Application Report

CRM/ZVS PFC Implementation Based on C2000™ Type-4 PWM Module

Minghan Dong, Aki Li, Chen Jiang, and Shamim Choudhury

ABSTRACT

This application report proposes a suitable configuration of type-4 ePWM module (Devices Covered: 2807x, 2837xD, 2837xS, 28004x, 28002x, 2838x) in order to implement CRM/ZVS switching for boost PFC topology. It presents the implementation details of CRM/ZVS boost PFC with TI's C2000™ microcontrollers type-4 ePWM, including registers configuration and system block diagrams demonstration. F28002x Control card is used to verify the firmware and peripherals operation. Detailed waveforms and test results along with an implementation guidance of the firmware are provided.

Table of Contents

List of Figures

Trademarks

C2000™ is a trademark of Texas Instruments.
All trademarks are the property of their respective owners.

1 Introduction

Increasing power conversion efficiency is an essential topic in recent years. In this way, many topologies and methods have been proposed to reduce conduction losses and switching losses in power devices. CRM and ZVS technique are critical soft-switching techniques to improve efficiency and reduce power devices losses.

A digitally controlled PFC converter is shown in Figure 2-1. For heavy PFC load, inductor current i_L can be continuous-conduction mode (CCM). While, at light load current may become discontinuous (DCM). During DCM operation, the boost inductor L resonates freely with PFC MOSFET output capacitance C_{ds}. Digital PFC designers can use external components and internal algorithm to detect MOSFET Vds's ZVS or CRM position cycle by cycle. C2000 type-4 PWM provides internal hardware to realize CRM/ZVS PFC with reduced MCU bandwidth utilization and without using these external components.

2 CRM/ZVS PFC

Figure 2-1 shows a conventional boost PFC topology. The inductor current $i_L$can maintain the critical-conduction mode (CRM). When the inductor current become discontinuous, the MOSFET's V_{ds} voltage can resonate to zero and thereby create a ZVS switching instant. Figure 2 demonstrates PWM signal, MOSFET's V_{ds} and inductor current waveform in CRM/ZVS status.

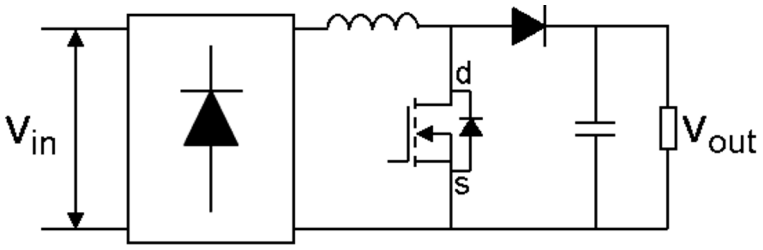


Figure 2-1. Boost PFC Topology

If AC input voltage is higher than half of DC bus voltage, the MOSFET should be turned ON at the valley point of the voltage to achieve ZCS(CRM). The MOSFET V_{ds} never resonate to zero as shown in Figure 2-2. However, when AC input voltage is lower than 1/2 output voltage, the MOSFET's V_{ds} can resonate to zero volts and be clamped by the MOSFET body diode. MOSFET can achieve ZVS&ZCS in this status as shown in Figure 2-3.

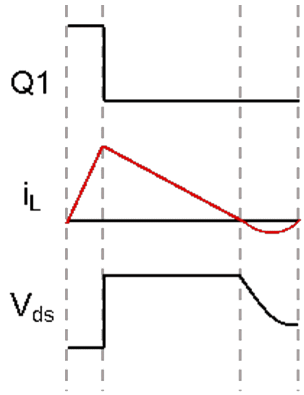


Figure 2-2. PFC Waveforms in CRM/ZVS Status: Vin>Vout/2

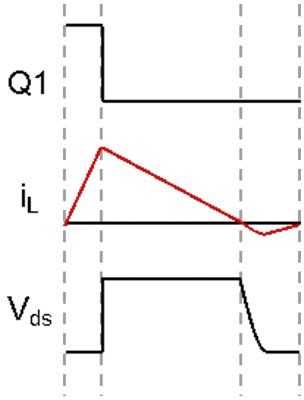


Figure 2-3. PFC Waveforms in CRM/ZVS Status: Vin<Vout/2

The traditional predict-period approach requires fast calculation and consumes significant CPU bandwidth. The extra resource is used for calculating the resonant time and the duration that the boost inductor current i_L return to zero from peak. This report proposes a MCU internal hardware-based method to realize CRM/ZVS PFC control by leveraging peripherals of C2000 MCU. These internal peripherals include CMPSS, cross bar (X-BAR) and type-4 PWM.

