Application Note

EPWM to MCPWM Migration Guide

Luke Jones

ABSTRACT

In the Application-Specific Microcontrollers (ASM) device family, there are different types of PWM modules: EPWM (Enhanced Pulse-Width Modulator) and MCPWM (Multi-Channel Pulse-Width Modulator) module. Most C2000 devices such as F2837xD, F28003x and F28P55x have utilized the EPWM module. The F28E12x devices are the first to utilize the MCPWM module.

The MCPWM module was designed as a low-cost version of the EPWM module specifically for motor control applications. More specifically, a single MCPWM module is intended to drive a 3-phase motor. For this reason, a single MCPWM instance can have up to six outputs unlike the EPWM module which only has two. Still, the core functionality of the MCPWM module remains the same.

Due to the reduced feature-set of MCPWM, and the combination of three PWM pairs into a single MCPWM instance, the programming procedure for the two modules have significant differences which are addressed in this document. The programming procedures and feature differences for the time-base, counter-compare, action-qualifier, and other submodules are detailed. The recommended workarounds to successfully migrate an EPWM-based application to MCPWM are described in this application note.

Table of Contents

List of Figures

List of Tables

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

Before migrating an EPWM application to MCPWM, there are high-level feature differences between the two modules to determine whether migration between the two modules is feasible for an existing application. Once this evaluation is complete, users can begin migrating a EPWM code to MCPWM. This document highlights differences for both register and driverlib-based programming. Differences between EPWM and MCPWM are divided between the following submodules:

- Time-based submodule
- Counter-compare submodule
- Action qualifier submodule
- Deadband submodule
- Trip-zone submodule
- Event trigger submodule
- PWM X-BAR

2 Key Differences Between EPWM and MCPWM

The key feature differences between EPWM and MCPWM are listed below.

- Each MCPWM instance has up to six outputs instead of two (1A, 1B,2A, 2B, 3A, 3B)
- Single TBCTR for one MCPWM instance (all PWM pairs within an MCPWM instance share a single TBCTR)
- Separate shadow registers for TBPRD, CMPA, and so forth are visible to users. Users must write to the shadow registers directly to use shadow loading
- Removed features: HRPWM, TZ interrupt, digital compare submodule, chopper submodule
- Dead-band, trip-zone, time base counter settings shared across three PWM pairs, reducing design flexibility

To visualize the differences between EPWM and MCPWM, the block diagrams for both modules are shown below.

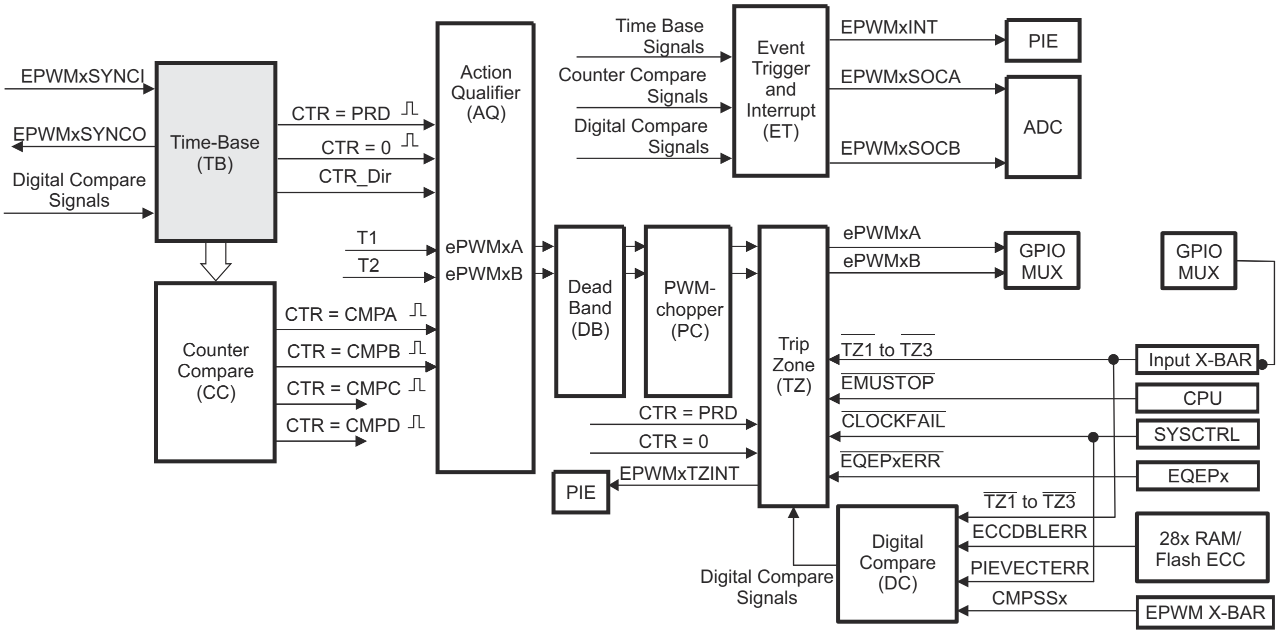


Figure 2-1. EPWM Block Diagram

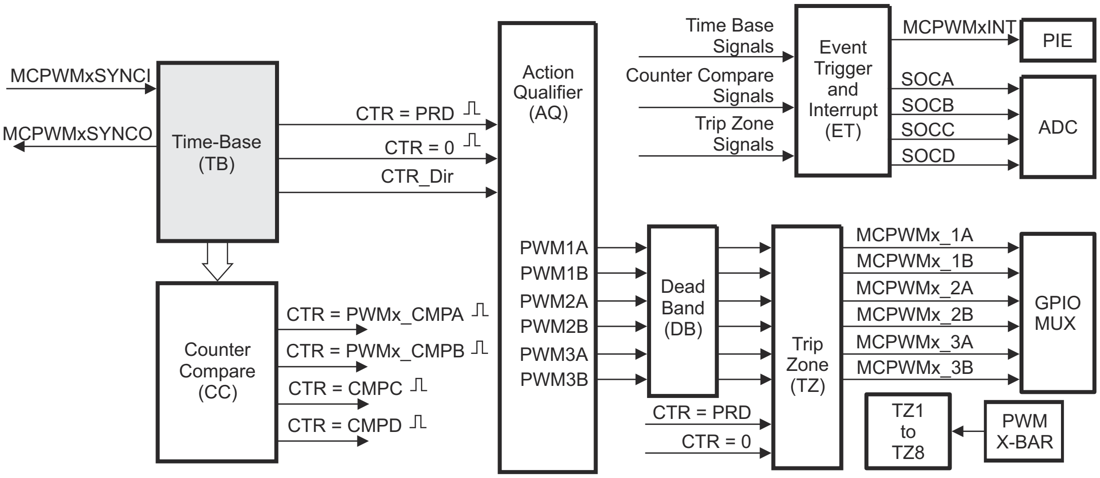


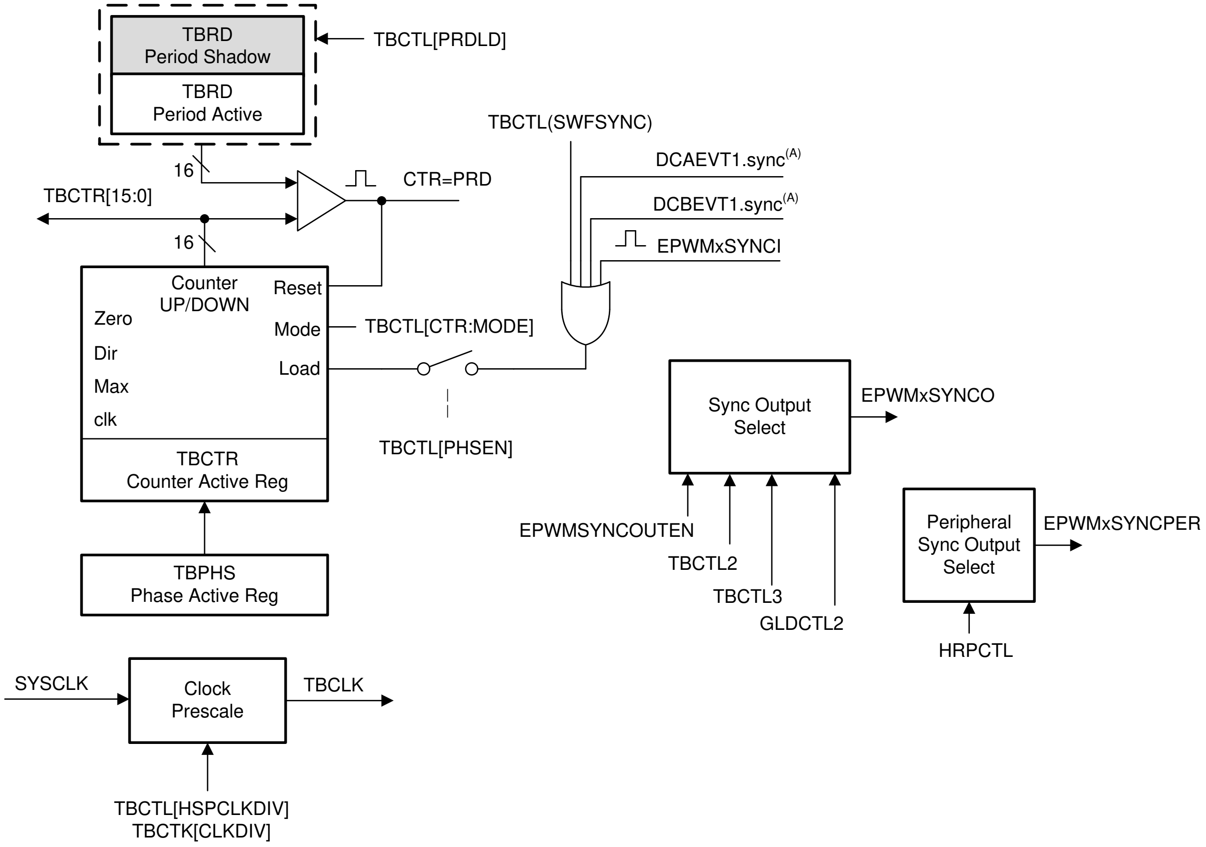
Figure 2-2. MCPWM Block Diagram

3 Time-Base Submodule Differences

The Time-Base submodule is responsible for the PWM frequency and synchronization between EPWM or MCPWM modules. This section details differences between the Time-Base Submodule in EPWM and MCPWM.

The feature differences of the MCPWM Time-Base submodule when compared to EPWM are as follows:

- Addition of memory mapped Time-Base Period shadow register (TBPRDS). When shadow loading is enabled, this register must be written to directly instead of the active register (TBPRD).
- Clock divider and high speed clock divider combined into single divider (TBCTL.CLKDIV) on MCPWM
- One-shot sync feature removed on MCPWM
- Down-Count mode removed on MCPWM
- CTRMAX flag removed
- Up to 3 PWM pairs share the same counter (phase-shifting between PWM pairs must be implemented via Counter-Compare Submodule)



A. These signals are generated by the digital compare (DC) submodule.

Figure 3-1. EPWM Time-Base Submodule Block Diagram

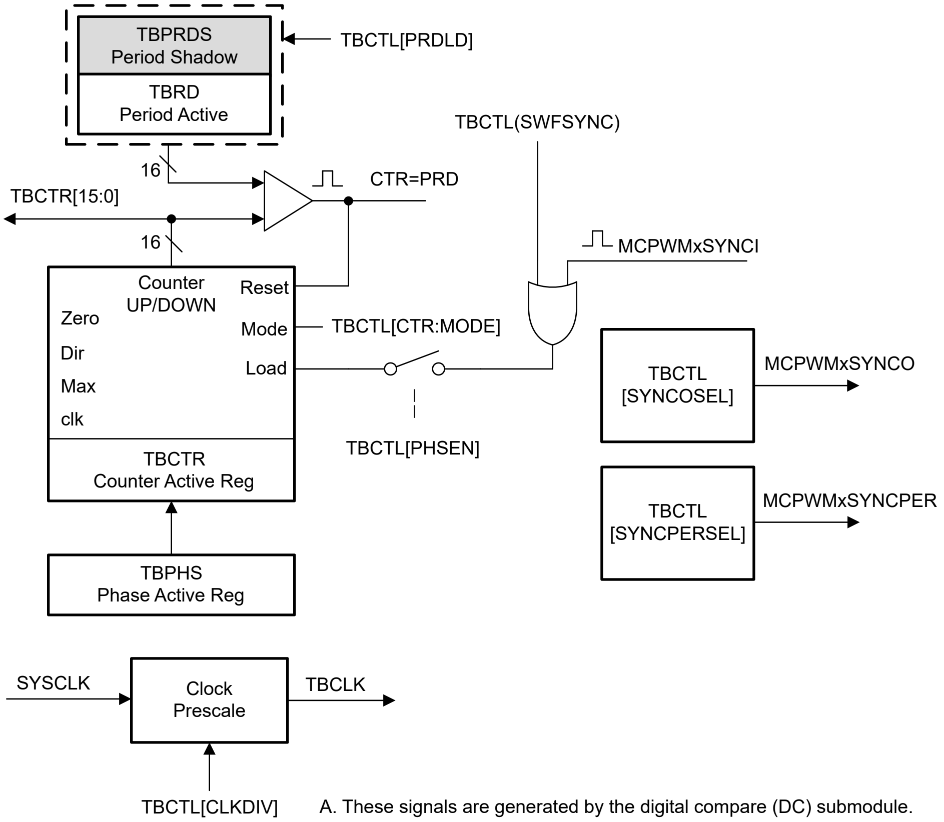


Figure 3-2. MCPWM Time-Base Submodule Block Diagram

3.1 Register Differences

The register differences for the Time-Base Submodule between EPWM and MCPWM are listed in Table 3-1.

