Section 30. Power Supply PWM

HIGHLIGHTS

This section of the manual contains the following major topics:

30.1 INTRODUCTION

The Power Supply PWM module in the dsPIC30F Switch Mode Power Supply (SMPS) and Digital Power Conversion device family supports a wide variety of PWM modes and output formats. The Power Supply PWM module is ideal for power conversion applications such as the following:

- AC to DC Converters
- DC to DC Converters
- Power Factor Correction (PFC)
- Uninterruptible Power Supply (UPS)
- Inverters
- Battery Chargers
- Digital Lighting

30.2 FEATURES OVERVIEW

The Power Supply PWM module has the following features:

- Two to four PWM generators with four to eight I/O
- Two to four independent time bases
- Duty Cycle resolution of 1.1 ns at 30 MIPS
- Dead-Time resolution of 4.2 ns at 30 MIPS
- Phase Shift resolution of 4.2 ns at 30 MIPS
- Frequency resolution of 8.4 ns at 30 MIPS
- Supported PWM modes:
  - Standard Edge-Aligned PWM
  - Complementary PWM Output
  - Push-Pull PWM
  - Multi-Phase PWM
  - Variable Phase PWM
  - Constant Off-time PWM
  - Current Reset PWM
  - Current-Limit PWM
  - Independent Time Base PWM
- On-the-fly changes to:
  - PWM frequency
  - PWM duty cycle
  - PWM phase shift
- Output override control
- Independent Current-Limit and Fault inputs
- Special Event Comparator for scheduling other peripheral events
- Comparator for each PWM generator for triggering ADC conversions

Figure 30-1 shows a simple block diagram representation of the PWM module. Figure 30-2 illustrates how the module hardware is partitioned for each PWM output pair for the Complementary PWM Output mode. Each functional unit of the PWM module is discussed in subsequent sections.

The Power Supply PWM module contains up to four PWM generators and has up to eight output pins. For complementary outputs, the following eight I/O pins are grouped into High/Low (H/L) pairs: PWM1H, PWM1L, PWM2H, PWM2L, PWM3H, PWM3L, PWM4H and PWM4L.

Figure 30-1: Simplified Conceptual Block Diagram of the Power Supply PWM

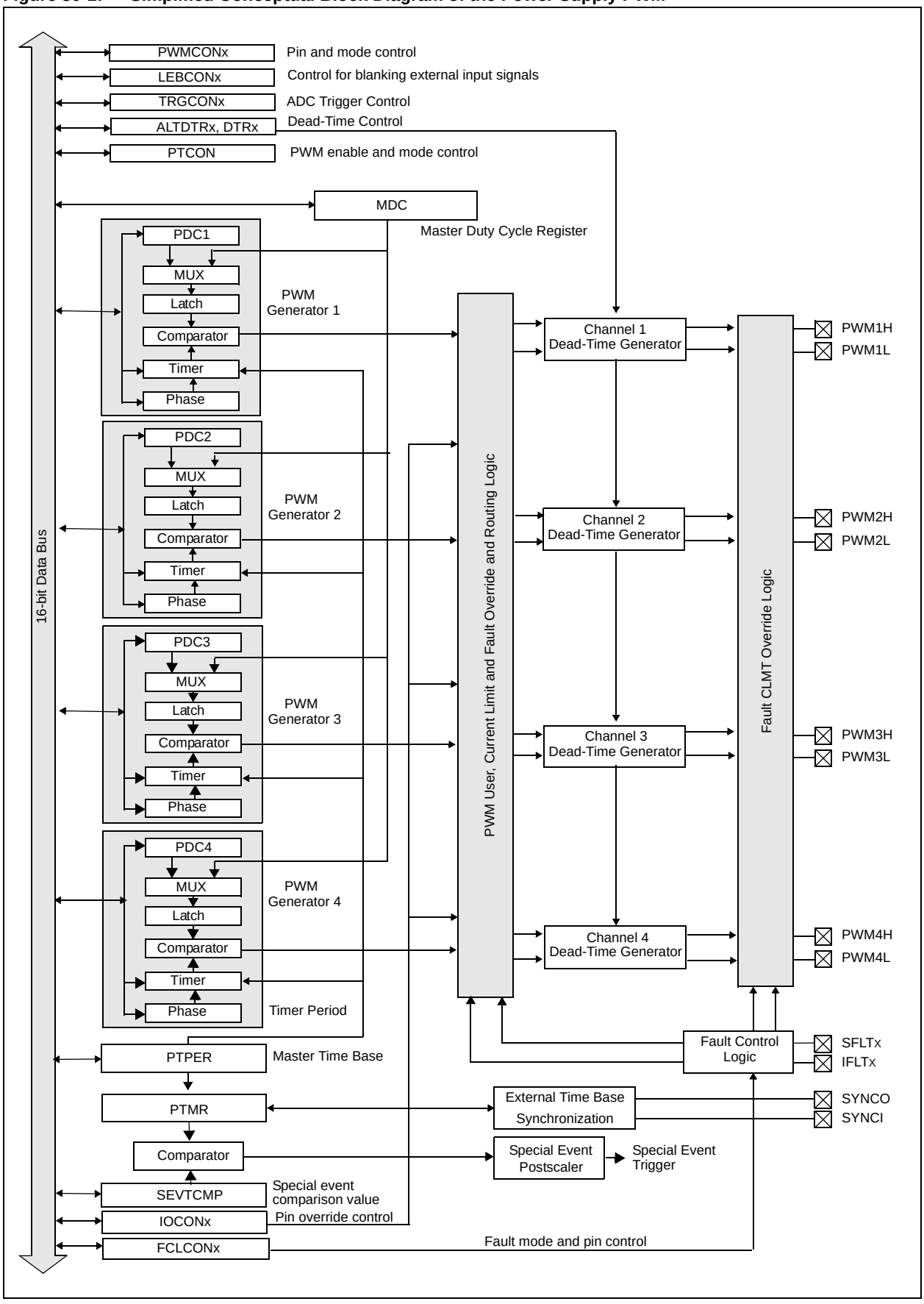
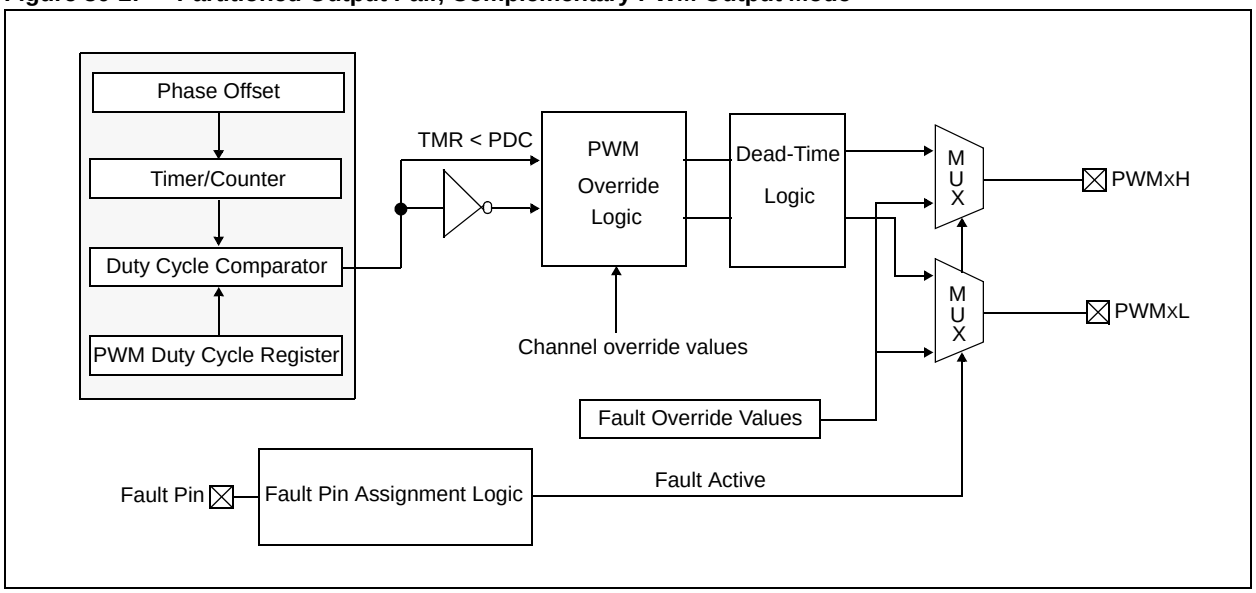


Figure 30-2: Partitioned Output Pair, Complementary PWM Output Mode



30.3 MODULE DESCRIPTION

The Power Supply PWM module is designed for applications that require the following features:

- High duty cycle resolution at high PWM frequencies
- The ability to drive standard Push-Pull or Half-Bridge converters
- The ability to create multi-phase PWM outputs

Push-Pull and Half-Bridge converters are two common, medium-power topologies that require the PWM signal to be switched between alternate pins, as provided by the Push-Pull PWM mode.

Phase-Shifted PWM describes the example where each PWM generator provides outputs, but the phase relationship between the generator outputs can be specified and changed.