3 PFC CRM/ZVS Realization Based on Type-4 PWM

Figure 3-1 shows the signal interface from external PFC power stage hardware to C2000 internal peripherals to generate PWM signal. This report takes CMPSS5, X-BAR(TRIP5), ePWM1 as an example to explain the realization of CRM/ZVS PFC by type-4 PWM.

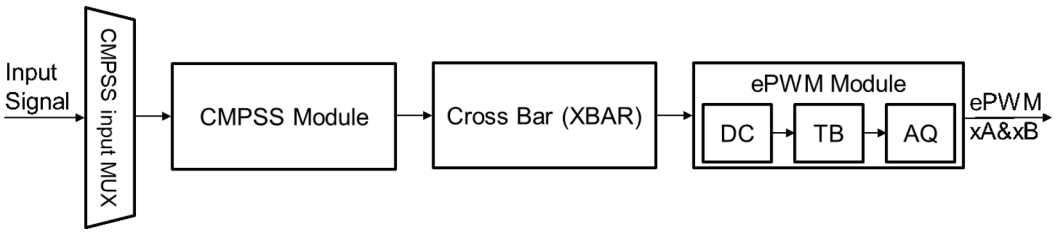


Figure 3-1. Signal Flow Chart

The input signal in Figure 3-1 is the PFC inductor current iL. This is chosen as an input to COMPH positive pin of the CMPSS module as detailed in Figure 3-2. The threshold for capturing current zero crossing (ZC) is a fixed value close to zero and applied to DACHVALS (shadow register DAC value). The current ZC threshold is set as the negative input of COMPH through the reference DAC. The invert output of comparator is chosen as CTRIPH and connected to X-BAR asynchronously.

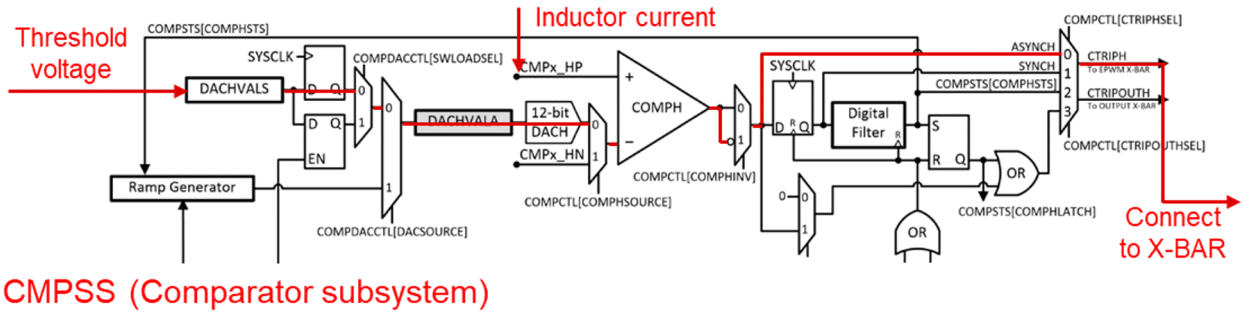


Figure 3-2. C2000 MCU CMPSS Block Diagram

The CMPSS5 CTRIPH signal connects to X-BAR TRIP5 mux8 as shown in Figure 3-3. In order to pass any signal through to the ePWM, enable the mux8 in the TRIP5MUXENABLE register. CMPSS output is fed into ePWM DC sub-module through ePWM X-BAR. TRIP5, refers to the inductor current negative zero crossing (NZC) event, generates DCAH signals by setting DCAHTRIPSEL.Note that positive zero crossing event is filtered by blanking window as shown in the Figure 3-7. Then, the configuration of TZDCSEL register qualifies the actions on the selected DCAH signal, which generates the DCAEVT1 event. The DCAEVT1 event then is filtered to provide DCAEVT1.sync with a filtered version of the event signals by configuring DCACTL. Then DCAEVT1.sync generates a synchronization pulse to the time-base counter.