Table 3-1. Time-Base Submodule Register Differences Between EPWM and MCPWM

EPWM	MCPWM	Description
TBCTL2.OSHTSYNC	**-**	**One-shot Sync Out Feature Removed on MCPWM**
TBCTL2.OSHTSYNCMODE	-	One-shot sync out feature removed on MCPWM
TBCTL2.PRDLDSYNC	-	Shadow loading of period on SYNC input pulse removed on MCPWM
TBCTL3.OSSFRCEN	-	One-shot sync out feature removed on MCPWM
TBPHS.TBPHSHR	-	HRPWM not present on MCPWM
TBSTS.CTRMAX	-	CTRMAX flag removed on MCPWM
TBCTL.HSPCLKDIV	TBCTL.CLKDIV	High Speed TBCLK Pre-scaler
TBCTL.SWFSYNC	TBCTL.SWSYNC	Bitfield name change
SYNCINSEL.SEL	TBCTL.SYNCISEL	Dedicated SYNC input select register combined into TBCTL register, refer to TBCTL register description.
SYNCOUTEN.CMPBEN	TBCTL.SYNCOSEL	Dedicated SYNC input select register combined into TBCTL register, refer to TBCTL register description.
SYNCOUTEN.CMPCEN	TBCTL.SYNCOSEL	Dedicated SYNC input select register combined into TBCTL register, refer to TBCTL register description.
SYNCOUTEN.CMPDEN	TBCTL.SYNCOSEL	Dedicated SYNC input select register combined into TBCTL register, refer to TBCTL register description.

Table 3-1. Time-Base Submodule Register Differences Between EPWM and MCPWM (continued)

EPWM	MCPWM	Description
TBCTL2.OSHTSYNC	**-**	**One-shot Sync Out Feature Removed on MCPWM**
SYNCOUTEN.DCAEVT1EN	TBCTL.SYNCOSEL	Dedicated SYNC input select register combined into TBCTL register, refer to TBCTL register description.
SYNCOUTEN.DCBEVT1EN	TBCTL.SYNCOSEL	Dedicated SYNC input select register combined into TBCTL register, refer to TBCTL register description.
SYNCOUTEN.SWEN	TBCTL.SYNCOSEL	Dedicated SYNC input select register combined into TBCTL register, refer to TBCTL register description.
SYNCOUTEN.ZEROEN	TBCTL.SYNCOSEL	Dedicated SYNC input select register combined into TBCTL register, refer to TBCTL register description.
HRPCTL.PWMSYNCSEL	TBCTL.SYNCPERSEL	SYNCPER control moved from HRPCTL register to TBCTL register
-	TBPRDS.TBPRDS	Addition of memory-mapped TBPRD shadow register
-	TBSTSCLR.SYNCI	Addition of separate register for clearing SYNC input status bit

3.2 Driverlib Differences

The driverlib differences between EPWM and MCPWM are listed in Table 3-2.

Table 3-2. Differences between EPWM and MCPWM

EPWM Function	MCPWM Function	Explanation
EPWM_clearTimeBaseCounterOverflowEvent	-	CTRMAX flag removed on MCPWM
EPWM_disableOneShotSync	-	One-Shot Sync Out feature removed on MCPWM
EPWM_enableOneShotSync	-	One-Shot Sync Out feature removed on MCPWM
EPWM_getTimeBaseCounterOverflowStatus	-	CTRMAX flag removed on MCPWM
EPWM_selectPeriodLoadEvent	-	TBPRD only loaded on CTR= 0 event
EPWM_setOneShotSyncOutTrigger	-	One-Shot Sync Out feature removed on MCPWM
EPWM_startOneShotSync	-	One-Shot Sync Out feature removed on MCPWM
EPWM_clearSyncEvent	MCPWM_clearSyncStatus	Function name change
EPWM_setClockPrescaler	MCPWM_setClockPrescaler	Number of arguments changed due to combination of clock-dividers
EPWM_disableSyncOutPulseSource	MCPWM_setSyncOutPulseMode	OR gate of enabled/disabled SYNCO signals replaced with mux
EPWM_enableSyncOutPulseSource	MCPWM_setSyncOutPulseMode	OR gate of enabled/disabled SYNCO signals replaced with mux
EPWM_getTimeBasePeriod	MCPWM_getTimeBasePeriodActive	Function name change to indicate TBPRD has a corresponding shadow register TBPRDS that is written to unless freeze loading is selected
EPWM_setTimeBasePeriod	MCPWM_setTimeBasePeriodActive	Function name change to indicate TBPRD has a corresponding shadow register TBPRDS that is read from unless freeze loading is selected

Table 3-2. Differences between EPWM and MCPWM (continued)

EPWM Function	MCPWM Function	Explanation
-	MCPWM_getTimeBasePeriodShadow	Addition of memory mapped TBPRD register
-	MCPWM_setSyncPulseSource	Addition of driverlib function to set SYNCPER source
-	MCPWM_setTimeBasePeriodShadow	Addition of memory mapped TBPRD register

4 Counter-Compare Submodule Differences

The Counter-Compare Submodule determines the timing of EPWM/MCPWM events such as rising or falling edges of the PWM output, ADC triggers, or CPU interrupts. The feature differences of MCPWM when compared to EPWM are as follows:

- Addition of memory mapped shadow registers PWMx_CMPAS, PWMx_CPMBS, CMPCS, CMPDS
- Separate CMPA and CMPB registers for each PWM pair: PWM1_CMPA, PWM1_CMPB, PWM2_CMPA, etc.
- CMPC and CMPD are shared across all 3 PWM pairs
- No shadow loading of CMPx registers on sync events

