウンタをディスエーブルにします。EPWMxSOCA パルスは生成されません。 01 最初のイベント (ETPS[SOCACNT]=0,1) で EPWMxSOCA パルスを生成します。 10 2番目のイベント (ETPS[SOCACNT]=1,0) で EPWMxSOCA パルスを生成します。 11 3番目のイベント (ETPS[SOCACNT]=1,1) で EPWMxSOCA パルスを生成します。
7-4	予約		予約
3-2	INTCNT	00 01 10 11	ePWM 割り込みイベント (EPWMx_INT) カウンタ・レジスタ これらのビットは、選択した ETSEL[INTSEL] イベントがいくつ発生したのかを示します。これらのビットは、割り込みパルスが生成されると自動的にクリアされます。割り込みをディスエーブル (ETSEL[INT]=0) にするか、または割り込みフラグをセット (ETFLG[INT]=1) した場合は、カウンタが周期値に達すると (ETPS[INTCNT]=ETPS[INTPRD])、イベントのカウンタが停止されます。 00 イベントは発生していません。 01 1つのイベントが発生しました。 10 2つのイベントが発生しました。 11 3つのイベントが発生しました。
1-0	INTPRD	00 01 10 11	ePWM 割り込み (EPWMx_INT) カウンタ・レジスタ これらのビットでは、選択した ETSEL[INTSEL] イベントが割り込みの生成前にいくつ発生する必要があるのかが決定されます。割り込みを生成するには、その割り込みをイネーブルにする必要があります (ETSEL[INT]=1)。割り込みステータス・フラグが前回の割り込みからセットされている場合は (ETFLG[INT]=1)、ETCLR[INT] ビットによってフラグがクリアされるまで割り込みは生成されません。割り込みが生成されると、ETPS[INTCNT] ビットが自動的にクリアされます。 割り込みがイネーブルになっていて、ステータス・フラグがクリアされている場合は、現在のカウンタ値と同じ INTPRD 値をライトすると、割り込みがトリガされます。 現在のカウンタ値よりも小さい INTPRD 値をライトすると、定義されていない状態になります。 新しいゼロまたは非ゼロの INTPRD 値のライトと同時にカウンタ・イベントが発生した場合は、カウンタがインクリメントされます。 00 割り込みイベント・カウンタをディスエーブルにします。割り込みは生成されず、ETFRC[INT] は無視されます。 01 INTCNT=0,1 (最初のイベント) で割り込みを生成します。 10 ETPS[INTCNT]=1,0 (2番目のイベント) で割り込みを生成します。 11 ETPS[INTCNT]=1,1 (3番目のイベント) で割り込みを生成します。

図 4-25 イベント・トリガ・フラグ・レジスタ (ETFLG)

15	予約 R-0					8
7	4	3	2	1	0	
	予約 R-0	SOCB R-0	SOCA R-0	予約 R-0	INT R-0	

凡例：R/W= リード/ライト、R= リードのみ、-n= リセット後の値

表 4-25 イベント・トリガ・フラグ・レジスタ (ETFLG) のフィールドの説明

ビット	名前	値	説明
15-4	予約		予約
3	SOCB	0 1	ラッチされた ePWM ADC 変換開始 B (EPWMxSOCB) ステータス・フラグ EPWMxSOCB イベントが発生していないことを示します。 1 EPWMxSOCB で変換開始パルスが生成されたことを示します。EPWMxSOCB 出力は、フラグ・ビットがセットされていても引き続き生成されます。
2	SOCA	0 1	ePWM ADC 変換開始 A (EPWMxSOCA) ステータス・ラッチ・フラグ ETFLG[INT] フラグとは異なり、EPWMxSOCA 出力はフラグ・ビットがセットされていても引き続きパルスを生成します。 0 イベントが発生していないことを示します。 1 EPWMxSOCA で変換開始パルスが生成されたことを示します。EPWMxSOCA 出力は、フラグ・ビットがセットされていても引き続き生成されます。
1	予約		予約
0	INT	0 1	ePWM 割り込み (EPWMx_INT) ステータス・ラッチ・フラグ 0 イベントが発生していないことを示します。 1 ePWMx 割り込み (EPWMx_INT) が生成されたことを示します。このフラグ・ビットがクリアされるまでは、後続の割り込みは生成されません。