The Multi-Phase PWM is often used to improve DC-DC converter load transient response and reduce the size of output filter capacitors and inductors. Multiple DC-DC converters are often operated in parallel, but are phase shifted in time. A single PWM output operating at 250 kHz has a period of 4 μs; however, an array of four PWM channels, staggered by 1 μs each, yields an effective switching frequency of 1 MHz. Multi-phase PWM applications typically use a fixed-phase relationship.

The Variable-Phase PWM is useful in Zero Voltage Transition (ZVT) power converters. Here the PWM duty cycle is always 50%, and the power flow is controlled by varying the relative phase shift between the two PWM generators.

Note:	The Phase-Locked Loop (PLL) must be enabled for the Power Supply PWM module to function. This is achieved by setting the FNOSC<1:0> bits in the FOSCSEL configuration register.

30.4 CONTROL REGISTERS

The following registers control the operation of the Power Supply PWM module:

- **PTCON: PWM Time Base Control Register**

  This register is used to turn the PWM module On or Off, to configure the Special Event Trigger, and to specify the Synchronization settings.

- **PTPER: Primary Time Base Register**

  The PWM time base value is written into this register, which determines the PWM operating frequency.

- **SEVTCMP: PWM Special Event Compare Register**

  This register contains the compare value to which the Special Event Trigger is generated.

- **MDC: PWM Master Duty Cycle Register**

  This register provides the duty cycle value when a PWM generator is configured to use the master duty cycle.

- **PWMCONx: PWM Control Register**

  This register controls the fault and current-limit interrupts, and also the dead-time, duty cycle and time base modes.

- **PDCx: PWM Generator Duty Cycle Register**

  The value in this register provides the duty cycle value for the PWMxH/PWMxL outputs when the master duty cycle is not selected.

- **PHASEx: PWM Phase-Shift Register**

  The value in this register provides the phase shift for the PWMxH/PWMxL outputs. When Independent Time Base PWM mode is selected, the PWMxH/PWMxL period value is provided.

- **DTRx: PWM Dead-Time Register**

  The value in this register provides the dead-time for the PWMxH output if positive dead-time is selected and PWMxL output if negative dead time is selected.

- **ALTDTRx: PWM Alternate Dead-Time Register**

  The value in this register provides the dead-time for the PWMxL output, if positive dead-time is selected, and PWMxH output, if negative dead-time is selected.

- **TRGCONx: PWM Trigger Control Register**

  This register provides the PWMx trigger postscaler and also the number of PWM cycles to skip before generating the first trigger.

- **IOCONx: PWM I/O Control Register**

  This register controls the PWMxH/PWMxL outputs, the PWM mode and PWM output override options.

- **FCLCONx: PWM Fault Current-Limit Control Register**

  This register configures the Fault and Current-Limit features.

- **TRIGx: PWM Trigger Compare Value Register**

  This register contains the compare value for generating the PWMx trigger. This value is compared with the selected PWMx time base.

- **LEBCONx: Leading-Edge Blanking Control Register**

  This register controls the Leading-Edge Blanking (LEB) feature of the PWM module.

Register 30-1: PTCON: PWM Time Base Control Register

Upper Byte:							
R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTEN	—	PTSIDL	SESTAT	SEIEN	EIPU	SYNCPOL	SYNCOEN
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SYNCEN	SYNCSRC<2:0>			SEVTPS<3:0>			
bit 7							bit 0

bit 15 **PTEN:** PWM Module Enable bit
1 = PWM module is enabled
0 = PWM module is disabled

bit 14 **Unimplemented:** Read as '0'

bit 13 **PTSIDL:** PWM Time Base Stop in Idle Mode bit
1 = PWM time base halts in CPU Idle mode
0 = PWM time base runs in CPU Idle mode

bit 12 **SESTAT:** Special Event Interrupt Status bit
1 = Special Event Interrupt is pending
0 = Special Event Interrupt is not pending

bit 11 **SEIEN:** Special Event Interrupt Enable bit
1 = Special Event Interrupt is enabled
0 = Special Event Interrupt is disabled

bit 10 **EIPU:** Enable Immediate Period Updates bit
1 = Active Period register is updated immediately
0 = Active Period register updates occur on PWM cycle boundaries

bit 9 **SYNCPOL:** Synchronize Input Polarity bit
1 = SYNCIN polarity is inverted (active-low)
0 = SYNCIN is active-high

bit 8 **SYNCOEN:** Primary Time Base Sync Enable bit
1 = SYNCO output is enabled
0 = SYNCO output is disabled

bit 7 **SYNCEN:** External Time Base Synchronization Enable bit
1 = External synchronization of primary time base is enabled
0 = External synchronization of primary time base is disabled

bit 6-4 **SYNCSRC<2:0>:** Sync Source Selection bits
111 = Reserved
•
•
•
001 = Reserved
000 = SYNCI

bit 3-0 **SEVTPS<3:0>:** PWM Special Event Trigger Output Postscale Select bits
1111 = 1:16 Postscale
•
•
•
0001 = 1:2 Postscale
0000 = 1:1 Postscale

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Register 30-2: PTPER: Primary Time Base Register

Upper Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTPER <15:8>							
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
PTPER <7:3>					—	—	—
bit 7							bit 0

bit 15-3 **PTPER<15:3>:** Primary Time Base (PTMR) Period Value bits

bit 2-0 **Unimplemented:** Read as '0'

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Register 30-3: SEVTCMP: PWM Special Event Compare Register

Upper Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SEVTCMP <15:8>							
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
SEVTCMP <7:3>					—	—	—
bit 7							bit 0

bit 15-3 **SEVTCMP<15:3>:** Special Event Compare Count Value bits

bit 2-0 **Unimplemented:** Read as '0'

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Register 30-4: MDC: PWM Master Duty Cycle Register

Upper Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MDC<15:8>							
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MDC<7:0>							
bit 7							bit 0

bit 15-0 **MDC<15:0>:** Master PWM Duty Cycle Value bits**(1)**

Note 1: The minimum value for this register is 0x0008 and the maximum value is 0xFFEF.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

Register 30-5: PWMCONx: PWM Control Register

Upper Byte:							
HS/HC-0	HS/HC-0	HS/HC-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FLTSTAT	CLSTAT	TRGSTAT	FLTIEN	CLIEN	TRGIEN	ITB	MDCS
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
DTC<1:0>		—	—	—	—	XPRES	IUE
bit 7							bit 0

bit 15 **FLTSTAT:** Fault Interrupt Status
1 = Fault Interrupt is pending
0 = No Fault Interrupt is pending
This bit is cleared by setting FLTIEN = 0.

Note: Software must clear the interrupt status here, and clear the corresponding IFS bit in the Interrupt Controller.

bit 14 **CLSTAT:** Current-Limit Interrupt Status bit
1 = Current-limit interrupt is pending
0 = No current-limit interrupt is pending
This bit is cleared by setting CLIEN = 0.

Note: Software must clear the interrupt status here, and clear the corresponding IFS bit in the Interrupt Controller.

bit 13 **TRGSTAT:** Trigger Interrupt Status bit
1 = Trigger interrupt is pending
0 = No trigger interrupt is pending
This bit is cleared by setting TRGIEN = 0.

bit 12 **FLTIEN:** Fault Interrupt Enable bit
1 = Fault interrupt enabled
0 = Fault interrupt disabled and FLTSTAT bit is cleared

bit 11 **CLIEN:** Current-Limit Interrupt Enable bit
1 = Current-limit interrupt enabled
0 = Current-limit interrupt disabled and CLSTAT bit is cleared

bit 10 **TRGIEN:** Trigger Interrupt Enable bit
1 = A trigger event generates an interrupt request
0 = Trigger event interrupts are disabled and TRGSTAT bit is cleared

bit 9 **ITB:** Independent Time Base Mode bit
1 = PHASEx register provides time base period for this PWM generator
0 = Primary time base provides timing for this PWM generator

bit 8 **MDCS:** Master Duty Cycle Register Select bit
1 = MDC register provides duty cycle information for this PWM generator
0 = PDCx register provides duty cycle information for this PWM generator

bit 7-6 **DTC<1:0>:** Dead-Time Control bits
00 = Positive dead-time actively applied for all output modes
01 = Negative dead-time actively applied for all output modes
10 = Dead-Time function is disabled
11 = Reserved

bit 5-2 **Unimplemented:** Read as '0'

Register 30-5: PWMCONx: PWM Control Register (Continued)

bit 1 **XPRES:** External PWM Reset Control bit

1 = Current-limit source resets time base for this PWM generator if it is in independent Time Base mode
0 = External pins do not affect PWM time base

bit 0 **IUE:** Immediate Update Enable bit

1 = Updates to the active PDC registers are immediate
0 = Updates to the active PDC registers are synchronized to the PWM time base

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown
HS = Hardware set	HC = Hardware clear		

Register 30-6: PDCx: PWM Generator Duty Cycle Register

Upper Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PDCx<15:8>(1)							
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PDCx<7:0>(1)							
bit 7							bit 0

bit 15-0 **PDCx<15:0>:** PWM Generator x Duty Cycle Value bits(1)