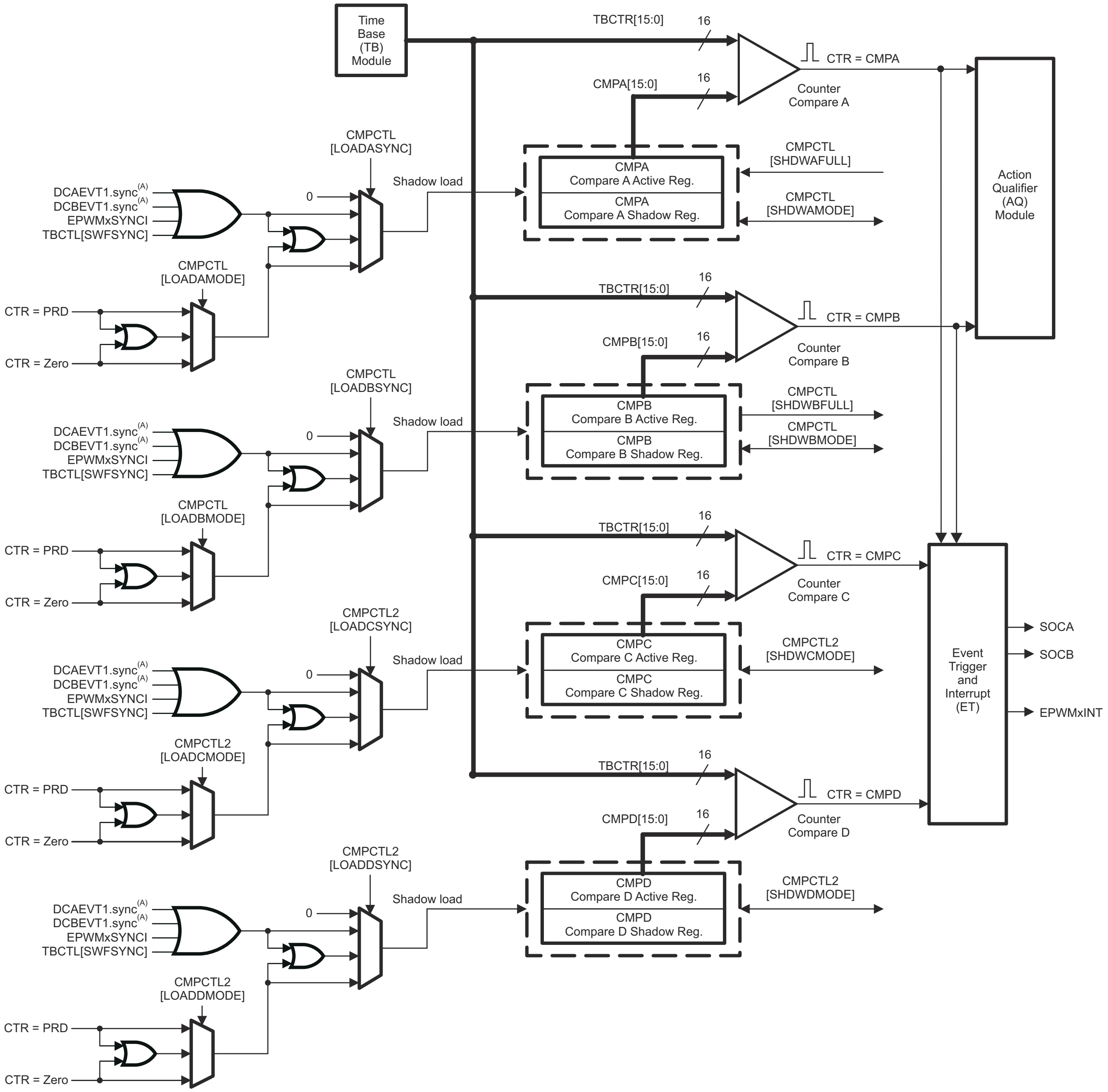


Figure 4-1. Counter-Compare Submodule Block Diagram for EPWM

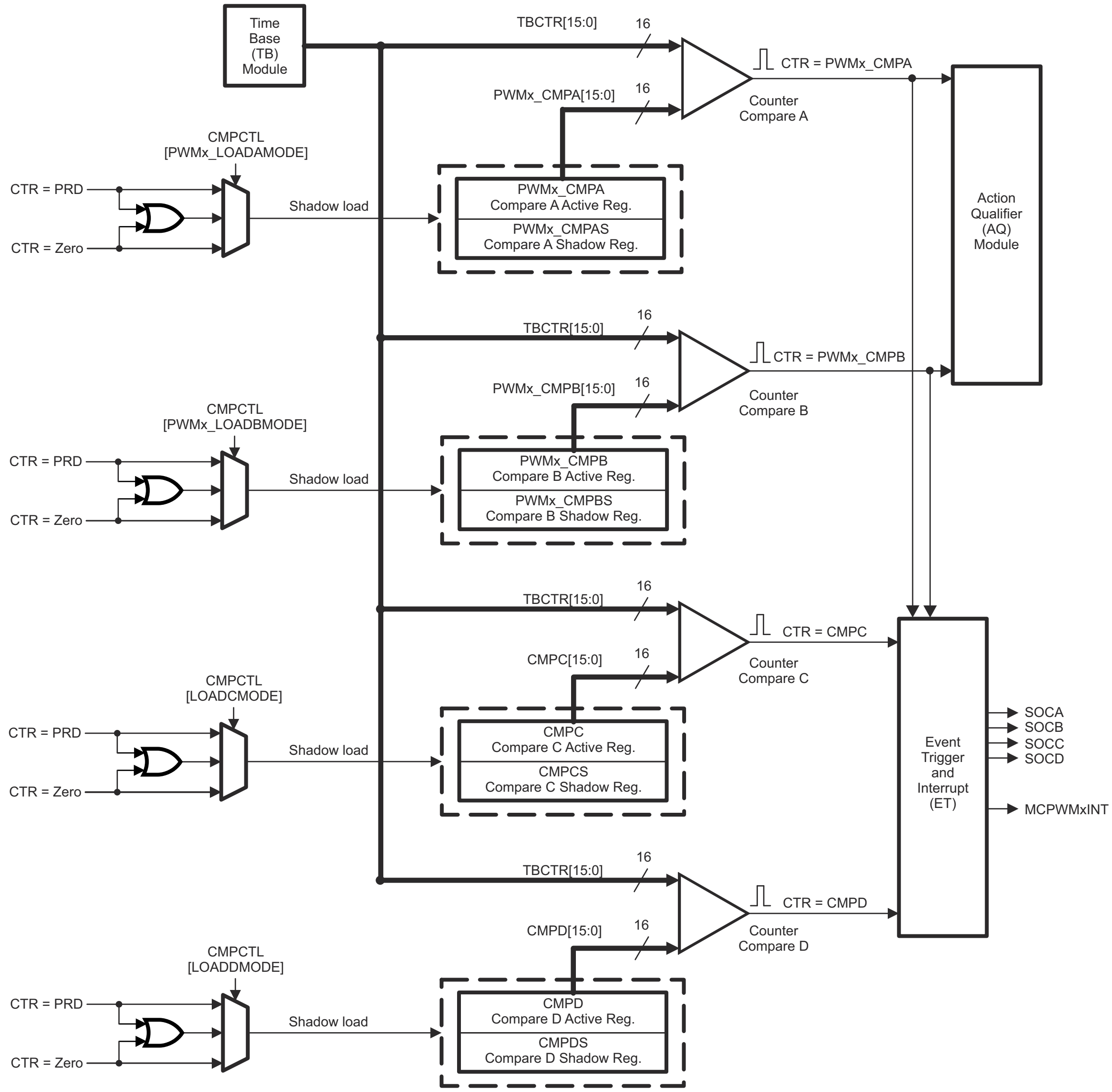


Figure 4-2. Counter-Compare Submodule Block Diagram for MCPWM

4.1 Register Differences

The register differences between EPWM and MCPWM for the Counter-Compare Submodule are listed in .

Table 4-1. Counter-Compare Submodule Register Differences Between EPWM and MCPWM

EPWM	MCPWM	Description
-	CMPCS.CMPCS	Addition of memory-mapped shadow register for CMPC
-	CMPDS.CMPDS	Addition of memory-mapped shadow register for CMPD
CMPA.CMPA	PWMx_CMPA.PWMx_CMPA	Separate CMPA registers for each PWM pair
CMPA.CMPA	PWMx_CMPAS.PWMx_CMPAS	Addition of memory-mapped shadow registers for CMPA
CMPB.CMPB	PWMx_CMPB.PWMx_CMPB	Separate CMPB registers for each PWM pair
CMPB.CMPB	PWMx_CMPBS.PWMx_CMPBS	Addition of memory-mapped shadow registers for CMPB
CMPCTL.LOADAMODE	CMPCTL.PWMx_LOADAMODE	Separate shadow load bits for CMPA for each PWM pair
CMPCTL.LOADBMODE	CMPCTL.PWMx_LOADBMODE	Separate shadow load bits for CMPB for each PWM pair
CMPCTL2.LOADCMODE	CMPCTL.LOADCMODE	Shadow load configuration of CMPD moved from CMPCTL2 to CMPCTL

Table 4-1. Counter-Compare Submodule Register Differences Between EPWM and MCPWM (continued)

EPWM	MCPWM	Description
CMPCTL2.LOADDMODE	CMPCTL.LOADDMODE	Shadow load configuration of CMPC moved from CMPCTL2 to CMPCTL
CMPCTL.LOADBSYNC	-	Loading CMPx registers on SYNCIN pulse removed from MCPWM
CMPCTL.SHDWAFULL	-	Replaced by memory-mapped shadow registers for CMPA
CMPCTL.SHDWAMODE	-	Shadow or active mode not selectable on MCPWM; only shadow load event needs to be configured and correct active or shadow register written to. **Note** Writing to the active register on MCPWM takes no effect unless *freeze* loading is selected in CMPCTL.LOADxMODE
CMPCTL.SHDWBFULL	-	Replaced by memory-mapped shadow registers for CMPA
CMPCTL.SHDWBMODE	-	Shadow or active mode not selectable on MCPWM; only shadow load event needs to be configured and correct active or shadow register be written to. **Note** Writing to the active register on MCPWM takes no effect unless *freeze* loading is selected in CMPCTL.LOADxMODE
CMPCTL2.LOADCSYNC	-	Loading CMPx registers on SYNCIN pulse removed from MCPWM
CMPA.CMPAHR	-	HRPWM removed from MCPWM
CMPCTL.LOADASYNC	-	Loading CMPx registers on SYNCIN pulse removed from MCPWM
CMPB.CMPBHR	-	HRPWM removed from MCPWM
CMPCTL2.LOADDSYNC	-	Loading CMPx registers on SYNCIN pulse removed from MCPWM
CMPCTL2.SHDWCMODE	-	Shadow or active mode not selectable on MCPWM; only shadow load event needs to be configured and correct active or shadow register written to. **Note** Writing to the active register on MCPWM takes no effect unless *freeze* loading is selected in CMPCTL.LOADxMODE
CMPCTL2.SHDWDMODE	-	Shadow or active mode not selectable on MCPWM; only shadow load event needs to be configured and correct active or shadow register written to. **Note** Writing to the active register on MCPWM takes no effect unless *freeze* loading is selected in CMPCTL.LOADxMODE

4.2 Driverlib Differences

The driverlib differences between EPWM and MCPWM for the Counter-Compare Submodule are listed in Table 4-2.

Table 4-2. Counter-Compare Submodule Driverlib Differences Between EPWM and MCPWM

EPWM Function	MCPWM Function	Explanation
-	MCPWM_getCounterCompareShadowValue	Addition of memory mapped CMPx shadow registers
-	MCPWM_setCounterCompareShadowValue	Addition of memory mapped CMPx shadow registers
EPWM_getCounterCompareValue	MCPWM_getCounterCompareActiveValue	Function name change to indicate difference between active register and shadow register
EPWM_setCounterCompareValue	MCPWM_setCounterCompareActiveValue	Function name change to indicate difference between active register and shadow register
EPWM_disableCounterCompareShadowLoadMode	-	Shadow mode is not enabled or disabled on MCPWM. Only shadow load event is selected and correct active or shadow register is written to.
EPWM_getCounterCompareShadowStatus	-	Shadow register full flag replaced by memory mapped shadow register that can be read or written to by the user.

5 Action-Qualifier Submodule Differences

The Action-Qualifier Submodule determines PWM output behavior on specified events such as CMPx events, CTR=0 or CTR=TBPRD events. The feature difference of MCPWM when compared to EPWM are as follows:

- Addition of memory mapped shadow registers AQCTLAS, AQCTLBS
- Separate AQCTLA and AQCTLB registers for each PWM pair: PWM1_AQCTLA, PWM1_AQCTLB, PWM2_AQCTLA, etc.
- T1/T2 events removed from MCPWM

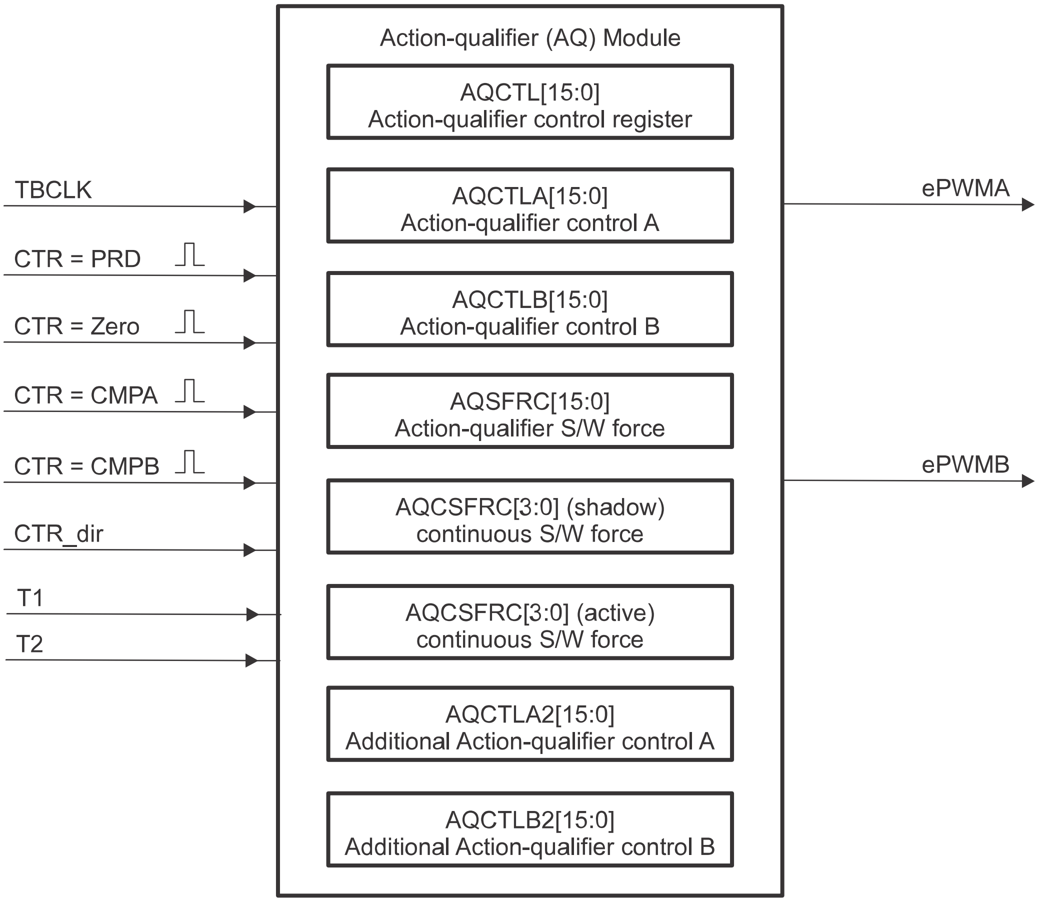


Figure 5-1. Action-Qualifier Submodule Block Diagram for EPWM

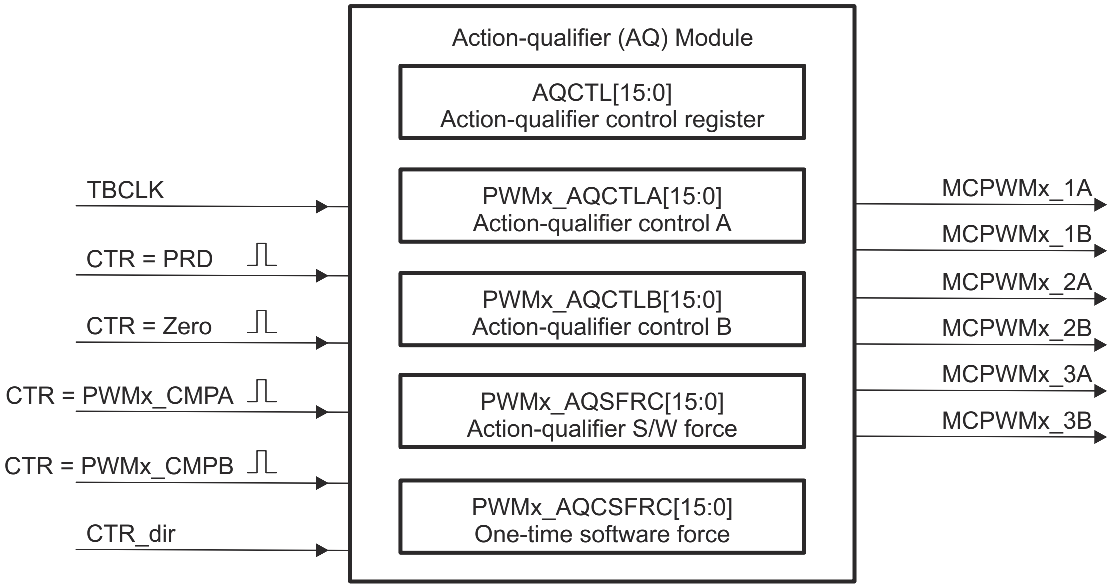


Figure 5-2. Action-Qualifier Submodule Block Diagram for MCPWM

5.1 Register Differences

The Action-Qualifier Submodule register differences between EPWM and MCPWM are listed in Table 5-1.