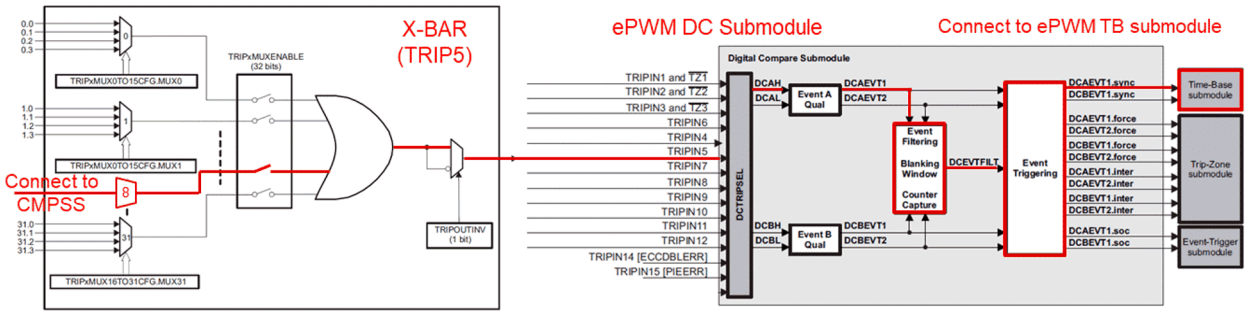


Figure 3-3. X-BAR and ePWM DC Module Interface

Detailed event filtering logic block as shown in Figure 3-4 is critical to realize ZVS/ZCS. The event filtering block can capture the TBCTR value of the trip event to provide control algorithm with PWM period and store the period value in DCCAP register. The blanking logic is used to filter out potential noise and inductor current positive zero crossing (PZC) event on the signal prior to generating right sync signal. Figure 3-4 shows signal flow in the event filtering logic module. Select the DCAEVT1 events as an input to the event filtering logic block (DCFCTL[SRCSEL]) with a blanking window (Blank Control Logic). This is where the comparator output is selected as an input to the event filtering logic block. Blanking window is ANDed with DCAEVT1. Software trigger source is selected as an event to reset and restart the edge filter (VCAPCTL[TRIGSEL]). At the end, configure VCAPCTL[EDGEFILTDLYSEL] to apply delay (a software programmed value in SWVDELVAL) based on the hardware parameters. Now the DCCAP can capture the TBCTR value of the event associated with the inductor current negative zero crossing. This TBCTR value represents the period of the present PWM switching cycle.