図 4-26 イベント・トリガ・クリア・レジスタ (ETCLR)

15	予約 R-0					8
7	4	3	2	1	0	
	予約 R-0	SOCB R/W-0	SOCA R/W-0	予約 R-0	INT R/W-0	

凡例：R/W= リード/ライト、R= リードのみ、-n= リセット後の値

表 4-26 イベント・トリガ・クリア・レジスタ (ETCLR) のフィールドの説明

ビット	名前	値	説明
15-4	予約		予約
3	SOCB	0 1	ePWM ADC 変換開始 B (EPWMxSOCB) フラグ・クリア・ビット 0 をライトしても影響はありません。常に 0 をリード・バックします。 1 ETFLG[SOCB] フラグ・ビットをクリアします。
2	SOCA	0 1	ePWM ADC 変換開始 A (EPWMxSOCA) フラグ・クリア・ビット 0 をライトしても影響はありません。常に 0 をリード・バックします。 1 ETFLG[SOCA] フラグ・ビットをクリアします。
1	予約		予約
0	INT	0 1	ePWM 割り込み (EPWMx_INT) フラグ・クリア・ビット 0 をライトしても影響はありません。常に 0 をリード・バックします。 1 ETFLG[INT] フラグ・ビットをクリアし、後続の割り込みパルスの生成をイネーブルにします。

#### 4.8 正しい割り込み初期化手順

ePWM ペリフェラル・クロックをイネーブルにすると、ePWM レジスタが正しく初期化されていないために誤ったイベントによって割り込みフラグがセットされる可能性があります。

ePWM ペリフェラルを初期化する正しい手順は以下のとおりです。

1. グローバル割り込みをディスエーブルにする (CPU INTM フラグ)
2. ePWM 割り込みをディスエーブルにする
3. ペリフェラル・レジスタを初期化する
4. 誤った ePWM フラグ (PIEIFR も含む) をすべてクリアする
5. ePWM 割り込みをイネーブルにする
6. グローバル割り込みをイネーブルにする

図 4-27 イベント・トリガ強制レジスタ (ETFRC)

	予約 R-0					
	予約 R-0	SOCB R-0	SOCA R-0	予約 R-0	INT R-0	

凡例：R/W= リード/ライト、R= リードのみ、-n= リセット後の値

表 4-27 イベント・トリガ強制レジスタ (ETFRC) のフィールドの説明

ビット	名前	値	説明
15-4	予約		予約
3	SOCB	0 1	SOCB 強制ビット。SOCB パルスは、ETSEL レジスタでイベントがイネーブルになっている場合にのみ生成されます。それとは関係なく、SOCBFLG フラグ・ビットはセットされます。 0 何の影響も及ぼしません。常に 0 をリード・バックします。 1 EPWMxSOCB でパルスを生成し、SOCBFLG ビットをセットします。このビットは、テストの目的で使用されます。
2	SOCA	0 1	SOCA 強制ビット。SOCA パルスは、ETSEL レジスタでイベントがイネーブルになっている場合にのみ生成されます。それとは関係なく、SOCAFLG フラグ・ビットはセットされます。 0 このビットに対する 0 のライトは無視されます。常に 0 をリード・バックします。 1 EPWMxSOCA でパルスを生成し、SOCAFLG ビットをセットします。このビットは、テストの目的で使用されます。
1	予約		予約
0	INT	0 1	INT 強制ビット。割り込みは、ETSEL レジスタでイベントがイネーブルになっている場合にのみ生成されます。それとは関係なく、INT フラグ・ビットはセットされます。 0 このビットに対する 0 のライトは無視されます。常に 0 をリード・バックします。 1 EPWMxINT で割り込みを生成し、INT フラグ・ビットをセットします。このビットは、テストの目的で使用されます。