Table 5-1. Action-Qualifier Submodule register differences between EPWM and MCPWM

EPWM	MCPWM	Description
-	PWMx_AQCTLAS.CAD	Addition of memory mapped shadow register AQCTLAS
-	PWMx_AQCTLAS.CAU	Addition of memory mapped shadow register AQCTLAS
-	PWMx_AQCTLAS.CBD	Addition of memory mapped shadow register AQCTLAS
-	PWMx_AQCTLAS.CBU	Addition of memory mapped shadow register AQCTLAS
-	PWMx_AQCTLAS.PRD	Addition of memory mapped shadow register AQCTLAS
-	PWMx_AQCTLAS.ZRO	Addition of memory mapped shadow register AQCTLAS
-	PWMx_AQCTLBS.CAD	Addition of memory mapped shadow register AQCTLBS
-	PWMx_AQCTLBS.CAU	Addition of memory mapped shadow register AQCTLBS
-	PWMx_AQCTLBS.CBD	Addition of memory mapped shadow register AQCTLBS
-	PWMx_AQCTLBS.CBU	Addition of memory mapped shadow register AQCTLBS
-	PWMx_AQCTLBS.PRD	Addition of memory mapped shadow register AQCTLBS
-	PWMx_AQCTLBS.ZRO	Addition of memory mapped shadow register AQCTLBS

Table 5-1. Action-Qualifier Submodule register differences between EPWM and MCPWM (continued)

EPWM	MCPWM	Description
AQCSFRC.CSFA	PWMx_AQSFRC.PWMA	Register name change for continuous software forced action qualifier behavior. Register replicated for each PWM pair.
AQCSFRC.CSFB	PWMx_AQSFRC.PWMB	Register name change for continuous software forced action qualifier behavior. Register replicated for each PWM pair.
AQCTL.LDAQAMODE	AQCTL.PWMx_LDAQAMODE	Register repeated for each PWM pair
AQCTL.LDAQASYNC	-	Loading of action qualifier registers on SYNC input removed from MCPWM
AQCTL.LDAQBMODE	AQCTL.PWMx_LDAQBMODE	Register repeated for each PWM pair
AQCTL.LDAQBSYNC	-	Loading of action qualifier registers on SYNC input removed from MCPWM
AQCTL.SHDWAQAMODE	-	Shadow loading is not enabled or disabled on MCPWM. Only shadow load events are selected and correct active or shadow register is written to
AQCTL.SHDWAQBMODE	-	Shadow loading is not enabled or disabled on MCPWM. Only shadow load events are selected and correct active or shadow register is written to
AQCTLA.CAD	PWMx_AQCTLA.CAD	Register repeated for each PWM pair
AQCTLA.CAU	PWMx_AQCTLA.CAU	Register repeated for each PWM pair
AQCTLA.CBD	PWMx_AQCTLA.CBD	Register repeated for each PWM pair
AQCTLA.CBU	PWMx_AQCTLA.CBU	Register repeated for each PWM pair
AQCTLA.PRD	PWMx_AQCTLA.PRD	Register repeated for each PWM pair
AQCTLA.ZRO	PWMx_AQCTLA.ZRO	Register repeated for each PWM pair
AQCTLA2.T1D	-	T1 or T2 events removed from MCPWM
AQCTLA2.T1U	-	T1 or T2 events removed from MCPWM
AQCTLA2.T2D	-	T1 or T2 events removed from MCPWM
AQCTLA2.T2U	-	T1 or T2 events removed from MCPWM
AQCTLB.CAD	PWMx_AQCTLB.CAD	Register repeated for each PWM pair
AQCTLB.CAU	PWMx_AQCTLB.CAU	Register repeated for each PWM pair
AQCTLB.CBD	PWMx_AQCTLB.CBD	Register repeated for each PWM pair
AQCTLB.CBU	PWMx_AQCTLB.CBU	Register repeated for each PWM pair
AQCTLB.PRD	PWMx_AQCTLB.PRD	Register repeated for each PWM pair
AQCTLB.ZRO	PWMx_AQCTLB.ZRO	Register repeated for each PWM pair
AQCTLB2.T1D	-	T1 or T2 events removed from MCPWM
AQCTLB2.T1U	-	T1 or T2 events removed from MCPWM
AQCTLB2.T2D	-	T1 or T2 events removed from MCPWM
AQCTLB2.T2U	-	T1 or T2 events removed from MCPWM
AQSFRC.ACTSFA	PWMx_AQOTSFRC.PWMA	Register name change for one-time software forced action qualifier events. Register repeated for each PWM pair.

Table 5-1. Action-Qualifier Submodule register differences between EPWM and MCPWM (continued)

EPWM	MCPWM	Description
AQSFRC.ACTSFB	PWMx_AQOTSFRC.PWMB	Register name change for one-time software forced action qualifier events. Register repeated for each PWM pair.
AQSFRC.OTSFA	-	Separate bitfield for triggering action qualifier events removed. Configuring one-time action qualifier software force behavior triggers action automatically
AQSFRC.OTSFB	-	Separate bitfield for triggering action qualifier events removed. Configuring one-time action qualifier software force behavior triggers action automatically
AQSFRC.RLDCSF	-	Shadow loading of action qualifier software force events is removed
AQTSRCSEL.T1SEL	-	T1 or T2 events removed from MCPWM
AQTSRCSEL.T2SEL	-	T1 or T2 events removed from MCPWM

5.2 Driverlib Differences

The Action-Qualifier Submodule driverlib differences between EPWM and MCPWM are listed Table 5-2.

Table 5-2. Action-Qualifier Submodule Driverlib Differences Between EPWM and MCPWM

EPWM Function	MCPWM Function	Explanation
-	MCPWM_setActionQualifierActionCompleteShadow	Additional function for shadow register
-	MCPWM_setActionQualifierActionShadow	Additional function for shadow register
EPWM_setActionQualifierAction	MCPWM_setActionQualifierActionActive	Function renamed to indicate this writes to the *active* register and not the shadow register
EPWM_setActionQualifierActionComplete	MCPWM_setActionQualifierActionCompleteActive	Function renamed to indicate this writes to the *active* register and not the shadow register
EPWM_setActionQualifierContSWForceAction	MCPWM_setActionQualifierSWAction	MCPWM_setActionQualifierSWAction shared between continuous and one-time AQ software forced events
EPWM_disableActionQualifierShadowLoadMode	-	Shadow loading not enabled or disabled on MCPWM. Only shadow load event is selected and correct shadow or active register is written to by the user
EPWM_setActionQualifierContSWForceShadowMode	-	Shadow loading not enabled or disabled on MCPWM. Only shadow load event is selected and correct shadow or active register is written to by the user
EPWM_setActionQualifierT1TriggerSource	-	T1 or T2 removed from MCPWM
EPWM_setActionQualifierT2TriggerSource	-	T1 or T2 removed from MCPWM
EPWM_setAdditionalActionQualifierActionComplete	-	T1 or T2 removed from MCPWM

6 Dead-Band Submodule Differences

The Dead-Band Submodule creates dead time between edges of complementary PWM outputs, and inverts and swaps PWM outputs as needed. The feature differences of the MCPWM Dead-Band submodule when compared to EPWM are as follows:

- Dead-band settings shared across all three PWM pairs within an MCPWM module. This is due to the MCPWM's intended use to drive a 3-phase motor by six PWM channels where the PWM configurations for each channel pair are nearly identical.
- Addition of memory-mapped shadow registers for DBRED, DBFED
- Half-cycle clocking mode removed from MCPWM since HRPWM is not present