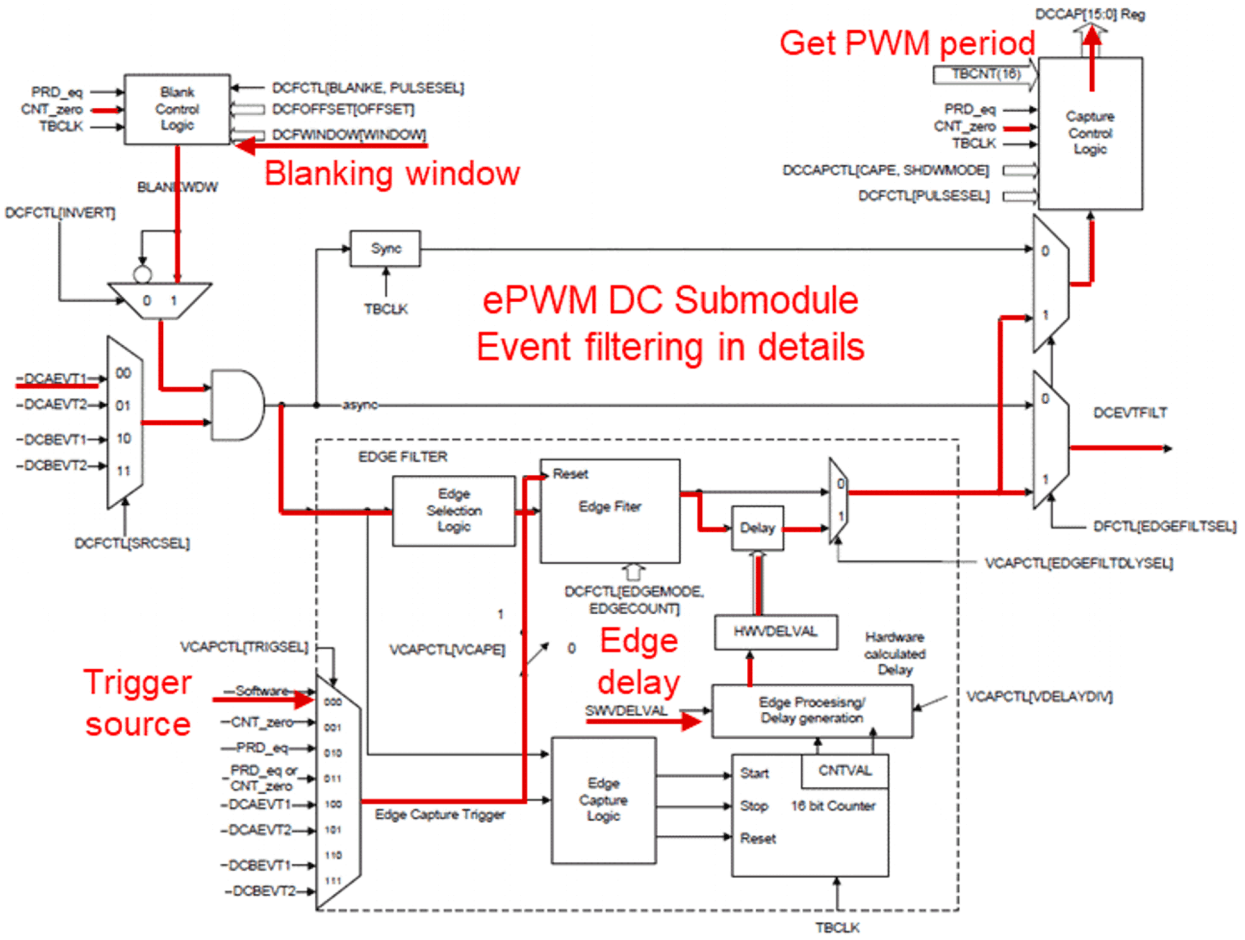


Figure 3-4. Edge Filter and Capture Control Logic in DC Submodule

At the end, DCAEVT1.sync is enabled by set DCACTL[EVT1SYNCE] of event triggering block as shown in Figure 3-5 and phase shift is enabled by setting TBCTL[PHSEN] of time base sub-module to sync TBCTR as shown in Figure 3-6.