Note 1: The minimum value for this register is 0x0008 and the maximum value is 0xFFEF.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Register 30-7: PHASEx: PWM Phase-Shift Register

Upper Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PHASEx<15:8>							
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
PHASEx<7:2>						—	—
bit 7							bit 0

bit 15-2 **PHASEx<15:2>:** PWM Phase-Shift Value or Independent Time Base Period for this PWM Generator bits
If used as an independent time base, bit 3 and bit 2 are not used.

bit 1-0 **Unimplemented:** Read as '0'

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Register 30-8: DTRx: PWM Dead-Time Register

Upper Byte:							
U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DTRx<13:8>					
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
DTRx<7:2>						—	—
bit 7							bit 0

bit 15-14 **Unimplemented:** Read as '0'

bit 13-2 **DTRx<13:2>:** Unsigned 12-bit Dead-Time Value bits for PWMx Dead-Time Unit

bit 1-0 **Unimplemented:** Read as '0'

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Register 30-9: ALTDTRx: PWM Alternate Dead-Time Register

Upper Byte:							
U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	ALTDTRx<13:8>					
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
ALTDTRx<7:2>						—	—
bit 7							bit 0

bit 15-14 **Unimplemented:** Read as '0'

bit 13-2 **ALTDTRx<13:2>:** Unsigned 12-bit Dead-Time Value bits for PWMx Dead-Time Unit

bit 1-0 **Unimplemented:** Read as '0'

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Register 30-10: TRGCONx: PWM Trigger Control Register

Upper Byte:							
R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TRGDIV<2:0>			—	—	—	—	—
bit 15							bit 8

Lower Byte:							
U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	TRGSTRT<5:0>					
bit 7							bit 0

bit 15-13 **TRGDIV<2:0>:** Trigger Output Divider

000 = Trigger output for every trigger event
001 = Trigger output for every 2nd trigger event
010 = Trigger output for every 3rd trigger event
011 = Trigger output for every 4th trigger event
100 = Trigger output for every 5th trigger event
101 = Trigger output for every 6th trigger event
110 = Trigger output for every 7th trigger event
111 = Trigger output for every 8th trigger event

bit 12-6 **Unimplemented:** Read as '0'

bit 5-0 **TRGSTRT<5:0>:** Trigger Postscaler Start Enable Select bits

This value specifies the ROLL counter value needed for a match that will then enable the trigger postscaler logic to begin counting trigger events.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Register 30-11: IOCONx: PWM I/O Control Register

Upper Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PENH	PENL	POLH	POLL	PMOD<1:0>		OVRENH	OVRENL
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
OVRDAT<1:0>		FLTDAT<1:0>		CLDAT<1:0>		—	OSYNC
bit 7							bit 0

bit 15 **PENH:** PWMxH Output Pin Ownership bit
1 = PWM module controls PWMxH pin
0 = GPIO module controls PWMxH pin

bit 14 **PENL:** PWMxL Output Pin Ownership bit
1 = PWM module controls PWMxL pin
0 = GPIO module controls PWMxL pin

bit 13 **POLH:** PWMxH Output Pin Polarity bit
1 = PWMxH pin is active-low
0 = PWMxH pin is active-high

bit 12 **POLL:** PWMxL Output Pin Polarity bit
1 = PWMxL pin is active-low
0 = PWMxL pin is active-high

bit 11-10 **PMOD<1:0>:** PWMxH/PWMxL I/O Pin Mode bits
00 = PWM I/O pin pair is in the Complementary PWM Output mode
01 = PWM I/O pin pair is in the Independent Output mode
10 = PWM I/O pin pair is in the Push-Pull Output mode
11 = Reserved

bit 9 **OVRENH:** Override Enable for PWMxH Pin bit
1 = OVRDAT<1> provides data for output on PWMxH pin
0 = PWM generator provides data for PWMxH pin

bit 8 **OVRENL:** Override Enable for PWMxL Pin bit
1 = OVRDAT<0> provides data for output on PWMxL pin
0 = PWM generator provides data for PWMxL pin

bit 7-6 **OVRDAT<1:0>:** Data for PWMxH/PWMxL Pins if Override is Enabled bits
If OVERENH = 1, then OVRDAT<1> provides data for PWMxH
If OVERENL = 1, then OVRDAT<0> provides data for PWMxL

bit 5-4 **FLTDAT<1:0>:** Data for PWMxH/PWMxL Pins if FLTMOD is Enabled bits
If fault is active, then FLTDAT<1> provides data for PWMxH
If fault is active, then FLTDAT<0> provides data for PWMxL

bit 3-2 **CLDAT<1:0>:** Data for PWMxH/PWMxL Pins if CLMODE is Enabled bits
If current limit is active, then CLDAT<1> provides data for PWMxH
If current limit is active, then CLDAT<0> provides data for PWMxL

bit 1 **Unimplemented:** Read as '0'

bit 0 **OSYNC:** Output Override Synchronization bit
1 = Output overrides via the OVRDAT<1:0> bits are synchronized to the PWM time base
0 = Output overrides via the OVRDAT<1:0> bits occur on next clock boundary

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Register 30-12: FCLCONx: PWM Fault Current-Limit Control Register

Upper Byte:							
U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	CLSRC<3:0>				CLPOL
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CLMODE	FLTSRC<3:0>				FLTPOL	FLTMOD<1:0>	
bit 7							bit 0

bit 15-13 **Unimplemented:** Read as '0'

bit 12-9 **CLSRC<3:0>:** Current-Limit Control Signal Source Select for PWM #x Generator bits

0000 = Analog Comparator 1
0001 = Analog Comparator 2
0010 = Analog Comparator 3
0011 = Analog Comparator 4

0100 = Reserved
0101 = Reserved
0110 = Reserved
0111 = Reserved

1000 = Shared Fault 1 (SFLT1)
1001 = Shared Fault 2 (SFLT2)
1010 = Shared Fault 3 (SFLT3)
1011 = Shared Fault 4 (SFLT4)

1100 = Reserved
1101 = Independent Fault 2 (IFLT2)
1110 = Reserved
1111 = Independent Fault 4 (IFLT4)

bit 8 **CLPOL:** Current-Limit Polarity for PWMxH/PWMxL Generator x bit
1 = The selected current-limit source is active-low
0 = The selected current-limit source is active-high

bit 7 **CLMODE:** Current-Limit Mode Enable for PWMxH/PWMxL Generator x bit
1 = Current-Limit mode is enabled
0 = Current-Limit mode is disabled

Register 30-12: FCLCONx: PWM Fault Current-Limit Control Register (Continued)

bit 6-3 **FLTSRC<3:0>:** Fault Control Signal Source Select for PWM Generator x bits

0000 = Analog Comparator 1
0001 = Analog Comparator 2
0010 = Analog Comparator 3
0011 = Analog Comparator 4

0100 = Reserved
0101 = Reserved
0110 = Reserved
0111 = Reserved

1000 = Shared Fault 1 (SFLT1)
1001 = Shared Fault 2 (SFLT2)
1020 = Shared Fault 3 (SFLT3)
1011 = Shared Fault 4 (SFLT4)

1100 = Reserved
1101 = Independent Fault 2 (IFLT2)
1110 = Reserved
1111 = Independent Fault 4 (IFLT4)

bit 2 **FLTPOL:** Fault Polarity for PWM Generator x bit

1 = The selected Fault source is active-low
0 = The selected Fault source is active-high

bit 1-0 **FLTMOD<1:0>:** Fault Mode for PWM Generator x bits

00 = The selected Fault source forces PWMxH/PWMxL pins to FLTDAT values (latched condition)
01 = The selected Fault source forces PWMxH/PWMxL pins to FLTDAT values (cycle)
10 = Reserved
11 = Fault input is disabled

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

Register 30-13: TRIGx: PWM Trigger Compare Value Register

Upper Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TRGCMP<15:8>							
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
TRGCMP<7:3>					—	—	—
bit 7							bit 0

bit 15-3 **TRGCMP<15:3>:** Trigger Control Value bits(1)
Register contains the compare value for the PWMx time base for generating a trigger to the ADC module for initiating a sample and conversion process or generating a trigger interrupt.

bit 2-0 **Unimplemented:** Read as '0'

Note 1: The minimum usable value for this register is 0x0008. A value of 0x0000 does not produce a trigger. If the TRIGx value is being calculated based on the duty cycle value, you must ensure that a minimum TRIGx value is written into the register at all times.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Register 30-14: LEBCONx: Leading-Edge Blanking Control Register