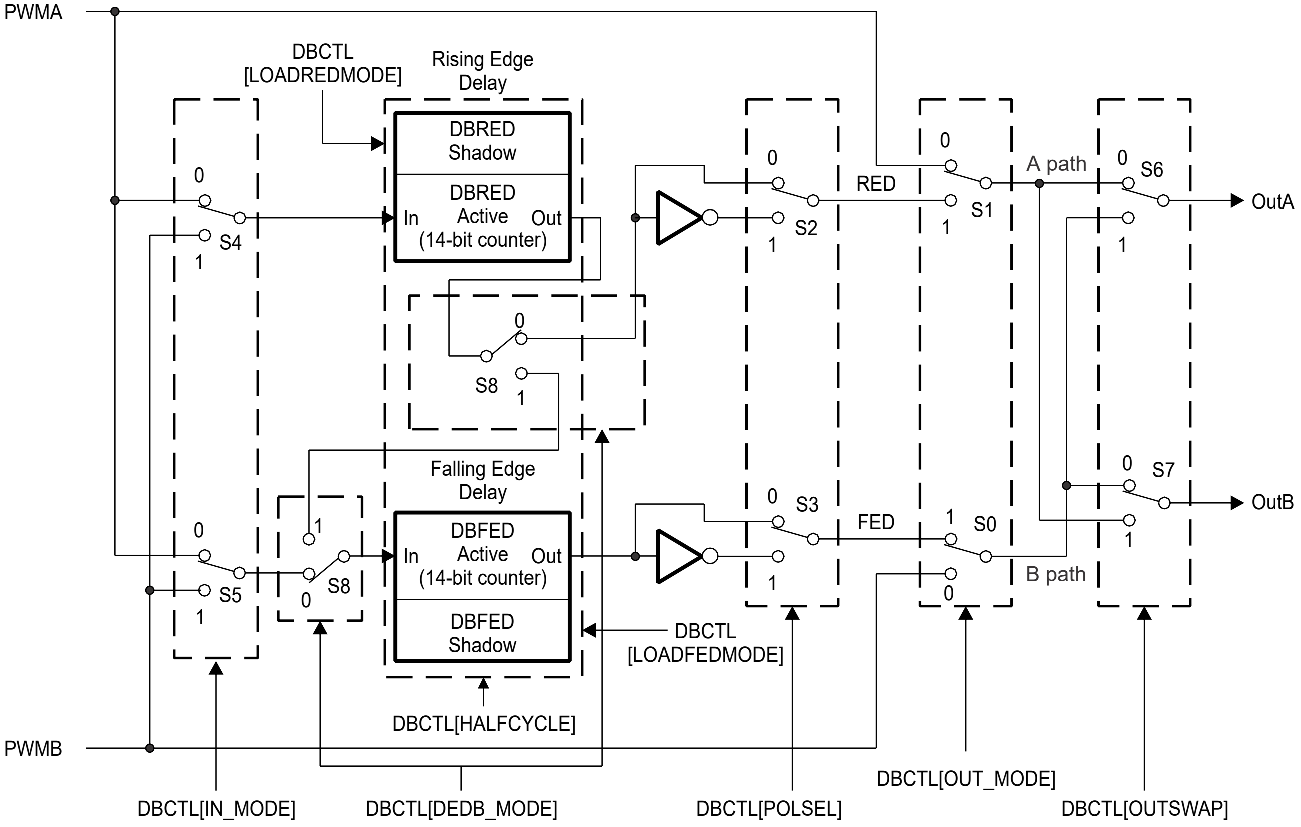


Figure 6-1. Dead-Band Submodule Block Diagram for EPWM

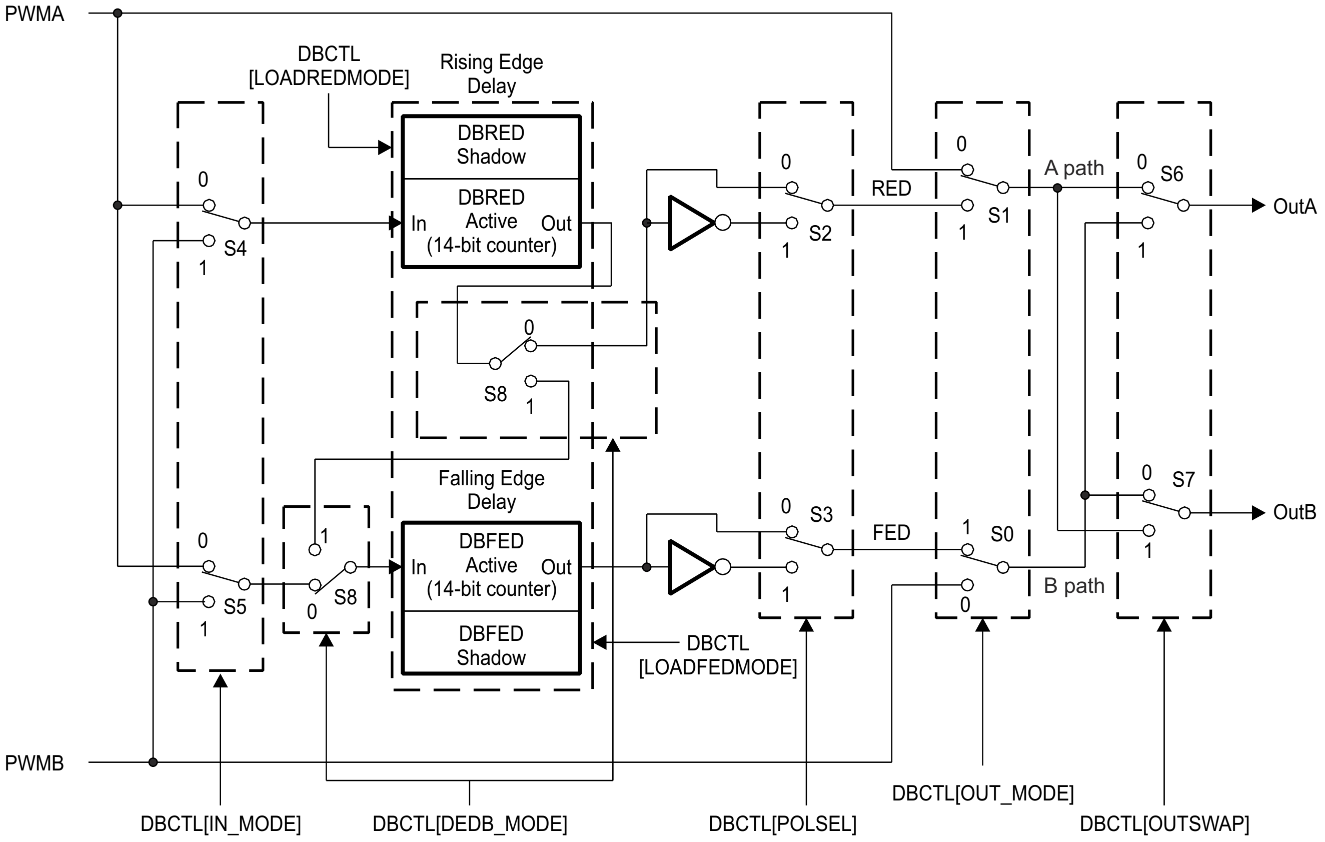


Figure 6-2. Dead-Band Submodule Block Diagram for MCPWM

6.1 Register Differences

The Dead-Band Submodule Register differences between EPWM and MCPWM are listed in Table 6-1.

Table 6-1. Dead-Band Submodule Register Difference Between EPWM and MCPWM

EPWM	MCPWM	Description
-	DBFEDS.DBFEDS	Addition of memory-mapped shadow register for DBFED
-	DBREDS.DBREDS	Addition of memory-mapped shadow register for DBRED
DBCTL.HALFCYCLE	-	Half-cycle clocking mode removed from MCPWM
DBCTL.SHDWDBFEDMODE	-	Shadow loading not enabled or disabled on MCPWM. Only shadow load event is selected and correct active or shadow register written to.
DBCTL.SHDWDBREDMODE	-	Shadow loading not enabled or disabled on MCPWM. Only shadow load event is selected and correct active or shadow register written to.
DBCTL2.LOADDBCTLMODE	-	Shadow loading not enabled or disabled on MCPWM. Only shadow load event is selected and correct active or shadow register written to.
DBCTL2.SHDWDBCTLMODE	-	Shadow loading not enabled or disabled on MCPWM. Only shadow load event is selected and correct active or shadow register written to.
DBFEDHR.DBFEDHR	-	HRPWM removed from MCPWM
DBREDHR.DBREDHR	-	HRPWM removed from MCPWM

6.2 Driverlib Differences

The Dead-Band Submodule driverlib differences between MCPWM and EPWM are listed in Table 6-2.

Table 6-2. Dead-Band Submodule Driverlib Differences Between MCPWM and EPWM

EPWM Function	MCPWM Function	Explanation
-	MCPWM_setFallingEdgeDelayCountShadow	Addition of memory-mapped shadow register for DBFED
-	MCPWM_setRisingEdgeDelayCountShadow	Addition of memory-mapped shadow register for DBRED
EPWM_setFallingEdgeDelayCount	MCPWM_setFallingEdgeDelayCountActive	Function name change to indicate difference between active register and shadow register
EPWM_setRisingEdgeDelayCount	MCPWM_setRisingEdgeDelayCountActive	Function name change to indicate difference between active register and shadow register
EPWM_disableDeadBandControlShadowLoadMode	-	Shadow loading not enabled or disabled on MCPWM. Only shadow load event is selected and correct shadow / active register is written to by the user.
EPWM_disableFallingEdgeDelayCountShadowLoadMode	-	Shadow loading not enabled or disabled on MCPWM. Only shadow load event is selected and correct shadow / active register is written to by the user.
EPWM_disableRisingEdgeDelayCountShadowLoadMode	-	Shadow loading not enabled or disabled on MCPWM. Only shadow load event is selected and correct shadow / active register is written to by the user.
EPWM_setDeadBandControlShadowLoadMode	-	No shadow register for DBCTL
EPWM_setDeadBandCounterClock	-	Half-cycle clocking mode removed from MCPWM

7 Trip-Zone Submodule Differences

The Trip-zone Submodule is used to override the PWM output to be a safe state based on external or software-based events. The feature differences of the MCPWM Trip-Zone submodule when compared to EPWM are as follows:

- Digital Compare Submodule removed
- Notion of TRIPn and TZn no longer applies; replaced with just TZ1-TZ8
- Each TZn signal is directly driven by a PWM X-BAR output instead of special internal signals like INPUTXBAR[1-3]
- Trip zone settings are shared across all 3 PWM pairs, similar to dead-band submodule
- Software forced tripping (TZFRC) is removed from MCPWM; use action-qualifier software force events or software-controlled GPIOs as a substitute
- Separate TZ interrupt removed; now shared with a single MCPWM interrupt

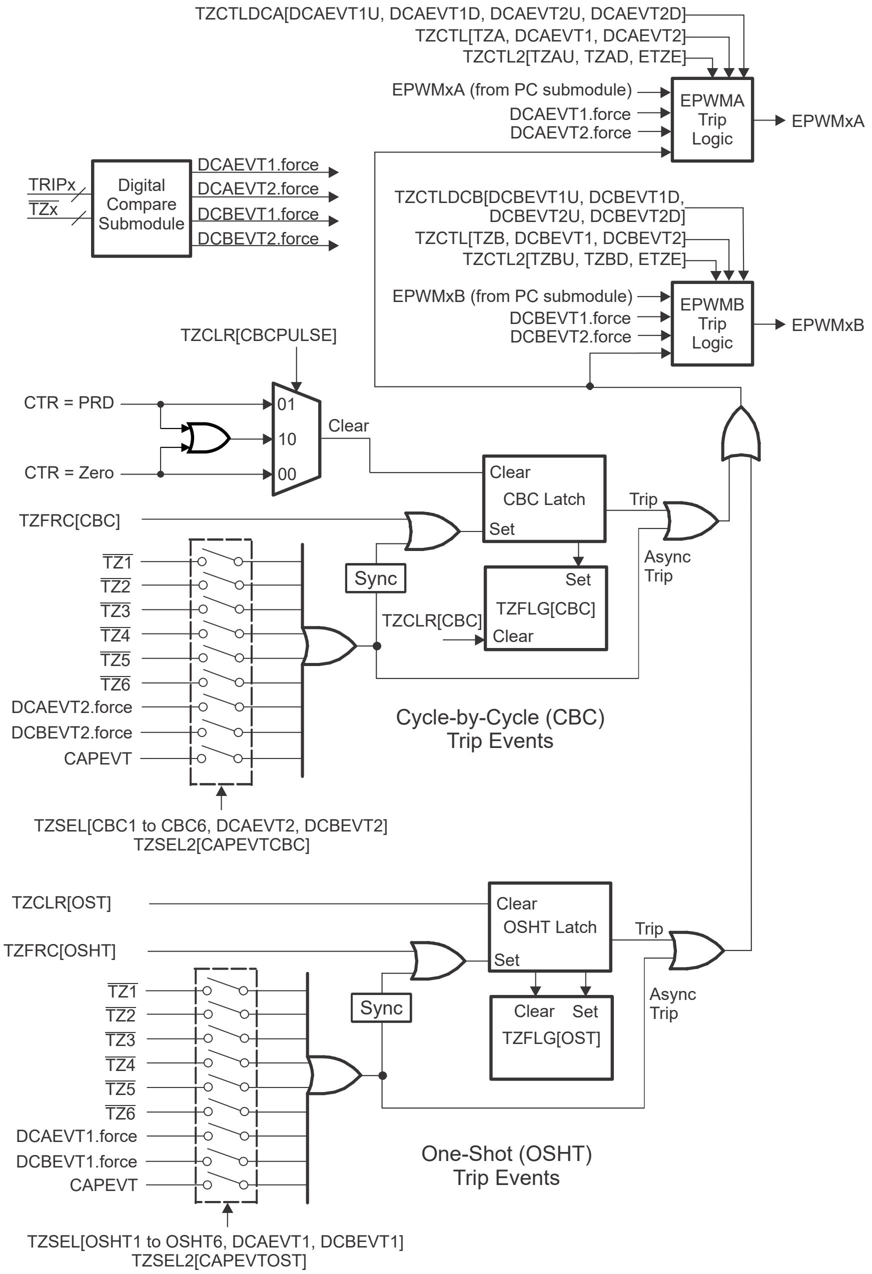


Figure 7-1. Trip-Zone Submodule Block Diagram for EPWM

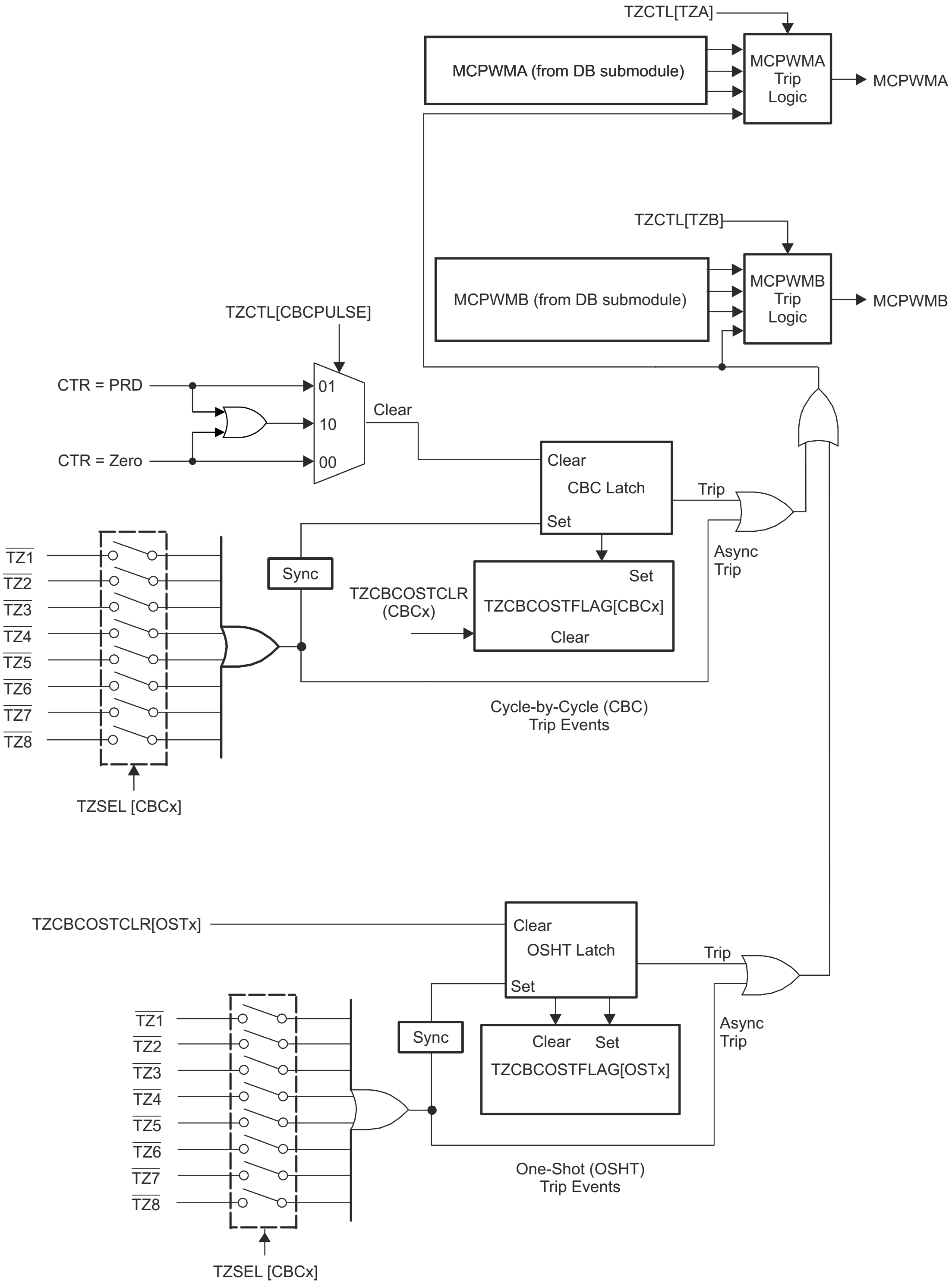


Figure 7-2. Trip-Zone Submodule Block Diagram for MCPWM

7.1 Register Differences

The Trip-Zone Submodule Register differences between EPWM and MCPWM are listed in Table 7-1.