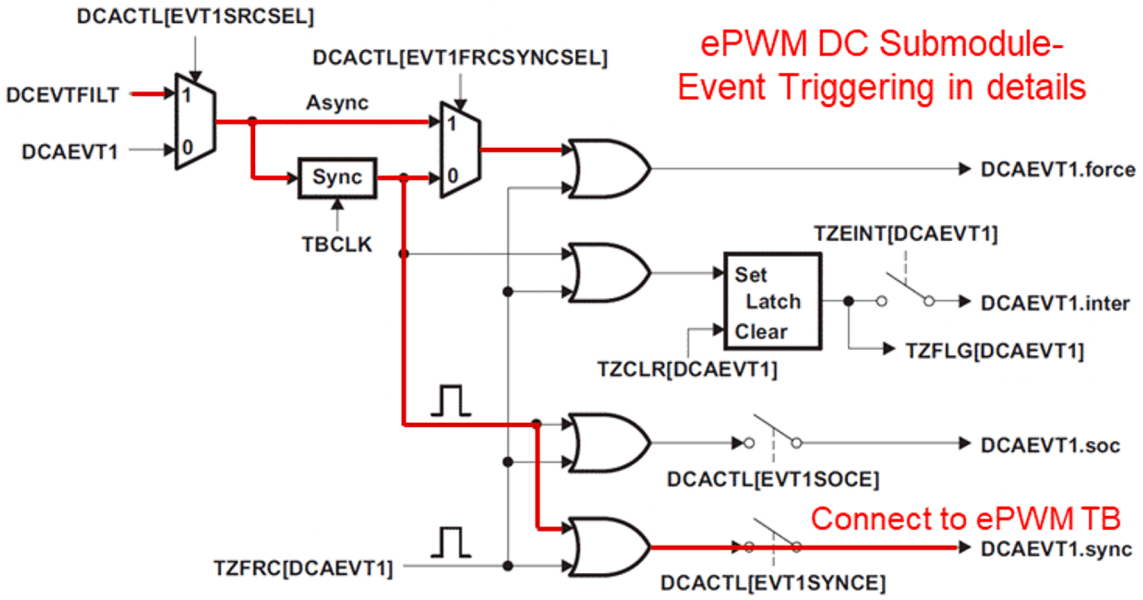


Figure 3-5. Event Triggering Block Used in CRM/ZVS PFC

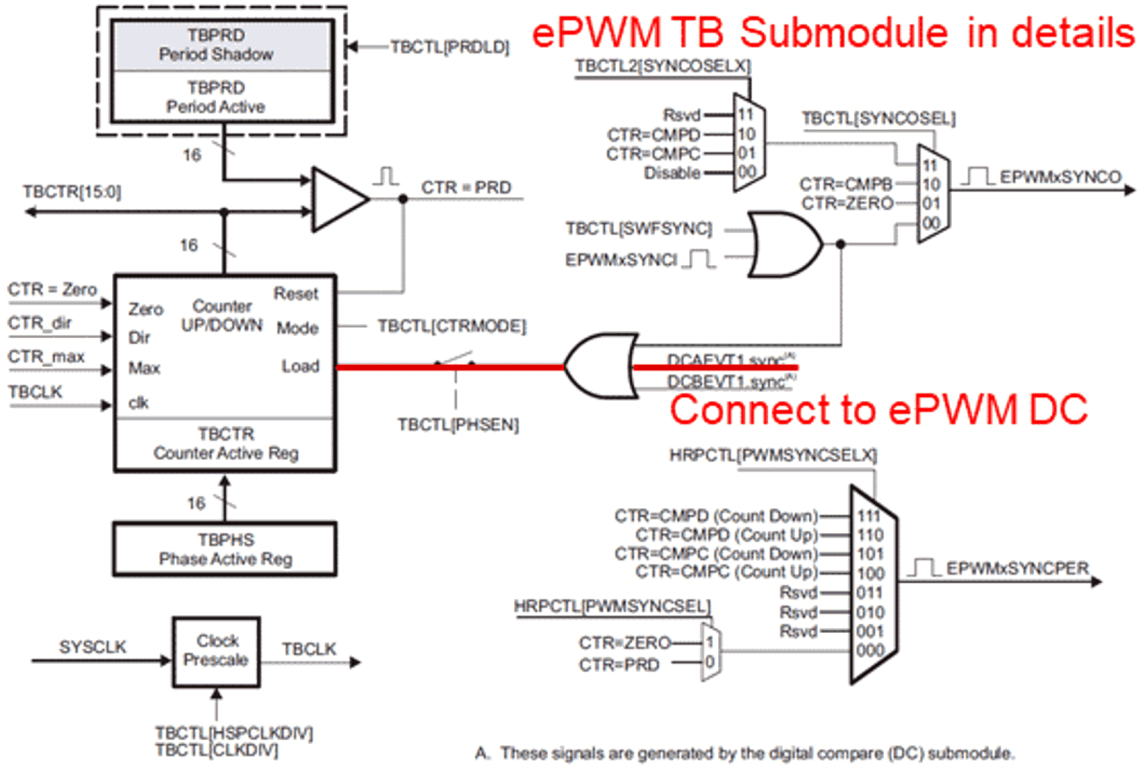


Figure 3-6. Time Base Submodule Used in CRM/ZVS PFC

Figure 3-7 shows the key signal waveforms after the peripheral configuration. The rising and falling edge of DCAEVT1 signal represents positive and negative zero crossing point accordingly based on Figure 3-7. Blanking window is ANDed with DCAEVT1 to generate DCEVTFILT so that inductor current positive zero crossing (PZC) edge and noise during the MOSFET on-time are filtered. In this way, the blanking window length should be longer than PWM on-time. DC sub-module applies a delay corresponding to inductor negative zero crossing (NZC) edge to generate sync event.