Upper Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PHR	PHF	PLR	PLF	FLTLEBEN	CLLEBEN	LEB<9:8>	
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
LEB<7:3>					—	—	—
bit 7							bit 0

bit 15 **PHR:** PWMxH Rising Edge Trigger Enable bit
1 = Rising edge of PWMxH will trigger Leading-Edge Blanking counter
0 = Leading-Edge Blanking ignores rising edge of PWMxH

bit 14 **PHF:** PWMxH Falling Edge Trigger Enable bit
1 = Falling edge of PWMxH will trigger Leading-Edge Blanking counter
0 = Leading-Edge Blanking ignores falling edge of PWMxH

bit 13 **PLR:** PWMxL Rising Edge Trigger Enable bit
1 = Rising edge of PWMxL will trigger Leading-Edge Blanking counter
0 = Leading-Edge Blanking ignores rising edge of PWMxL

bit 12 **PLF:** PWMxL Falling Edge Trigger Enable bit
1 = Falling edge of PWMxL will trigger Leading-Edge Blanking counter
0 = Leading-Edge Blanking ignores falling edge of PWMxL

bit 11 **FLTLEBEN:** Fault Input Leading-Edge Blanking Enable bit
1 = Leading-Edge Blanking is applied to selected Fault Input
0 = Leading-Edge Blanking is not applied to selected Fault Input

bit 10 **CLLEBEN:** Current-Limit Leading-Edge Blanking Enable bit
1 = Leading-Edge Blanking is applied to selected Current-Limit Input
0 = Leading-Edge Blanking is not applied to selected Current-Limit Input

bit 9-3 **LEB<6:0>:** Leading-Edge Blanking for Current-Limit and Fault Inputs bits
Value is 8 ns increments.

bit 2-0 **Unimplemented:** Read as '0'

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

30.5 MODULE FUNCTIONALITY

The Power Supply PWM module is a high-speed design that provides the capabilities that are not found in other PWM generators. The following PWM modes are supported by this module:

- Standard Edge-Aligned PWM mode
- Complementary PWM Output mode
- Push-Pull PWM mode
- Multi-Phase PWM mode
- Variable Phase PWM mode
- Cycle-by-Cycle Current-Limit PWM mode
- Constant Off-time PWM mode
- Current Reset PWM mode
- Independent Time Base PWM mode

Some of these modes can be used in combination with other modes. For example, PWM1 can be configured for Complementary PWM Output mode operation and Variable Phase mode operation simultaneously. Similarly, the Standard Edge-Aligned mode can be used in combination with the Independent Time Base PWM mode.

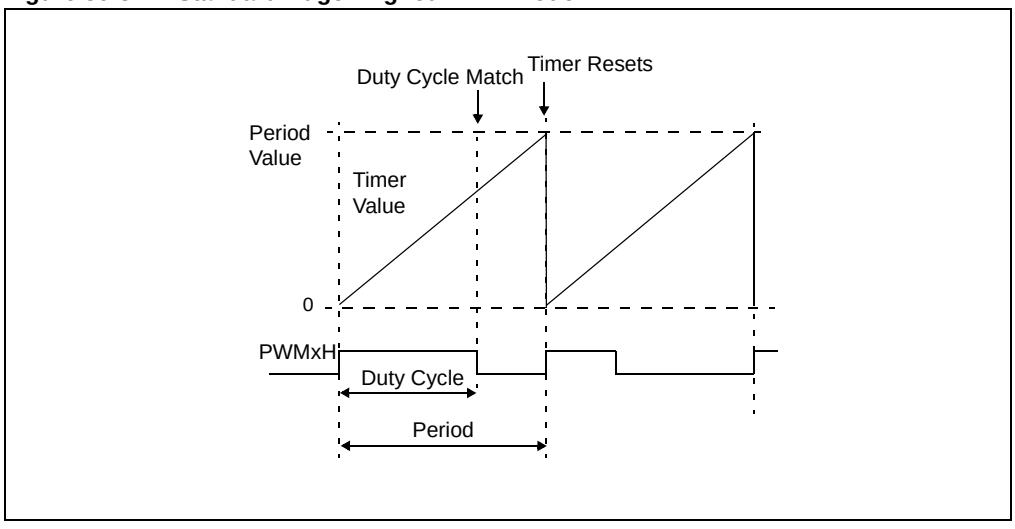
While there exist additional combinations that would be feasible to implement, there are certain combinations that are not possible to implement. As an example, it is not possible to use the same PWM generator to produce both Standard Edge-Aligned PWM mode and Complementary PWM Output mode.

Following are brief descriptions of all modes of operation supported by the dsPIC30F SMPS device family.

30.5.1 Standard Edge-Aligned PWM Mode

The Standard Edge-Aligned PWM mode shown in Figure 30-3 is the basic PWM mode used by many power converter topologies, including Buck, Boost and Forward. To create the edge-aligned PWM, a timer/counter circuit counts upward from zero to a specified maximum value for the period. Another register contains the value for the duty cycle, which is constantly compared to the timer (period) value. While the timer/counter value is less than or equal to the duty cycle value, the PWM output signal is asserted. When the timer value exceeds the duty cycle value, the PWM signal is deasserted. When the timer is greater than the period value, the timer is reset and the process repeats.

Figure 30-3: Standard Edge-Aligned PWM Mode

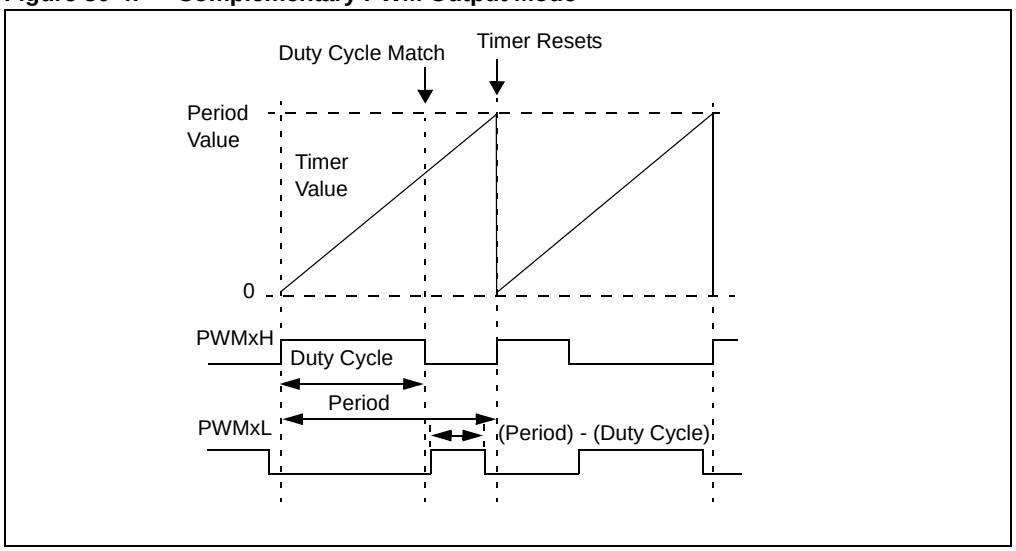


30.5.2 Complementary PWM Output Mode

The Complementary PWM Output mode shown in Figure 30-4 is generated in a manner similar to Standard Edge-Aligned PWM mode. This mode provides a second PWM output signal on the PWMxL pin that is the complement of the primary PWM signal (PWMxH). This mode is widely used in synchronous buck converters and resonant converters.

In the dsPIC30F SMPS device family, this mode is enabled by setting PMOD = (00)b in the IOCONx register. The Complementary PWM Output mode is shown in Figure 30-4.

Figure 30-4: Complementary PWM Output Mode

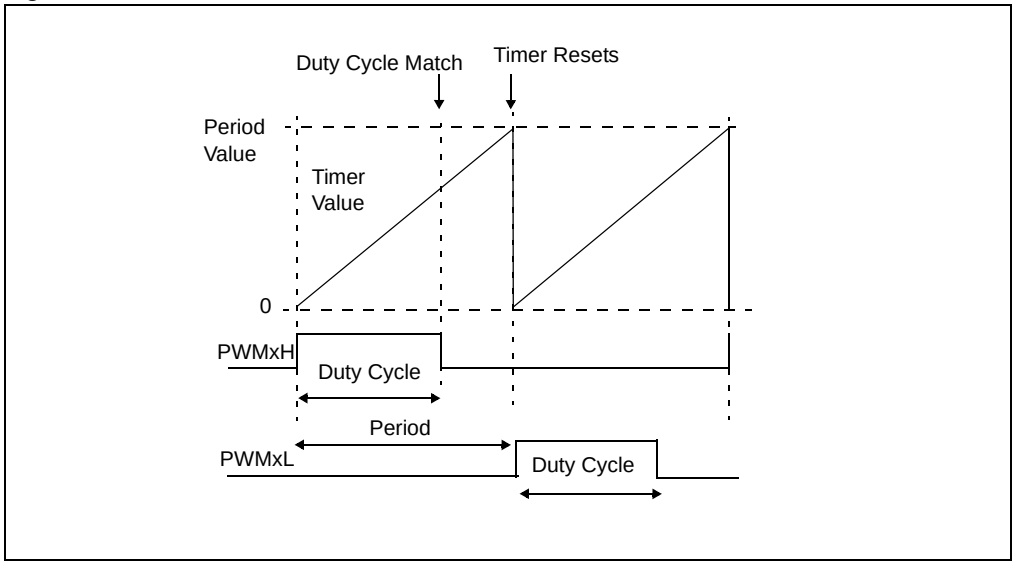


30.5.3 Push-Pull PWM Mode

The Push-Pull PWM mode shown in Figure 30-5 is a version of the Standard Edge-Aligned PWM mode where the active PWM signal is alternately output on one of the two PWM pins. Complementary PWM output is not available. This mode is useful in transformer-based power converters. Transformer-based circuits must avoid any direct currents that will cause their cores to saturate. The Push-Pull mode ensures that the duty cycle of the two phases is identical, thereby yielding a net DC bias of zero.