Table 7-1. Trip-Zone Submodule Register Differences Between EPWM and MCPWM

EPWM	MCPWM	Description
LOCK.TZCFGLOCK	-	Locking feature removed from MCPWM
LOCK.TZCLRLOCK	-	Locking feature removed from MCPWM
TZCBCCLR.CBC1	TZCBCOSTCLR.CBC1	CBC and OST Trip Flag clear registers combined
TZCBCCLR.CBC2	TZCBCOSTCLR.CBC2	CBC and OST Trip Flag clear registers combined
TZCBCCLR.CBC3	TZCBCOSTCLR.CBC3	CBC and OST Trip Flag clear registers combined
TZCBCCLR.CBC4	TZCBCOSTCLR.CBC4	CBC and OST Trip Flag clear registers combined
TZCBCCLR.CBC5	TZCBCOSTCLR.CBC5	CBC and OST Trip Flag clear registers combined
TZCBCCLR.CBC6	TZCBCOSTCLR.CBC6	CBC and OST Trip Flag clear registers combined
-	TZCBCOSTCLR.CBC7	Additional TZ7 Trip Signal
-	TZCBCOSTCLR.CBC8	Additional TZ8 Trip Signal
TZCBCCLR.DCAEVT2	-	Digital Compare Submodule removed
TZCBCCLR.DCBEVT2	-	Digital Compare Submodule removed
TZCBCFLG.CBC1	TZCBCOSTFLAG.CBC1	CBC and OST Trip Flag clear registers combined
TZCBCFLG.CBC2	TZCBCOSTFLAG.CBC2	CBC and OST Trip Flag clear registers combined
TZCBCFLG.CBC3	TZCBCOSTFLAG.CBC3	CBC and OST Trip Flag clear registers combined
TZCBCFLG.CBC4	TZCBCOSTFLAG.CBC4	CBC and OST Trip Flag clear registers combined
TZCBCFLG.CBC5	TZCBCOSTFLAG.CBC5	CBC and OST Trip Flag clear registers combined
TZCBCFLG.CBC6	TZCBCOSTFLAG.CBC6	CBC and OST Trip Flag clear registers combined
TZCBCFLG.CBC7	TZCBCOSTFLAG.CBC7	Additional TZ7 Trip Signal
TZCBCFLG.CBC8	TZCBCOSTFLAG.CBC8	Additional TZ8 Trip Signal
TZCBCFLG.DCAEVT2	-	Digital Compare Submodule removed
TZCBCFLG.DCBEVT2	-	Digital Compare Submodule removed
TZCLR.CBC	-	TZCLR register removed
TZCLR.CBCPULSE	TZCTL.CBCPULSE	Clear Pulse for CBC Trip Latch
TZCLR.DCAEVT1	-	Digital Compare Submodule removed
TZCLR.DCAEVT2	-	Digital Compare Submodule removed
TZCLR.DCBEVT1	-	Digital Compare Submodule removed
TZCLR.DCBEVT2	-	Digital Compare Submodule removed

Table 7-1. Trip-Zone Submodule Register Differences Between EPWM and MCPWM (continued)

EPWM	MCPWM	Description
TZCLR.INT	-	TZCLR register removed
TZCLR.OST	-	TZCLR register removed
TZCTL.DCAEVT1	-	Digital Compare Submodule removed
TZCTL.DCAEVT2	-	Digital Compare Submodule removed
TZCTL.DCBEVT1	-	Digital Compare Submodule removed
TZCTL.DCBEVT2	-	Digital Compare Submodule removed
TZCTL2.ETZE	-	TZCTL2 register removed (count-direction specific trip events not available)
TZCTL2.TZAD	-	TZCTL2 register removed (count-direction specific trip events not available)
TZCTL2.TZAU	-	TZCTL2 register removed (count-direction specific trip events not available)
TZCTL2.TZBD	-	TZCTL2 register removed (count-direction specific trip events not available)
TZCTL2.TZBU	-	TZCTL2 register removed (count-direction specific trip events not available)
TZCTLDCA.DCAEVT1D	-	Digital Compare Submodule removed
TZCTLDCA.DCAEVT1U	-	Digital Compare Submodule removed
TZCTLDCA.DCAEVT2D	-	Digital Compare Submodule removed
TZCTLDCA.DCAEVT2U	-	Digital Compare Submodule removed
TZCTLDCB.DCBEVT1D	-	Digital Compare Submodule removed
TZCTLDCB.DCBEVT1U	-	Digital Compare Submodule removed
TZCTLDCB.DCBEVT2D	-	Digital Compare Submodule removed
TZCTLDCB.DCBEVT2U	-	Digital Compare Submodule removed
TZDCSEL.DCAEVT1	-	Digital Compare Submodule removed
TZDCSEL.DCAEVT2	-	Digital Compare Submodule removed
TZDCSEL.DCBEVT1	-	Digital Compare Submodule removed
TZDCSEL.DCBEVT2	-	Digital Compare Submodule removed
TZEINT.CBC	-	Trip Interrupt removed from MCPWM
TZEINT.DCAEVT1	-	Trip Interrupt removed from MCPWM
TZEINT.DCAEVT2	-	Trip Interrupt removed from MCPWM
TZEINT.DCBEVT1	-	Trip Interrupt removed from MCPWM
TZEINT.DCBEVT2	-	Trip Interrupt removed from MCPWM
TZEINT.OST	-	Trip Interrupt removed from MCPWM
TZFLG.CBC	-	TZFLG register removed, refer to TZCBCOSTFLAG
TZFLG.DCAEVT1	-	TZFLG register removed, refer to TZCBCOSTFLAG
TZFLG.DCAEVT2	-	TZFLG register removed, refer to TZCBCOSTFLAG
TZFLG.DCBEVT1	-	TZFLG register removed, refer to TZCBCOSTFLAG

Table 7-1. Trip-Zone Submodule Register Differences Between EPWM and MCPWM (continued)

EPWM	MCPWM	Description
TZFLG.DCBEVT2	-	TZFLG register removed, refer to TZCBCOSTFLAG
TZFLG.INT	-	TZFLG register removed, refer to TZCBCOSTFLAG
TZFLG.OST	-	TZFLG register removed, refer to TZCBCOSTFLAG
TZFRC.CBC	-	Software-forced trip events removed from MCPWM
TZFRC.DCAEVT1	-	Software-forced trip events removed from MCPWM
TZFRC.DCAEVT2	-	Software-forced trip events removed from MCPWM
TZFRC.DCBEVT1	-	Software-forced trip events removed from MCPWM
TZFRC.DCBEVT2	-	Software-forced trip events removed from MCPWM
TZFRC.OST	-	Software-forced trip events removed from MCPWM
TZOSTCLR.DCAEVT1	-	Digital Compare Submodule removed
TZOSTCLR.DCBEVT1	-	Digital Compare Submodule removed
TZOSTCLR.OST1	TZCBCOSTCLR.OST1	CBC and OST Trip Flag clear registers combined
TZOSTCLR.OST2	TZCBCOSTCLR.OST2	CBC and OST Trip Flag clear registers combined
TZOSTCLR.OST3	TZCBCOSTCLR.OST3	CBC and OST Trip Flag clear registers combined
TZOSTCLR.OST4	TZCBCOSTCLR.OST4	CBC and OST Trip Flag clear registers combined
TZOSTCLR.OST5	TZCBCOSTCLR.OST5	CBC and OST Trip Flag clear registers combined
TZOSTCLR.OST6	TZCBCOSTCLR.OST6	CBC and OST Trip Flag clear registers combined
	TZCBCOSTCLR.OST7	CBC and OST Trip Flag clear registers combined
	TZCBCOSTCLR.OST8	CBC and OST Trip Flag clear registers combined
TZOSTFLG.DCAEVT1	-	Digital Compare Submodule removed
TZOSTFLG.DCBEVT1	-	Digital Compare Submodule removed
TZOSTFLG.OST1	TZCBCOSTFLAG.OST1	CBC and OST Trip Flag clear registers combined
TZOSTFLG.OST2	TZCBCOSTFLAG.OST2	CBC and OST Trip Flag clear registers combined
TZOSTFLG.OST3	TZCBCOSTFLAG.OST3	CBC and OST Trip Flag clear registers combined

Table 7-1. Trip-Zone Submodule Register Differences Between EPWM and MCPWM (continued)

EPWM	MCPWM	Description
TZOSTFLG.OST4	TZCBCOSTFLAG.OST4	CBC and OST Trip Flag clear registers combined
TZOSTFLG.OST5	TZCBCOSTFLAG.OST5	CBC and OST Trip Flag clear registers combined
TZOSTFLG.OST6	TZCBCOSTFLAG.OST6	CBC and OST Trip Flag clear registers combined
-	TZCBCOSTFLAG.OST7	Additional TZ7 Trip Signal
-	TZCBCOSTFLAG.OST8	Additional TZ8 Trip Signal
TZSEL.DCAEVT1	-	Digital Compare Submodule removed
TZSEL.DCAEVT2	-	Digital Compare Submodule removed
TZSEL.DCBEVT1	-	Digital Compare Submodule removed
TZSEL.DCBEVT2	-	Digital Compare Submodule removed
TZSEL.OSHT1	TZSEL.OST1	Bitfield renamed
TZSEL.OSHT2	TZSEL.OST2	Bitfield renamed
TZSEL.OSHT3	TZSEL.OST3	Bitfield renamed
TZSEL.OSHT4	TZSEL.OST4	Bitfield renamed
TZSEL.OSHT5	TZSEL.OST5	Bitfield renamed
TZSEL.OSHT6	TZSEL.OST6	Bitfield renamed
-	TZSEL.OST7	Additional TZ7 Trip Signal
-	TZSEL.OST8	Additional TZ8 Trip Signal

7.2 Driverlib Differences

The Trip-Zone Submodule driverlib differences between MCPWM and EPWM are listed in Table 7-2.