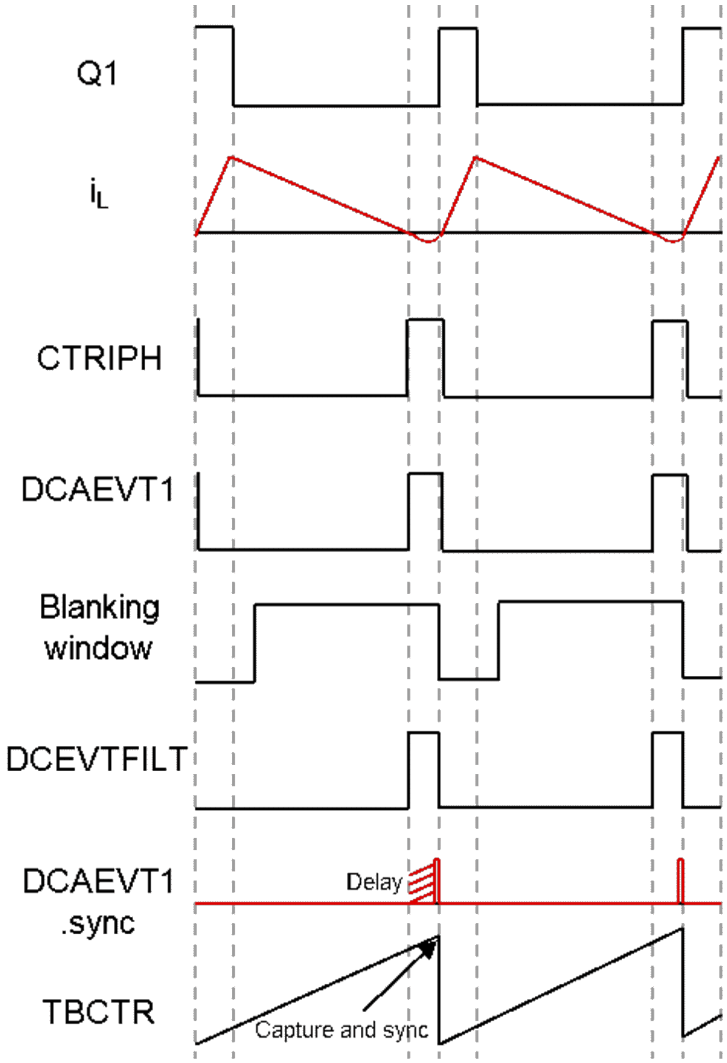


Figure 3-7. Signal Waveform in CRM PFC Type-4 PWM Implementation

4 Demo Code and Flow Charts

The flow chart of the realization process is shown in Figure 4-1. CMPSS, X-BAR and ePWM are configured in start status.