# ご注意

日本テキサス・インスツルメンツ株式会社(以下TIJといひます)及びTexas Instruments Incorporated(TIJの親会社、以下TIJないしTexas Instruments Incorporatedを総称してTIといひます)は、その製品及びサービスを任意に修正し、改善、改良、その他の変更をし、もしくは製品の製造中止またはサービスの提供を中止する権利を留保します。従ひまして、お客様は、発注される前に、関連する最新の情報を取得して頂き、その情報が現在有効かつ完全なものであるかどうかご確認下さい。全ての製品は、お客様とTIJとの間に取引契約が締結されている場合は、当該契約条件に基づき、また当該取引契約が締結されていない場合は、ご注文の受諾の際に提示されるTIJの標準販売契約約款に従って販売されます。

TIJは、そのハードウェア製品が、TIの標準保証条件に従ひ販売時の仕様に対応した性能を有していること、またはお客様とTIJとの間で合意された保証条件に従ひ合意された仕様に対応した性能を有していることを保証します。検査およびその他の品質管理技法は、TIが当該保証を支援するのに必要とみなす範囲で行なわれております。各デバイスの全てのパラメーターに関する固有の検査は、政府がそれ等の実行を義務づけている場合を除き、必ずしも行なわれておりません。

TIJは、製品のアプリケーションに関する支援もしくはお客様の製品の設計について責任を負うことはありません。TI製部品を使用しているお客様の製品及びそのアプリケーションについての責任はお客様にあります。TI製部品を使用したお客様の製品及びアプリケーションについて想定される危険を最小のものとするため、適切な設計上および操作上の安全対策は、必ずお客様にてお取り下さい。

TIJは、TIの製品もしくはサービスが使用されている組み合わせ、機械装置、もしくは方法に関連しているTIの特許権、著作権、回路配置利用権、その他のTIの知的財産権に基づいて何らかのライセンスを許諾するということは明示的にも黙示的にも保証も表明もしていません。TIが第三者の製品もしくはサービスについて情報を提供することは、TIが当該製品もしくはサービスを使用することについてライセンスを与えるとか、保証もしくは承認をすることを意味しません。そのような情報を使用するには第三者の特許その他の知的財産権に基づき当該第三者からライセンスを得なければならない場合もあり、またTIの特許その他の知的財産権に基づきTIからライセンスを得て頂かなければならない場合もあります。

TIのデータ・ブックもしくはデータ・シートの中にある情報を複製することは、その情報に一切の変更を加えること無く、かつその情報と結び付けられた全ての保証、条件、制限及び通知と共に複製がなされる限りにおいて許されるものとします。当該情報に変更を加えて複製することは不正で誤認を生じさせる行為です。TIは、そのような変更された情報や複製については何の義務も責任も負いません。

TIの製品もしくはサービスについてTIJにより示された数値、特性、条件その他のパラメーターと異なる、あるいは、それを超えてなされた説明で当該TI製品もしくはサービスを再販売することは、当該TI製品もしくはサービスに対する全ての明示的保証、及び何らかの黙示的保証を無効にし、かつ不正で誤認を生じさせる行為です。TIJは、そのような説明については何の義務も責任もありません。

TIJは、TIの製品が、安全でないことが致命的となる用途ないしアプリケーション(例えば、生命維持装置のように、TI製品に不良があった場合に、その不良により相当な確率で死傷等の重篤な事故が発生するようなもの)に使用されることを認めておりません。但し、お客様とTIの双方の権限有る役員が書面でそのような使用について明確に合意した場合は除きます。たとえTIJがアプリケーションに関連した情報やサポートを提供したとしても、お客様は、そのようなアプリケーションの安全面及び規制面から見た諸問題を解決するために必要とされる専門的知識及び技術を持ち、かつ、お客様の製品について、またTI製品をそのような安全でないことが致命的となる用途に使用することについて、お客様が全ての法的責任、規制を遵守する責任、及び安全に関する要求事項を満足させる責任を負っていることを認め、かつそのことに同意します。さらに、もし万一、TIの製品がそのような安全でないことが致命的となる用途に使用されたことによって損害が発生し、TIないしその代表者がその損害を賠償した場合は、お客様がTIないしその代表者にその全額の補償をするものとします。