Table 7-2. Trip-Zone Submodule Driverlib Differences Between MCPWM and EPWM

EPWM Function	MCPWM Function	Explanation
EPWM_clearCycleByCycleTripZoneFlag	MCPWM_clearTripZoneFlagStatus	One-shot and CBC Trip registers combined
EPWM_clearOneShotTripZoneFlag	MCPWM_clearTripZoneFlagStatus	One-shot and CBC Trip registers combined
EPWM_clearTripZoneFlag	MCPWM_clearTripZoneFlagStatus	EPWM only
EPWM_disableTripZoneAdvAction	-	Advanced tripping behavior not present on MCPWM
EPWM_disableTripZoneInterrupt	-	Trip Interrupts not present on MCPWM
EPWM_enableTripZoneAdvAction	-	Advanced tripping behavior not present on MCPWM
EPWM_enableTripZoneInterrupt	-	Trip Interrupts not present on MCPWM
EPWM_forceTripZoneEvent	-	Software-forced trips not present on MCPWM
EPWM_getCycleByCycleTripZoneFlagStatus	MCPWM_getTripZoneFlagStatus	One-shot and CBC Trip registers combined
EPWM_getOneShotTripZoneFlagStatus	MCPWM_getTripZoneFlagStatus	One-shot and CBC Trip registers combined
EPWM_selectCycleByCycleTripZoneClearEvent	MCPWM_selectTripZoneCBCEvent	Function rename
EPWM_setTripZoneAdvAction	-	Advanced tripping behavior not present on MCPWM

Table 7-2. Trip-Zone Submodule Driverlib Differences Between MCPWM and EPWM (continued)

EPWM Function	MCPWM Function	Explanation
EPWM_setTripZoneAdvDigitalCompareActionA	-	Advanced tripping behavior not present on MCPWM
EPWM_setTripZoneAdvDigitalCompareActionB	-	Advanced tripping behavior not present on MCPWM
EPWM_setTripZoneDigitalCompareEventCondition	-	Digital Compare Submodule removed from MCPWM

8 Event-Trigger Submodule Differences

The Event-Trigger Submodule is responsible for triggering CPU interrupts and ADC start-of-conversion events. The feature differences of the MCPWM Event-Trigger Submodule when compared to EPWM are as follows:

- Two separate ET events (ET1 and ET2) can be configured independently
- Pre-scaling feature retained, however pre-scaler counters cannot be initialized with custom value (must always start at 0)
- Pre-scaler counters reduced from 4-bit to 3-bit counters
- Addition of SOCC and SOCD which can be configured independently from SOCA and SOCB
- Flags to indicate SOCA/SOCB events have occurred are removed from MCPWM

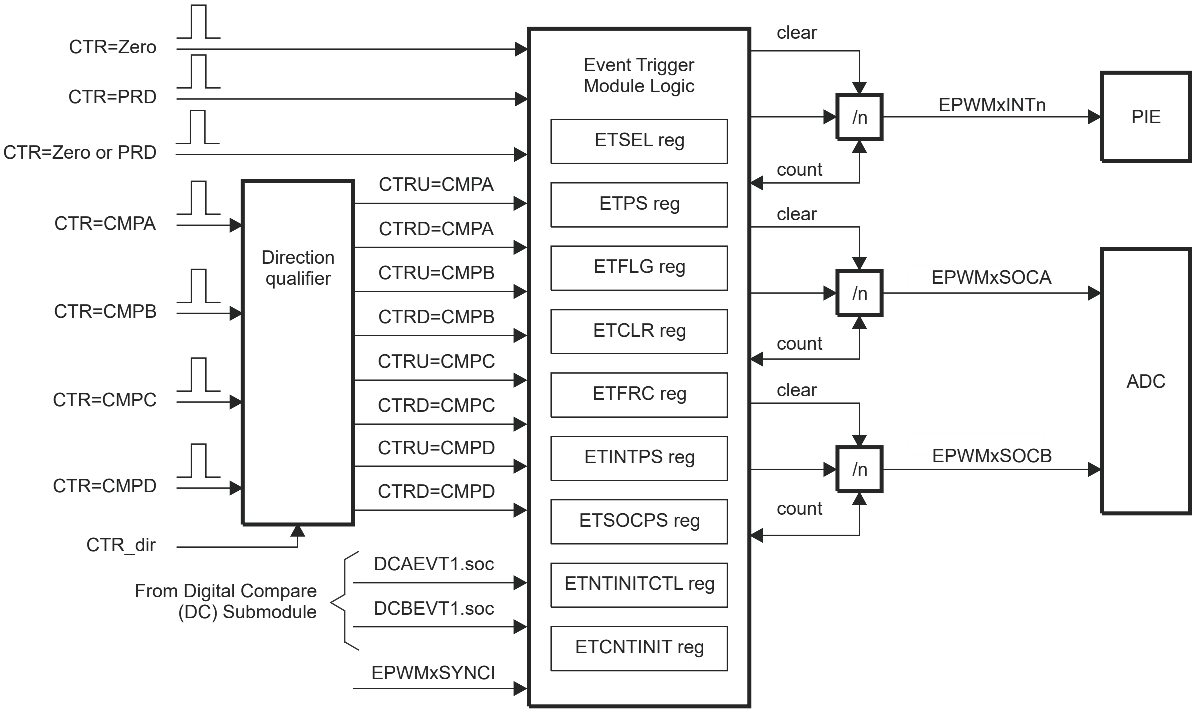


Figure 8-1. EPWM Event-Trigger Block Diagram

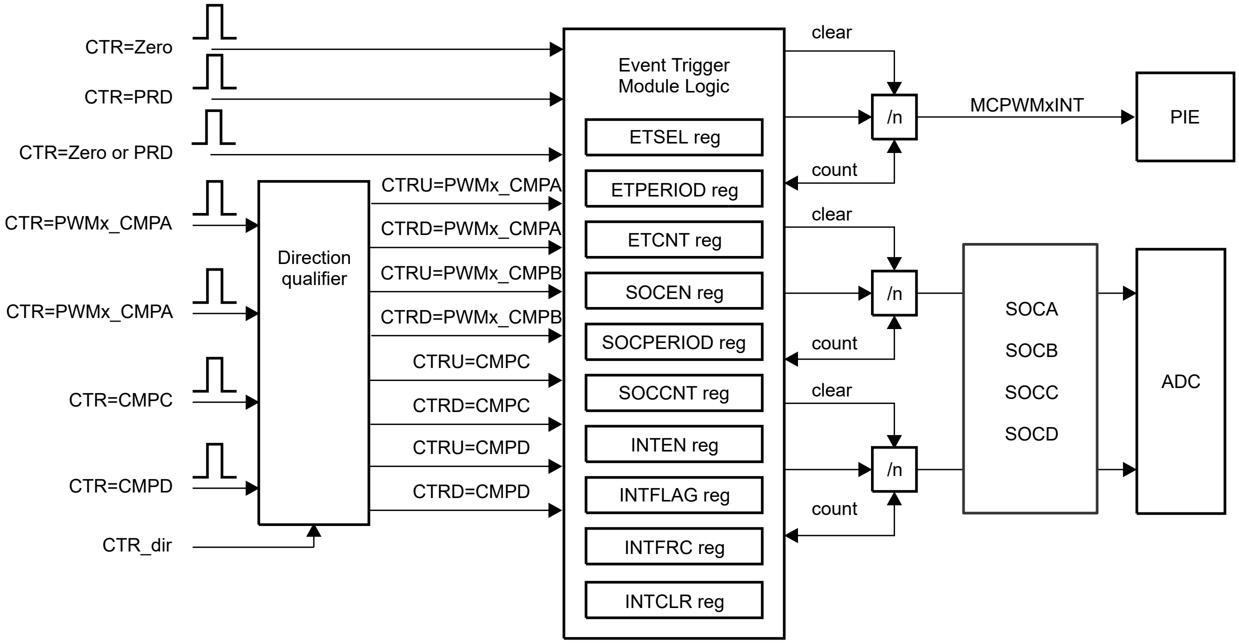


Figure 8-2. MCPWM Event-Trigger Block Diagram

8.1 Register Differences

The Event-Trigger Submodule register differences between EPWM and MCPWM are listed in Table 8-1.

Table 8-1. Event-Trigger Submodule Register Differences Between EPWM and MCPWM

EPWM	MCPWM	Description
ETCLR.INT	INTCLR.INT	Clear Interrupt bit register renamed
	INTCLR.ET1	New interrupt status clear bits for ET1
	INTCLR.ET2	New interrupt status clear bits for ET2
	INTCLR.CNT_OVF	New interrupt status clear bits for TBCTR overflow
	INTCLR.CBC	New interrupt status clear bits for CBC trip
	INTCLR.OST	New interrupt status clear bits for one-shot trip
ETCLR.SOCA	SOCCLR.SOCA	New Register for clearing SOCx events
ETCLR.SOCB	SOCCLR.SOCB	New Register for clearing SOCx events
	SOCCLR.SOCC	New Register for clearing SOCx events
	SOCCLR.SOCD	New Register for clearing SOCx events
ETCNTINIT.INTINIT	-	Prescaler custom init value feature removed
ETCNTINIT.SOCAINIT	-	Prescaler custom init value feature removed
ETCNTINIT.SOCBINIT	-	Prescaler custom init value feature removed
ETCNTINITCTL.INTINITEN	-	Prescaler custom init value feature removed
ETCNTINITCTL.INTINITFRC	-	Prescaler custom init value feature removed
ETCNTINITCTL.SOCAINITEN	-	Prescaler custom init value feature removed
ETCNTINITCTL.SOCAINITFRC	-	Prescaler custom init value feature removed
ETCNTINITCTL.SOCBINITEN	-	Prescaler custom init value feature removed
ETCNTINITCTL.SOCBINITFRC	-	Prescaler custom init value feature removed

Table 8-1. Event-Trigger Submodule Register Differences Between EPWM and MCPWM (continued)

EPWM	MCPWM	Description
ETFLG.INT	INTFLAG.INT	Register rename (INTFLAG)
	INTFLAG.CBC	New flag bit for CBC interrupt
	INTFLAG.CNT_OVF	New flag bit for TBCTL overflow interrupt
	INTFLAG.ET1	New flag bit for ET1 interrupt
	INTFLAG.ET2	New flag bit for ET2 interrupt
	INTFLAG.OST	New flag bit for OST interrupt
ETFLG.SOCA	SOCFLAG.SOCA	New register for SOCx flags
ETFLG.SOCB	SOCFLAG.SOCB	New register for SOCx flags
	SOCFLAG.SOCC	New register for SOCx flags
	SOCFLAG.SOCD	New register for SOCx flags
ETFRC.INT	-	Global interrupt force removed
ETFRC.SOCA	-	SOCA force removed
ETFRC.SOCB	-	SOCB force removed
ETINTPS.INTCNT2	-	Expanded prescaler counter removed
ETINTPS.INTPRD2	-	Expanded prescaler period removed
ETPS.INTCNT	-	Event Trigger INT separated into ET1 and ET2
	ETCNT.ET1_CNT	Event Trigger INT separated into ET1 and ET2
	ETCNT.ET2_CNT	Event Trigger INT separated into ET1 and ET2
ETPS.INTPRD	-	Event Trigger INT separated into ET1 and ET2
	ETPERIOD.ET1_PERIOD	Event Trigger INT separated into ET1 and ET2
	ETPERIOD.ET2_PERIOD	Event Trigger INT separated into ET1 and ET2
ETPS.INTPSSEL	-	Expanded prescaler counter removed
ETPS.SOCACNT	SOCCNT.SOCA_CNT	Register rename
ETPS.SOCAPRD	SOCPERIOD.SOCA_PERIOD	Register rename
ETPS.SOCBCNT	SOCCNT.SOCB_CNT	Register rename
ETPS.SOCBPRD	SOCPERIOD.SOCB_PERIOD	Register rename
	SOCCNT.SOCC_CNT	Additional SOCC prescaler counter
	SOCCNT.SOCD_CNT	Additional SOCD prescaler counter
	SOCPERIOD.SOCC_PERIOD	Additional SOCC prescaler counter
	SOCPERIOD.SOCD_PERIOD	Additional SOCD prescaler counter
ETPS.SOCPSSEL	-	Expanded prescaler counter removed
ETSEL.INTEN	-	Event Trigger INT separated into ET1 and ET2
ETSEL.INTSEL	-	Event Trigger INT separated into ET1 and ET2
	INTEN.CBC	New en bit for CBC interrupt

Table 8-1. Event-Trigger Submodule Register Differences Between EPWM and MCPWM (continued)

EPWM	MCPWM	Description
	INTEN.CNT_OVF	New en bit for TBCTL overflow interrupt
	INTEN.ET1	New en bit for ET1 interrupt
	INTEN.ET2	New en bit for ET2 interrupt
	INTEN.OST	New en bit for OST interrupt
	ETSEL.ET1_SEL	Event Trigger INT separated into ET1 and ET2
	ETSEL.ET2_SEL	Event Trigger INT separated into ET1 and ET2
ETSEL.INTSELCMP	-	EPWMxINT Compare Select
ETSEL.SOCAEN	SOCEN.SOCA_ENABLE	Start of Conversion A Enable
ETSEL.SOCASEL	SOCSEL.SOCA_SEL	Start of Conversion A Select
ETSEL.SOCASELCMP	-	Not needed since SOCC and SOCD events are independent events
ETSEL.SOCBEN	SOCEN.SOCB_ENABLE	Register rename
ETSEL.SOCBSEL	SOCSEL.SOCB_SEL	Register rename
ETSEL.SOCBSELCMP	-	Not needed since SOCC and SOCD events are independent events
	SOCEN.SOCC_ENABLE	New SOCC and SOCD events
	SOCEN.SOCD_ENABLE	New SOCC and SOCD events
	SOCSEL.SOCC_SEL	New SOCC and SOCD events
	SOCSEL.SOCD_SEL	New SOCC and SOCD events
ETSOCPS.SOCACNT2	-	Expanded prescaler counter removed
ETSOCPS.SOCAPRD2	-	Expanded prescaler counter removed
ETSOCPS.SOCBCNT2	-	Expanded prescaler counter removed
ETSOCPS.SOCBPRD2	-	Expanded prescaler counter removed
-	INTFRC.CBC	Independent software force bits for each interrupt source
-	INTFRC.CNT_OVF	Independent software force bits for each interrupt source
-	INTFRC.ET1	Independent software force bits for each interrupt source
-	INTFRC.ET2	Independent software force bits for each interrupt source
-	INTFRC.OST	Independent software force bits for each interrupt source

8.2 Driverlib Differences

The Event-Trigger submodule driverlib differences between EPWM and MCPWM are listed in Table 8-2.