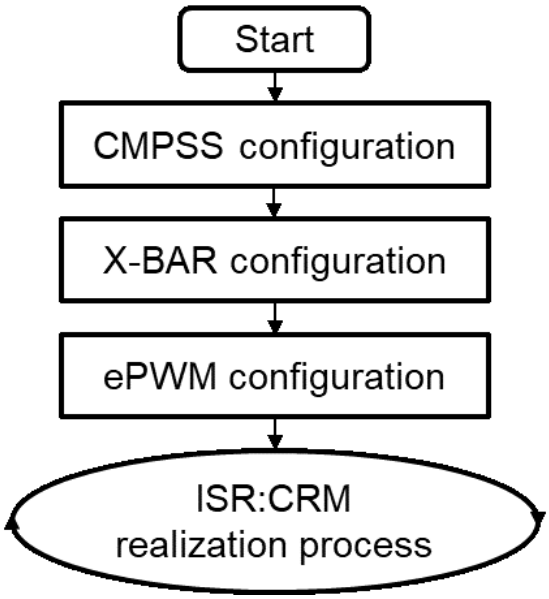


Figure 4-1. Flow Chart of CRM/ZVS Realization Code

4.1 Peripheral Configuration

```
//-----------------------------------------------------------------------
// CMPSS5H Capture ZVS (Inductor current NZC)
//-----------------------------------------------------------------------
//setup analog input of CMPSS 5, connect inductor current signal to CMPSS5
ASysCtl_selectCMPHPMux(ASYSCTL_CMPHPMUX_SELECT_5, 2);
// Power up Comparator locally
CMPSS_enableModule(base5);
// Connect the inverting input to internal DAC. Invert CMPH output.
CMPSS_configHighComparator(base5, CMPSS_INSRC_DAC|CMPSS_INV_INVERTED);
// Set DAC output
CMPSS_setDACValueHigh(base5, (int16_t)(iL_limit));
// DACHVALA is updated from DACHVALS (no Internal ramp)
CMPSS_configDAC(base5, CMPSS_DACSRC_SHDW);
// Output = asynch comparator output (output inverted)
CMPSS_configOutputsHigh(base5, CMPSS_TRIP_ASYNC_COMP);
// Xbar setting
//-----------------------------------------------------------------------
// CMPSS5H - ZVS (Inductor current NZC)
//-----------------------------------------------------------------------
XBAR_setEPWMMuxConfig(XBAR_TRIP5, ZVS_CAPTURE_XBAR_MUX_CONFIG);
XBAR_enableEPWMMux(XBAR_TRIP5, ZVS_CAPTURE_XBAR_MUX);
//Clear XBAR flag during initialization
XBAR_clearInputFlag(XBAR_INPUT_FLG_CMPSS2_CTRIPH);
//*************************************************************************
// Configure PWM1A for switching Frequency and valley switching.
//*************************************************************************
// ZVS(Inductor current NZC) related registers setup starts here
// Select one of the DCAEVT1 events as input to the event filtering logic block
EPWM_setDigitalCompareFilterInput(base1, EPWM_DC_WINDOW_SOURCE_DCAEVT1);
// Edge filter is selected
EPWM_enableDigitalCompareEdgeFilter(base1);
// Edge mode is selected
EPWM_setDigitalCompareEdgeFilterMode(base1, EPWM_DC_EDGEFILT_MODE_BOTH);
// Edge count is selected
EPWM_setDigitalCompareEdgeFilterEdgeCount(base1,1);
//Software trigger (triggered in ISR)
EPWM_setValleyTriggerSource(base1, EPWM_VALLEY_TRIGGER_EVENT_SOFTWARE);
//valley capture enabled
EPWM_enableValleyCapture(base1);
//DELAY applied
```

```
EPWM_enableValleyHWDelay(base1);
//SW DELAY duration
EPWM_setValleySWDelayValue(base1, SWDELAY);
// DCAH = Comparator 5 output = TRIP5in
EPWM_selectDigitalCompareTripInput(base1, EPWM_DC_TRIP_TRIPIN5,EPWM_DC_TYPE_DCAH);
//DCAH = high (1 TBCLK wide pulse is generated by edge filter)
EPWM_setTripZoneDigitalCompareEventCondition(base1, EPWM_TZ_DC_OUTPUT_A1,
                                             EPWM_TZ_EVENT_DCXH_HIGH);
//Source is filtered DCAEVT1
EPWM_setDigitalCompareEventSource(base1, EPWM_DC_MODULE_A, EPWM_DC_EVENT_1,
                                  EPWM_DC_EVENT_SOURCE_FILT_SIGNAL);
//Async
EPWM_setDigitalCompareEventSyncMode(base1, EPWM_DC_MODULE_A, EPWM_DC_EVENT_1,
                                    EPWM_DC_EVENT_INPUT_NOT_SYNCED);
//Use blanking window to filtering PZC edge and noise
EPWM_enableDigitalCompareBlankingWindow(base1);
EPWM_setDigitalCompareBlankingEvent(base1, EPWM_DC_WINDOW_START_TBCTR_ZERO);
// Blanking Window Offset = CMPA(n+1)
EPWM_setDigitalCompareWindowOffset(base1, 1);
// Blanking window length
EPWM_setDigitalCompareWindowLength(base1, MIN_PERIOD);
// Enable time-base counter capture
EPWM_enableDigitalCompareCounterCapture(base1);
//ZVS code stops here
```

4.2 Solution Code

The codes below are placed in ePWM1 ISR since ePWM1 provides the driving signals for PFC MOSFET. ePWM1 ISR occurs every TB counter=0.

```
//Set bit 1 (VCAPSTART), software trigger source of edge filter
EPWM_startValleyCapture(EPWM1_BASE);
//DCCAP
capturedPeriod = EPWM_getDigitalCompareCaptureCount(EPWM1_BASE);
// blankingWindowLength=50~70 ticks margin + duty*DCCAP (MOSFET on-time)
blankingWindowLength = blankingBsuffer+ePwm1Cmpb;
EPWM_setDigitalCompareWindowLength(EPWM1_BASE, blankingWindowLength);
```

5 Experimental Results

To validate the CRM implementation, prototype based on TMDSCNCD280049C is set up as shown in Figure 5-1. In Figure 5-2 and Figure 5-3, some test results are shown to verify ZVS PFC implementation based on type-4 PWM. The inductor current signal frequency is 160 kHz and 200 kHz separately. PWM can sync with inductor current NZC event and DCCAP can capture right PWM period as shown in the result.