TI製品は、軍事的用途もしくは宇宙航空アプリケーションないし軍事的環境、航空宇宙環境にて使用されるようには設計もされていませんし、使用されることを意図されていません。但し、当該TI製品が、軍需対応グレード品、若しくは「強化プラスチック」製品としてTIが特別に指定した製品である場合は除きます。TIが軍需対応グレード品として指定した製品のみが軍需品の仕様書に合致いたします。お客様は、TIが軍需対応グレード品として指定していない製品を、軍事的用途もしくは軍事的環境下で使用することは、もっぱらお客様の危険負担においてなされるということ、及び、お客様がもっぱら責任をもって、そのような使用に関して必要とされる全ての法的要求事項及び規制上の要求事項を満足させなければならないことを認め、かつ同意します。

TI製品は、自動車用アプリケーションないし自動車の環境において使用されるようには設計されていませんし、また使用されることを意図されていません。但し、TIがISO/TS 16949の要求事項を満たしていると特別に指定したTI製品は除きます。お客様は、お客様が当該TI指定品以外のTI製品を自動車用アプリケーションに使用しても、TIは当該要求事項を満たしていなかったことについて、いかなる責任も負わないことを認め、かつ同意します。

Copyright © 2009, Texas Instruments Incorporated  
日本語版 日本テキサス・インスツルメンツ株式会社

## 弊社半導体製品の取り扱い・保管について

半導体製品は、取り扱い、保管・輸送環境、基板実装条件によっては、お客様での実装前後に破壊/劣化、または故障を起こすことがあります。

弊社半導体製品のお取り扱い、ご使用にあたっては下記の点を遵守して下さい。

### 1. 静電気

素手で半導体製品単体を触らないこと。どうしても触る必要がある場合は、リストストラップ等で人体からアースをとり、導電性手袋等をして取り扱うこと。

弊社出荷梱包単位(外装から取り出された内装及び個装)又は製品単品で取り扱いを行う場合は、接地された導電性のテーブル上で(導電性マットにアースをとったもの等)、アースをした作業者が行うこと。また、コンテナ等も、導電性のものを使用すること。

マウンタやはんだ付け設備等、半導体の実装に関わる全ての装置類は、静電気の帯電を防止する措置を施すこと。

前記のリストストラップ・導電性手袋・テーブル表面及び実装装置類の接地等の静電気帯電防止措置は、常に管理されその機能が確認されていること。

### 2. 温・湿度環境

温度: 0 ~ 40 °C、相対湿度: 40 ~ 85%で保管・輸送及び取り扱いを行うこと。(但し、結露しないこと。)

直射日光があたる状態で保管・輸送しないこと。

### 3. 防湿梱包

防湿梱包品は、開封後は個別推奨保管環境及び期間に従ひ基板実装すること。

### 4. 機械的衝撃

梱包品(外装、内装、個装)及び製品単品を落下させたり、衝撃を与えないこと。

### 5. 熱衝撃

はんだ付け時は、最低限260 °C以上の高温状態に、10秒以上さらさないこと。(個別推奨条件がある時はそれに従うこと。)

### 6. 汚染

はんだ付け性を損なう、又はアルミ配線腐食の原因となるような汚染物質(硫黄、塩素等ハロゲン)のある環境で保管・輸送しないこと。はんだ付け後は十分にフラックスの洗浄を行うこと。(不純物含有率が一定以下に保証された無洗浄タイプのフラックスは除く。)

以上