In the dsPIC30F SMPS device family, this mode is enabled by setting PMOD = (10)b in the IOCONx register.

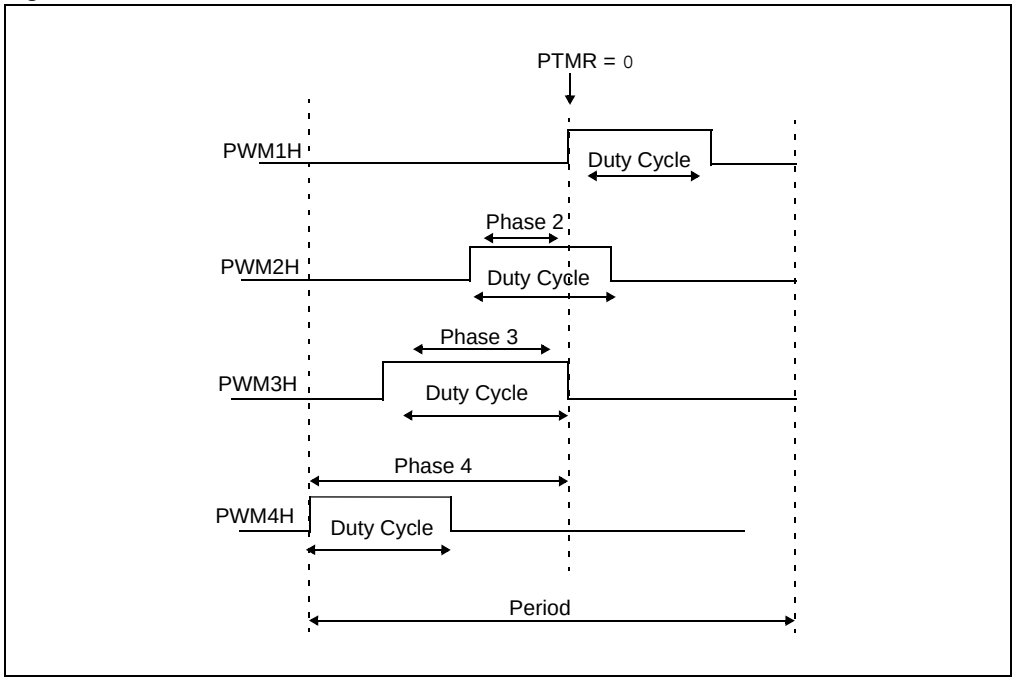
Figure 30-5: Push-Pull PWM Mode



30.5.4 Multi-Phase PWM Mode

The Multi-Phase PWM mode shown in Figure 30-6 uses phase-shift values in the PHASEx registers to shift the PWM outputs relative to the primary time base. As the phase-shift values are added to the primary time base, the phase-shifted outputs occur earlier than in a PWM channel that specifies zero phase shift. In Multi-Phase mode, the specified phase shift is fixed by the user-assigned application design.

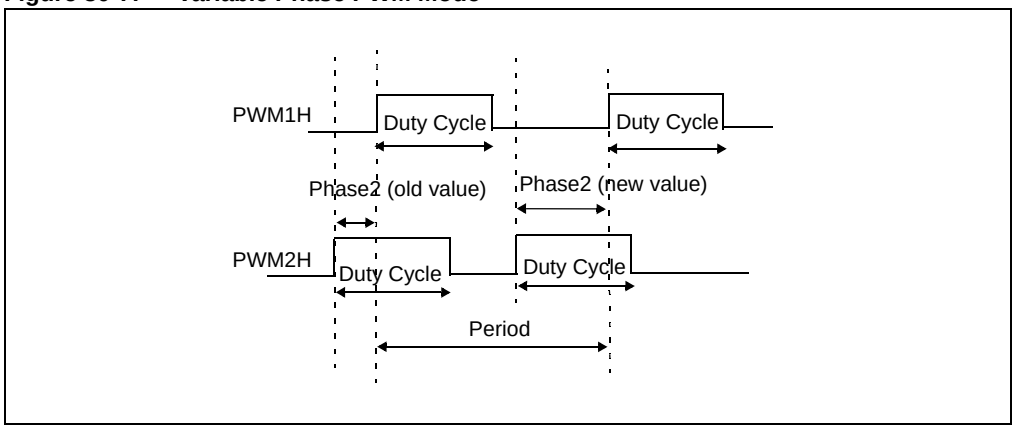
Figure 30-6: Multi-Phase PWM Mode



30.5.5 Variable Phase PWM Mode

Figure 30-7 shows the waveforms for Variable Phase PWM mode. Power-converter circuits constantly change the phase shift among PWM channels to control the flow of power, in contrast to most PWM circuits that vary the duty cycle of PWM signals to control power flow. Often in variable phase applications, the PWM duty cycle is maintained at 50%. The phase-shift value should be updated when the PWM signal is not asserted. Complementary PWM outputs are available in Variable Phase mode.

Figure 30-7: Variable Phase PWM Mode



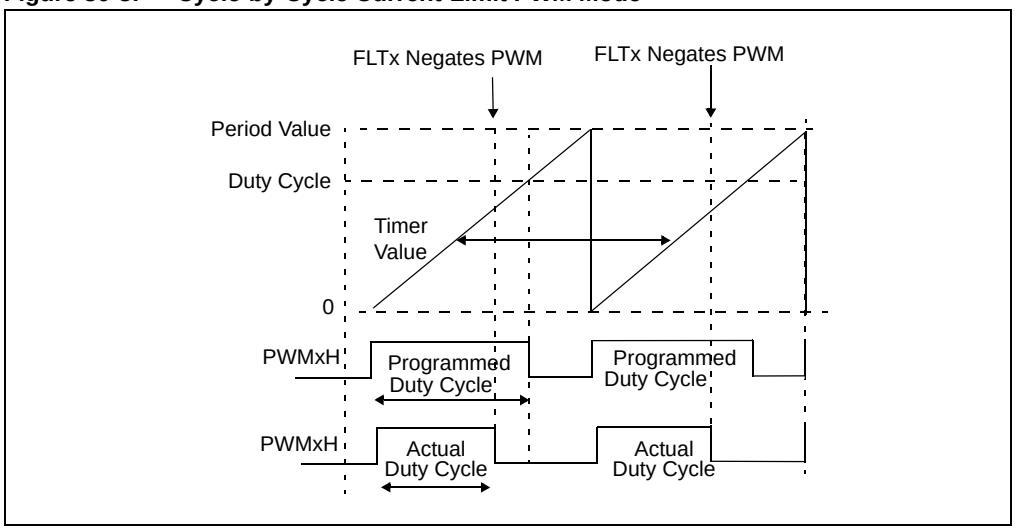
The difference between Multi-Phase PWM mode and Variable Phase PWM mode is that in the former there is a a phase difference between two separate PWM outputs. Conversely, in variable phase mode, the phase of a single PWM output is changed in every PWM cycle.

30.5.6 Cycle-by-Cycle Current-Limit PWM Mode

The Cycle-by-Cycle Current-Limit mode shown in Figure 30-8 truncates the asserted PWM signal when the current-limit signal is asserted. The PWM output values are specified by the Current-Limit override bits (CLDAT<1:0>) in the IOCONx register. The override output remains in effect until the beginning of the next PWM cycle. This mode is sometimes used in PFC circuits where the inductor current controls the PWM on time. This is a constant frequency PWM mode.

An analog comparator output, a shared fault pin (SFLTx) or an independent fault pin (IFLTx) can be used as the current-limit signal by configuring the Current-Limit Control Signal Source Select Generator (CLSRC<3:0>) bits in the PWM Fault Current-Limit Control (FCLCONx<12:9>) register. The current-limit signal can also be configured to be active-high or active-low.

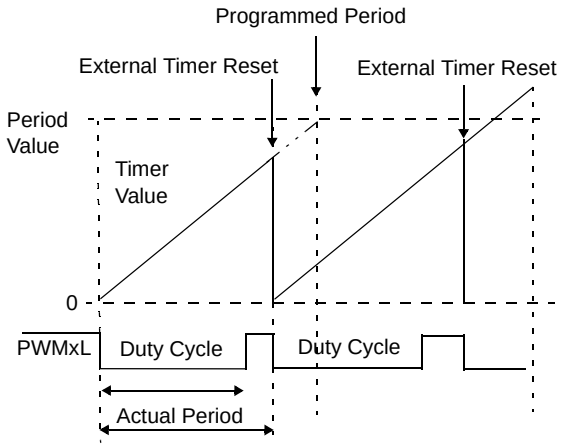
Figure 30-8: Cycle-by-Cycle Current-Limit PWM Mode



30.5.7 Constant Off-Time PWM Mode

The Constant Off-Time PWM mode shown in Figure 30-9 is a variable-frequency mode where the actual PWM period is less than or equal to the specified period value. The PWM time base is externally reset after the PWM signal duty cycle value is reached, and the PWM signal has been deasserted. This mode is implemented by enabling the Current Reset PWM mode and using the complementary PWM output.