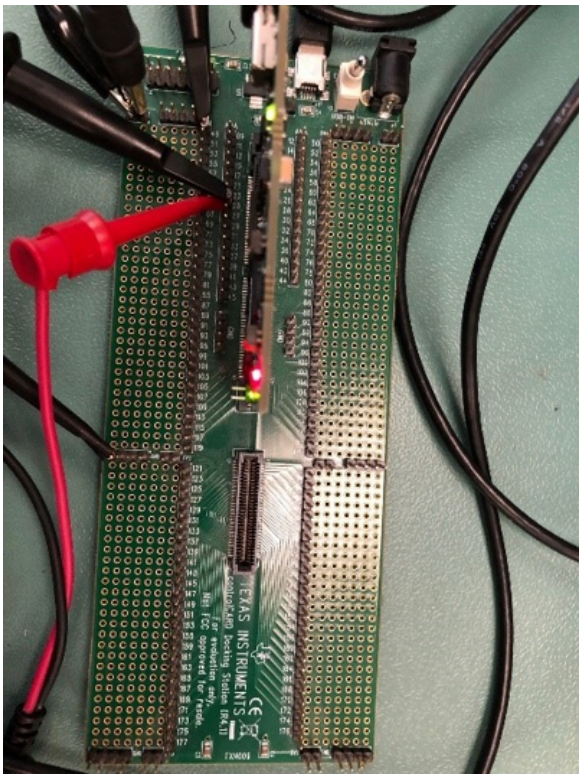
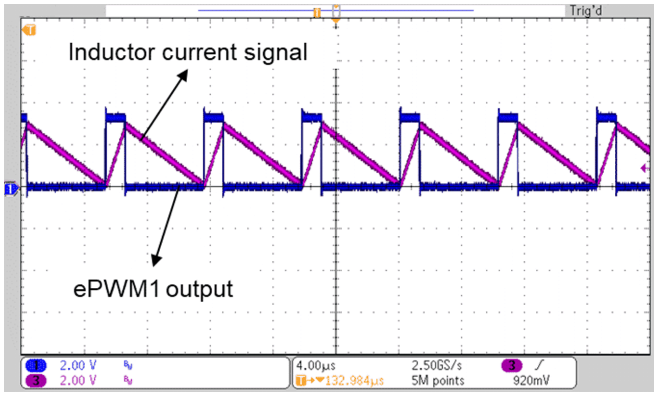
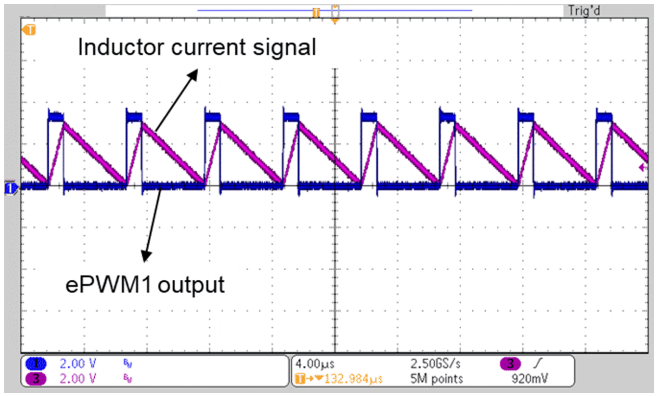


Figure 5-1. Test Prototype



vds_Th	float	100.0	0x00008036@L
blankingWindowLength	long	300	0x0000805C@D
EPwm1Regs.DCCAP	Register	622 (Decimal)	
ePwm1Cmpc	long	622	0x0000805A@D
EPwm1Regs.DCFWINDOW	Register	300 (Decimal)	
EPwm1Regs.DCFOFFSET	Register	1 (Decimal)	

Figure 5-2. 160 kHz Inductor Current Signal: ePWM1 Output Waveform and DCCAP Captured Value



blankingWindowLength	long	300	
EPwm1Regs.DCCAP	Register	497 (Decimal)	
ePwm1Cmpc	long	498	
EPwm1Regs.DCFWINDOW	Register	300 (Decimal)	
EPwm1Regs.DCFOFFSET	Register	1 (Decimal)	

Figure 5-3. 200 kHz Inductor Current Signal: ePWM1 Output Waveform and DCCAP Captured Value

6 Summary

This document introduces the CRM/ZVS PFC technique and the configuration of C2000 type-4 PWM module in order to realize this soft-switching technique. It presents an optimized digital control technique for a boost PFC converter without using any complex external logic circuits. The proposed optimization method utilizes the internal programmable digital compare block of the C2000 MCU in order to implement the CRM/ZVS switching for PFC. The proposed method is validated using a Texas Instruments TMDSCNCD280049C MCU.

7 References

- PFC Efficiency Improvement and THD Reduction at Light Loads with ZVS and Valley Switching
- Texas Instruments: *Light Load THD and Efficiency Optimization of Digitally Controlled PFC Converter with Integrated Valley Switching Control*
- Totem-Pole Boost Bridgeless PFC Rectifier with Simple Zero-Current Detection and Full-Range ZVS Operating at the Boundary of DCM/CCM
- Texas Instruments: *TMS320F28004x 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