Table 8-2. Event-Trigger Submodule Driverlib Differences Between EPWM and MCPWM

EPWM Function	MCPWM Function	Explanation
-	MCPWM_clearGlobalInterrupt	Addition of global interrupt clear flag
-	MCPWM_getEventTriggerEventCount	Separate prescaler counters for ET1 and ET2
-	MCPWM_getGlobalInterruptStatus	Addition of global interrupt flag
-	MCPWM_setEventTriggerEventPrescale	Separate prescaler counters for ET1 and ET2
-	MCPWM_setEventTriggerSource	Separate ET1 and ET2 instead of INT and SOCx
EPWM_clearEventTriggerInterruptFlag	MCPWM_clearInterrupt	Global interrupt clear bit on MCPWM
EPWM_disableADCTriggerEventCountInit	-	Removed customer prescaler counter init value
EPWM_disableInterruptEventCountInit	-	Removed customer prescaler counter init value
EPWM_enableADCTriggerEventCountInit	-	Removed customer prescaler counter init value
EPWM_enableInterruptEventCountInit	-	Removed customer prescaler counter init value
EPWM_forceADCTrigger	-	Separate ET1 and ET2 instead of INT and SOCx
EPWM_forceADCTriggerEventCountInit	-	Separate ET1 and ET2 instead of INT and SOCx
EPWM_forceEventTriggerInterrupt	MCPWM_forceInterrupt	Changes in interrupt force structure
EPWM_forceInterruptEventCountInit	-	Removed customer prescaler counter init value
EPWM_getInterruptEventCount	-	Separate ET1 and ET2 instead of INT and SOCx
EPWM_setADCTriggerEventCountInitValue	-	Separate ET1 and ET2 instead of INT and SOCx
EPWM_setInterruptEventCount	-	Separate ET1 and ET2 instead of INT and SOCx
EPWM_setInterruptEventCountInitValue	-	Removed customer prescaler counter init value
EPWM_setInterruptSource	-	Separate ET1 and ET2 instead of INT and SOCx

9 Global Load Differences

The global load feature allows for simultaneous loading of several shadow to active registers triggered by hardware or software events. The MCPWM global load feature differences when compared to EPWM are as follows:

- SYNCIN signal cannot be used to trigger global load
- Global load prescaler counter removed from MCPWM
- Global load for action qualifier continuous software force removed
- Global load for CMPC / CMPD removed
- Global load for DBCTL removed

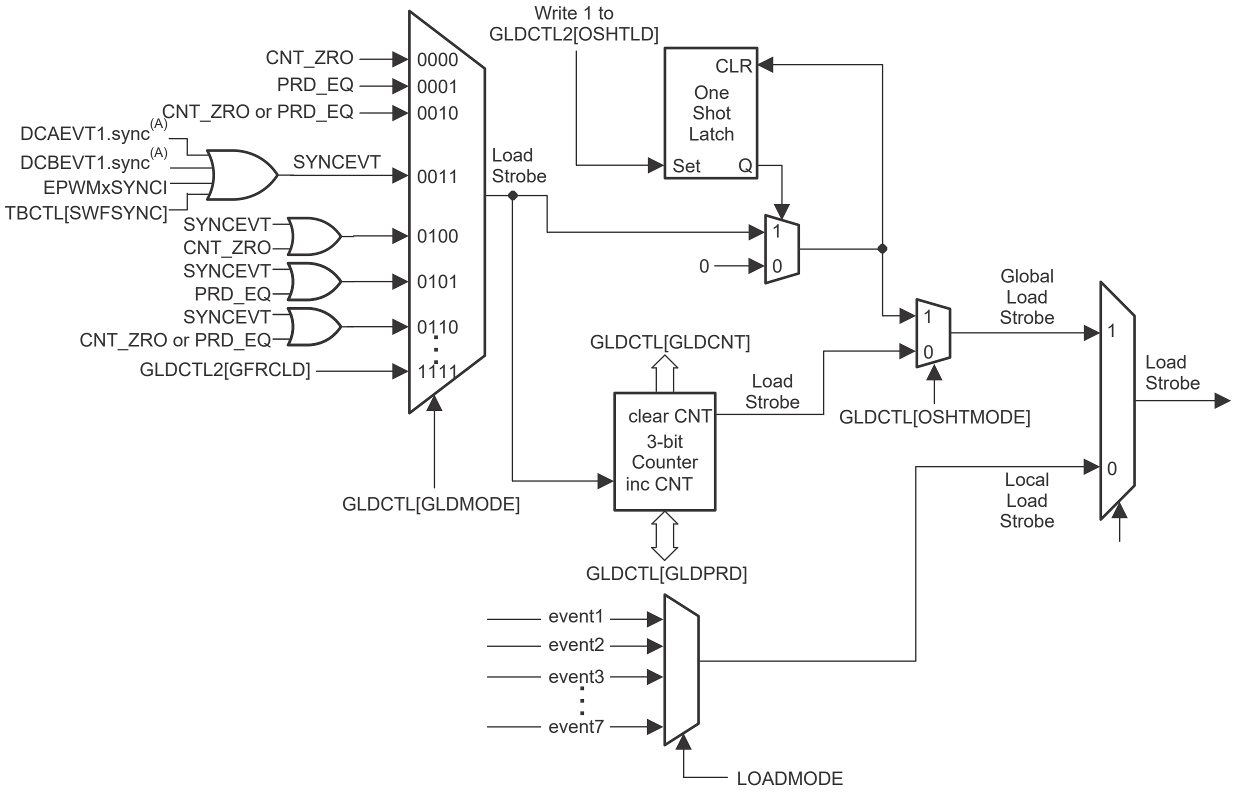


Figure 9-1. EPWM Global Load Block Diagram

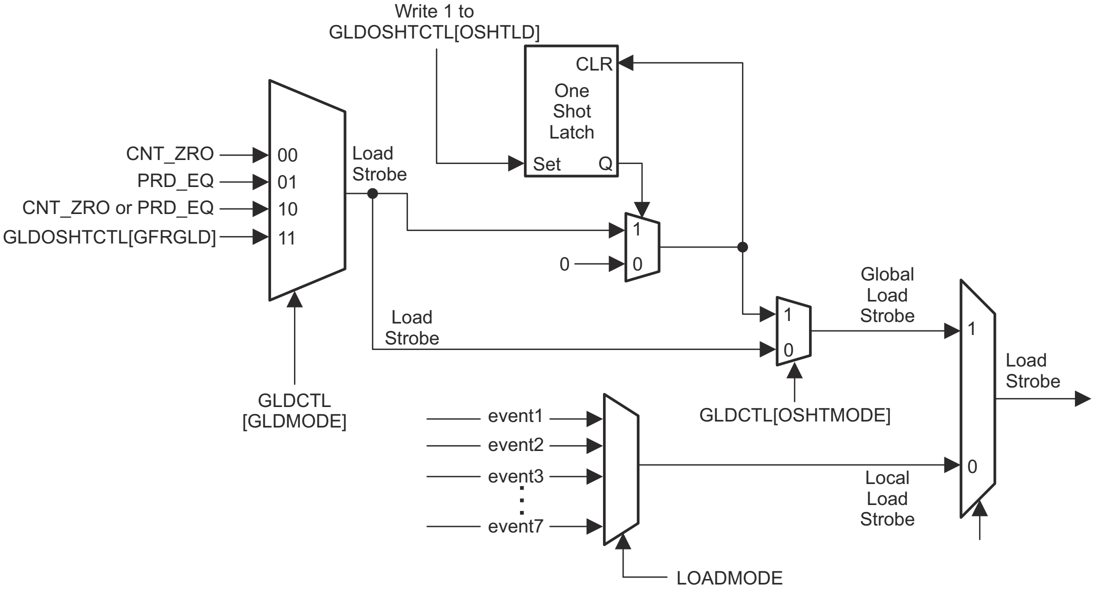


Figure 9-2. MCPWM Global Load Block Diagram

9.1 Register Differences

The Global Load register differences between EPWM and MCPWM are listed in Table 9-1.

Table 9-1. Global Load Register Differences Between EPWM and MCPWM

EPWM	MCPWM	Description
-	**GLDOSHTSTS.OSHTLATCH**	**Addition of one-shot latch status register**
GLDCFG.AQCSFRC	-	Global load removed for action qualifier continuous software force
GLDCFG.AQCTLA_AQCTLA2	-	T1/T2 events removed from MCPWM
GLDCFG.AQCTLB_AQCTLB2	-	T1/T2 events removed from MCPWM
GLDCFG.CMPA_CMPAHR	-	HRPWM not present on MCPWM
GLDCFG.CMPB_CMPBHR	-	HRPWM not present on MCPWM
GLDCFG.CMPC	-	Global load for CMPC / CMPD removed
GLDCFG.CMPD	-	Global load for CMPC / CMPD removed
GLDCFG.DBCTL	-	Global load for DBCTL removed
GLDCFG.DBFED_DBFEDHR	-	HRPWM not present on MCPWM
GLDCFG.DBRED_DBREDHR	-	HRPWM not present on MCPWM
GLDCFG.TBPRD_TBPRDHR	-	HRPWM not present on MCPWM
GLDCTL.GLDCNT	-	Global load prescaler counter removed
GLDCTL.GLDPRD	-	Global load prescaler counter removed
GLDCTL2.GFRCLD	GLDOSHTCTL.GFRCLD	Register rename
GLDCTL2.OSHTLD	GLDOSHTCTL.OSHTLD	Register rename, bit not cleared automatically by hardware
	GLDOSHTCTL.OSHTCLR	Addition of new bit to manually clear one-shot load bit
LOCK.GLLOCK	-	LOCK register removed from MCPWM

9.2 Driverlib Differences

The Global Load Driverlib differences between EPWM and MCPWM are listed in Table 9-2.

Table 9-2. Global Load Driverlib differences between EPWM and MCPWM

EPWM Function	MCPWM Function	Explanation
-	MCPWM_clearGlobalLoadOneShotLatch	Addition of clear bit for one-shot latch
-	MCPWM_getGlobalLoadOneShotLatchStatus	Addition of status bit for one-shot latch
EPWM_disableGlobalLoadRegisters	-	Individual enable bits for global load not present on MCPWM
EPWM_enableGlobalLoadRegisters	-	Individual enable bits for global load not present on MCPWM
EPWM_getGlobalLoadEventCount	-	Global load prescaler counter removed
EPWM_setGlobalLoadEventPrescale	-	Global load prescaler counter removed

10 Summary

The MCPWM module is a low-cost version of the EPWM module. Due to the reduced feature-set of MCPWM, and the combination of three PWM pairs into a single MCPWM instance, the programming procedure for the two modules have significant differences which are addressed in this document.

11 References

- Texas Instruments, F28E12x Real-Time Microcontrollers Technical Reference Manual

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265