Figure 30-9: Constant Off-Time PWM Mode



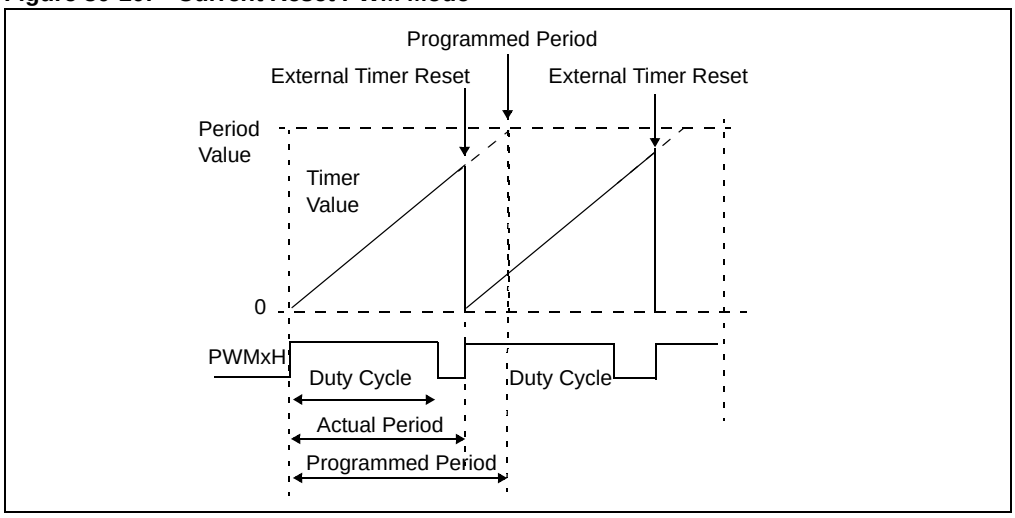
Note: Duty Cycle represents off time.

30.5.8 Current Reset PWM Mode

The Current Reset PWM mode shown in Figure 30-10 is a variable-frequency mode where the actual PWM period is less than or equal to the specified period value. The PWM time base is externally reset after the PWM signal duty cycle value is reached, and the PWM signal has been deasserted. Current Reset PWM mode is a constant on time PWM mode.

Typically, in the converter application, an energy storage inductor is charged with current while the PWM signal is asserted, and the inductor current is discharged by the load when the PWM signal is deasserted. In this application of Current Reset PWM mode, an external current measurement circuit determines when the inductor is discharged, and then generates a signal that the PWM module uses to reset the time base counter. In Current Reset PWM mode, complementary PWM outputs are available.

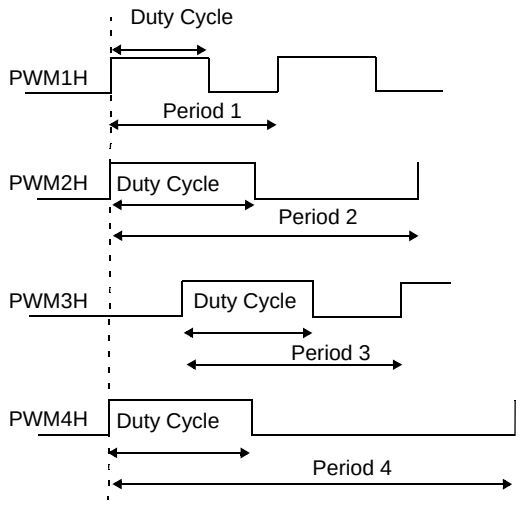
Figure 30-10: Current Reset PWM Mode



30.5.9 Independent Time Base PWM Mode

The Independent Time Base PWM mode shown in Figure 30-11 is often used when the dsPIC30F SMPS device is controlling different power converter sub-circuits such as the PFC circuit, which can use 100 kHz PWM, and the full-bridge forward converter section, which can use 250 kHz PWM.

Figure 30-11: Independent Time Base PWM Mode

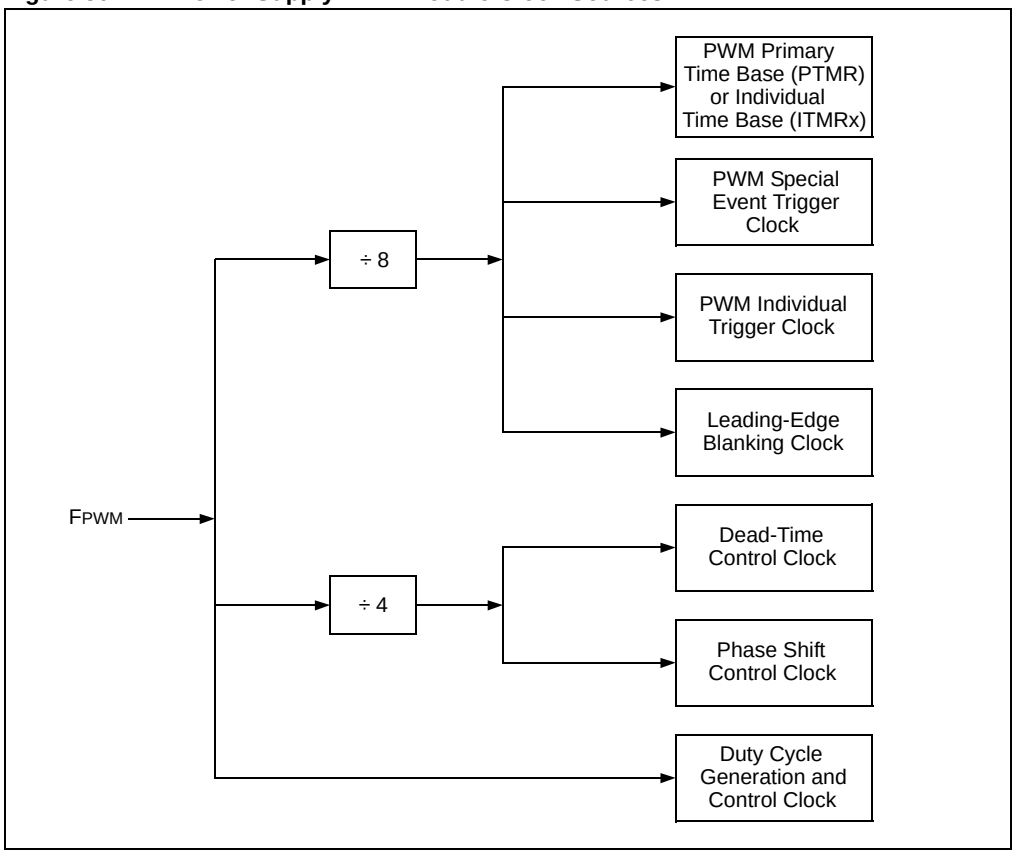


Note: With independent time bases, PWM signals are no longer phase related to each other.

30.6 CLOCK SOURCES

The Power Supply PWM module incorporates a number of clock frequencies for operating the various functional blocks of the module. The PLL output frequency (FPWM) from the Oscillator module is used to generate the clock signals used in the PWM module. Therefore, the PLL must be enabled to use the PWM module. Figure 30-12 shows the clock sources used in the PWM module and their relation with FPWM. Refer to **Section 29. "Oscillator" (DS70268)** in the "*dsPIC30F Family Reference Manual*" for more information on how FPWM is obtained.

Figure 30-12: Power Supply PWM Module Clock Sources

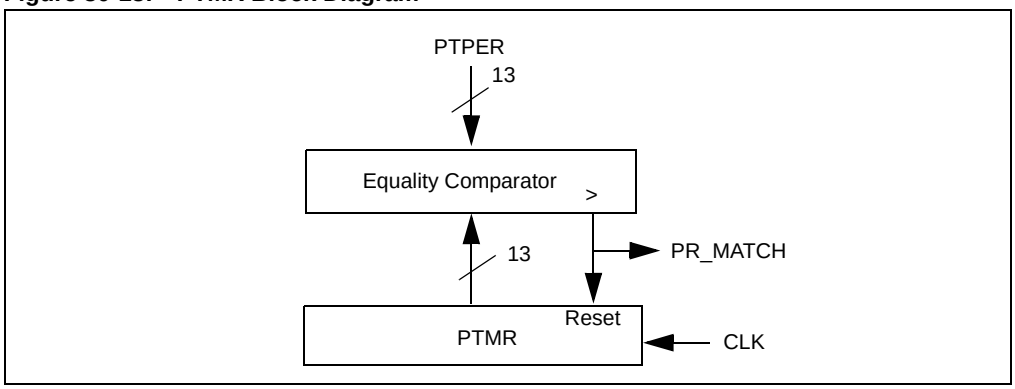


30.7 PRIMARY PWM TIME BASE

A Primary PWM Time Base (PTMR) counter exists for the entire PWM module. In addition, each PWM generator has an individual time base counter. The PTMR determines when the individual time base counters are to update their duty cycle and phase-shift registers. The master time base is also responsible for generating the Special Event Triggers and timer-based interrupts. Figure 30-13 shows a block diagram of the primary time base logic.

The primary time base can be reset by an external signal specified via the Sync Source Selection (SYNCSRC<2:0>) bits in the PWM Time Base Control (PTCON<6:4>) register. The External Reset feature is enabled using the External Time Base Synchronization Enable (SYNCEN) bit in the PTCON (PTCON<7>) register. The Primary Time Base Reset feature supports synchronization of the primary time base with another dsPIC30F SMPS device or other circuitry in the user-assigned application. The primary time base logic also provides an output signal (SYNCO) when a period match occurs.

Figure 30-13: PTMR Block Diagram



30.7.1 PTMR Synchronization

Because absolute synchronization is not possible, the user-assigned application should program the time base period of the secondary (Slave) device to be slightly longer than the primary device time base period to ensure that the two time bases will reset at the same time.

30.7.2 Primary PWM Time Base ROLL Counter

The primary time base has an additional 6-bit counter that counts the period matches of the primary time base. This counter is referred to as the ROLL counter and is not accessible for reading. Each PWM generator has six Trigger Postscaler Start Enable Select (TRGSTRT<5:0>) bits in the PWM Trigger Control (TRGCONx<5:0>) register that specify how many counts of the ROLL counter are allowed to pass before generating the first Analog-to-Digital (A/D) conversion trigger. The purpose of the ROLL counter and the TRGSTRT bits is to allow the user-assigned application to spread the system workload over a series of PWM cycles.

The TRGDIV<2:0> bits in the TRGCONx register act as a postscaler for the TRIGx register to generate ADC triggers. These bits specify how frequently the ADC trigger is generated. Once the TRGDIV<2:0> postscaler is enabled, the ROLL bits and TRGSTRT bits have no further effect until the PWM module is disabled and then re-enabled.

A typical application of the ROLL counter is a dsPIC® controlled multi-channel DC-DC converter. In this application, each individual converter operating on a PWM channel triggers an ADC and executes the control loop, ensuring that no two PWM channels generate an ADC trigger at the same time. This feature allows efficient utilization of the available CPU resources.

An additional use of the ROLL counter is to allow the internal FRC oscillator to be varied on a PWM cycle basis to reduce peak EMI emissions generated by switching transistors in the power conversion application.

The ROLL counter is cleared when the PWM module is disabled (PTEN = 0), and the TRIGx postscalers (TRGDIV) are also disabled. A new match of the ROLL counter and the TRGSTRT bit value must then take place to begin counting again.

30.8 INDIVIDUAL PWM TIME BASE(S)

Each PWM generator also has its own PWM time base. Figure 30-14 shows a block diagram for the individual time base circuits. With a time base per PWM generator, the PWM module can generate PWM outputs that are phase shifted relative to each other or totally independent of each other. The individual PWM timers (ITMRx) provide the time base values that are compared to the duty cycle registers to create the PWM signals. These individual time base counters can be initialized before or during an operation using the phase-shift registers. The primary (PTMR) and the individual (ITMRx) timers are not readable by the user-assigned application.

Normally, the Primary PWM Time Base (PTMR) provides synchronization control to the individual timers/counters so they count in lock-step unison.

If the PWM Phase-Shift feature is used, the PTMR provides the synchronization signal to each individual timer/counter that causes them to reinitialize with their individual phase-shift values.

If a PWM generator is operating in Independent Time Base mode, the individual timers/counters count upward until their count values match the value stored in their phase registers, then they reset and the cycle repeats.

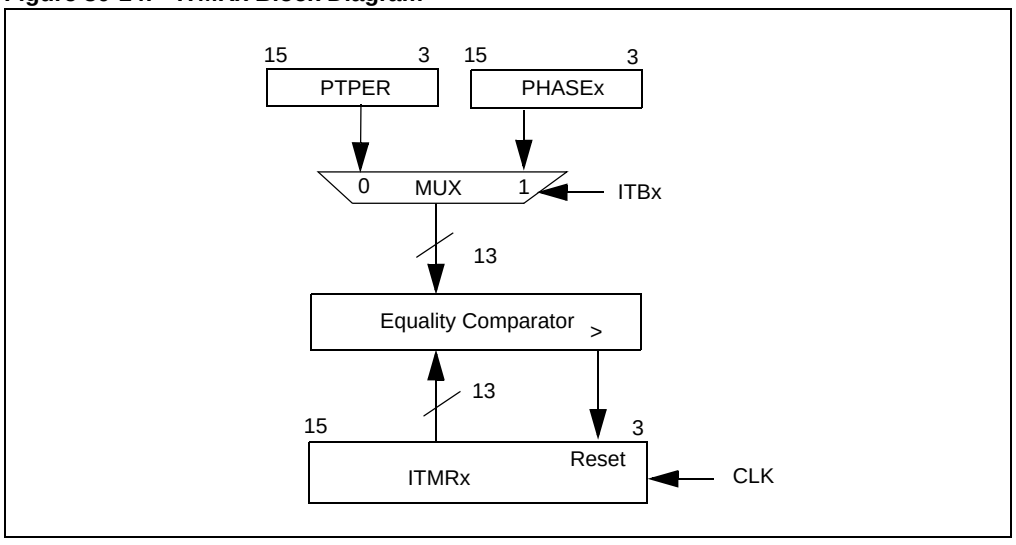
The primary time base and the individual time bases are implemented as 13-bit counters. The timers/counters are clocked at 120 MHz at 30 MIPS, which provides a frequency resolution of 8.4 ns.

All of the timers/counters are enabled or disabled by setting or clearing the PWM Module Enable (PTEN) bit in the PWM Time Base Control (PTCON<15>) register.

Note:	The timers are cleared when the PTEN bit is cleared in software.

If an individual time base is not used, the Primary Time Base (PTPER) register sets the counting period for PTMR. The user-assigned application must write a 13-bit value to PTPER<15:3>. When the value in PTMR<15:3> matches the value in PTPER<15:3>, the primary time base is reset to '0' and the individual time base counters are reinitialized to their phase values (except in Independent Time Base mode).

Figure 30-14: ITMRx Block Diagram



30.9 PWM PERIOD

The Primary Time Base Period (PTPER) register holds the 13-bit value that specifies the counting period for the primary PWM time base. The timer period can be updated at any time by the user-assigned application. When the PWM module operates in the Independent Time Base mode, the PHASEx register will provide the time base period. The PWM period can be determined from the formula shown in Equation 30-1.

Equation 30-1: PWM Period Formula

$$PTPER, PHASEx = \frac{ReferenceClock \times PLL \times 2}{PWM\ Switching\ Frequency}$$

where,

PLL = 32

Note: Refer to **Section 29. "Oscillator"** (DS70268) in the "*dsPIC30F Family Reference Manual*" for more information on FRC clock selection.

Example 30-1: PWM Period Calculation

$$PTPER = \frac{14.55\ MHz \times 32 \times 2}{100\ kHz} = 9312$$

where,

$PWM\ Frequency$ = 100 kHz

FRC = 14.55 MHz

PLL = 32

30.10 PWM FREQUENCY AND DUTY CYCLE RESOLUTION

The PWM duty cycle resolution is 1.05 ns per Least Significant Byte (LSB) at 30 MIPS. The PWM period resolution is 8.4 ns at 30 MIPS. Table 30-1 and Table 30-2 show the duty cycle resolution versus PWM frequencies for 30 MIPS and 20 MIPS execution speed. Equation 30-2 provides the PWM Duty Cycle Resolution formula.

Equation 30-2: PWM Duty Cycle Resolution

$$PWMDutyCycleResolution = \frac{(\log 2)[ReferenceClock \times PLL \times 2]}{PWM\ Switching\ Frequency}$$

where,

PLL = 32

Note: Refer to **Section 29. "Oscillator"** (DS70268) in the "*dsPIC30F Family Reference Manual*" for more information on FRC clock selection.

Table 30-1: Available PWM Frequencies and Resolutions @ 30 MIPS

MIPS	PWM Duty Cycle Resolution	PWM Frequency
30	16 bits	14.6 kHz
30	15 bits	29.3 kHz
30	14 bits	58.6 kHz
30	13 bits	117.2 kHz
30	12 bits	234.4 kHz
30	11 bits	468.9 kHz
30	10 bits	937.9 kHz
30	9 bits	1.87 MHz
30	8 bits	3.75 MHz

Table 30-2: Available PWM Frequencies and Resolutions @ 20 MIPS

MIPS	PWM Duty Cycle Resolution	PWM Frequency
20	16 bits	9.4 kHz
20	15 bits	18.9 kHz
20	14 bits	37.8 kHz
20	13 bits	75.7 kHz
20	12 bits	151.5 kHz
20	11 bits	303.1 kHz
20	10 bits	606.2 kHz
20	9 bits	1.21 MHz
20	8 bits	2.42 MHz

The reduction in available resolution for a given PWM frequency is due to the reduced clock rate and the fact that the LSB of duty cycle resolution is derived from a fixed-delay element. At operating frequencies below 30 MIPS, the contribution of the fixed-delay element to the output resolution becomes less than 1 LSB. For frequency resonant mode power conversion applications, it is desirable to know the available PWM frequency resolution. The available frequency resolution varies with the PWM frequency. The PWM time base clocks at 120 MHz at 30 MIPS. Equation 30-3 provides the frequency resolution versus PWM period:

Equation 30-3: Frequency Resolution Versus PWM Period

$$FrequencyResolution = \frac{120\ MHz}{Period}$$

where,

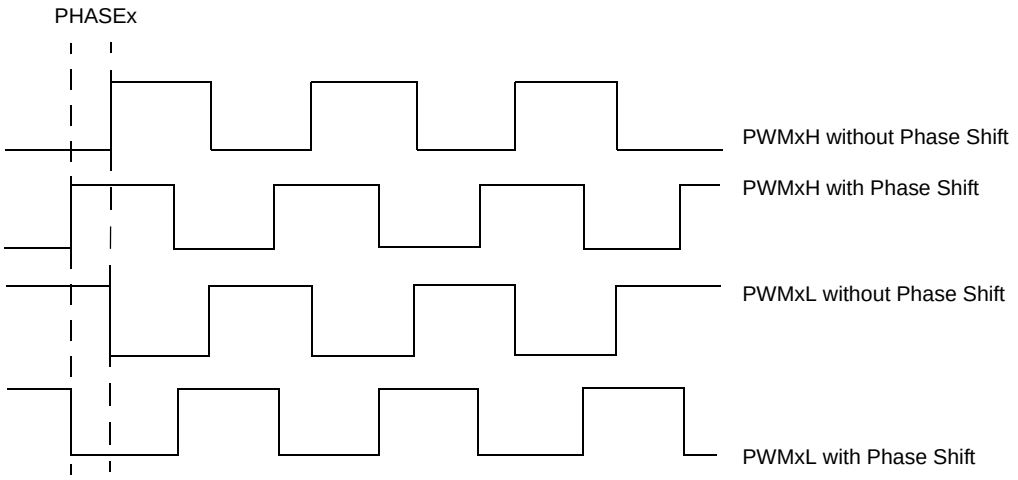
$Period$ = PTPER<15:3>

30.10.1 Phase Shift

Phase shift is the relative offset between PWMxH or PWMxL with respect to the master time base. In Independent Output mode, the PHASEx register determines the relative phase shift between PWMxH and the master time base. The contents of the PHASEx register are used as an initialization value for the PTMRx register.

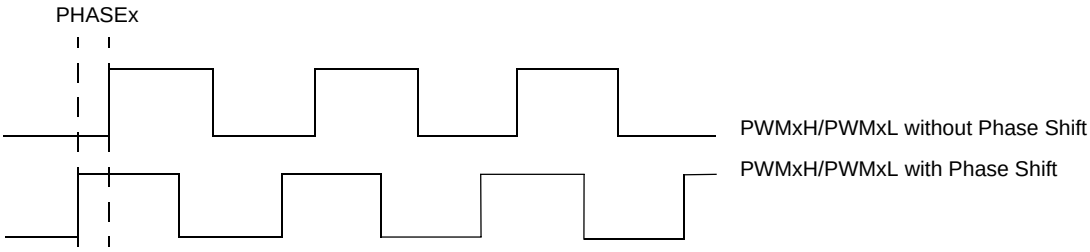
Figure 30-15 and Figure 30-16 provide example waveforms for phase shifting in Complementary mode and Independent Output mode, respectively.

Figure 30-15: Phase Shifting (Complementary Mode)



Note: In Complementary, Push-Pull, and Redundant Output modes, PHASEx controls the phase shift for PWMxH and PWMxL.

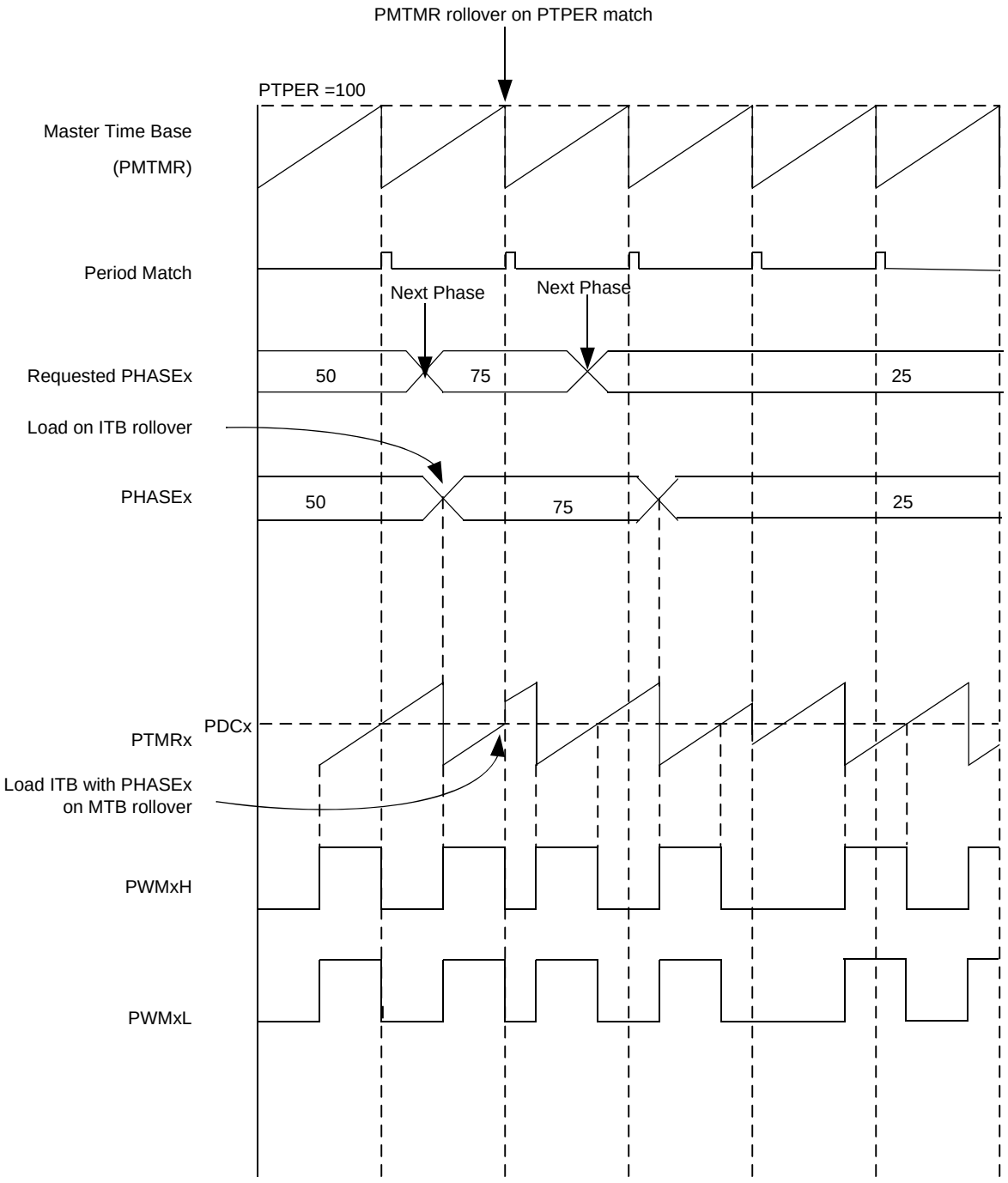
Figure 30-16: Phase Shifting (Independent Output Mode)



In addition, there is a shadow registers for the PHASEx registers that are updated whenever new values are written by the user-assigned application. These values are transferred from the shadow registers to the PHASEx registers on an independent time base reset. The actual application of these phase offsets on the PWM output will occur on a master time base reset.

Figure 30-17 shows the timing diagram that illustrates how these events are generated. The phase offset value can be any value between zero and the value in the PTPER register. Any PHASEx value greater than the PERIOD value will be treated as a value equal to the Period. It is not possible to create phase shifts greater than the Period.

Figure 30-17: Phase Shift Waveform



Note: Operation of High-Speed PWM module with independent time base is controlled by the master time base.

Example 30-2: PWM Phase Shift Initialization

```
/* Initialize phase shift value for the PWM output */
/* Phase shifts are initialized when operating in Master Time Base */

PHASEx = 100; // Phase shift value of 104 ns
```

30.11 PWM DUTY CYCLE COMPARISON UNITS

The PWM module has up to four PWM duty cycle generators. Up to five 16-bit special function registers are used to specify duty cycle values for the PWM module:

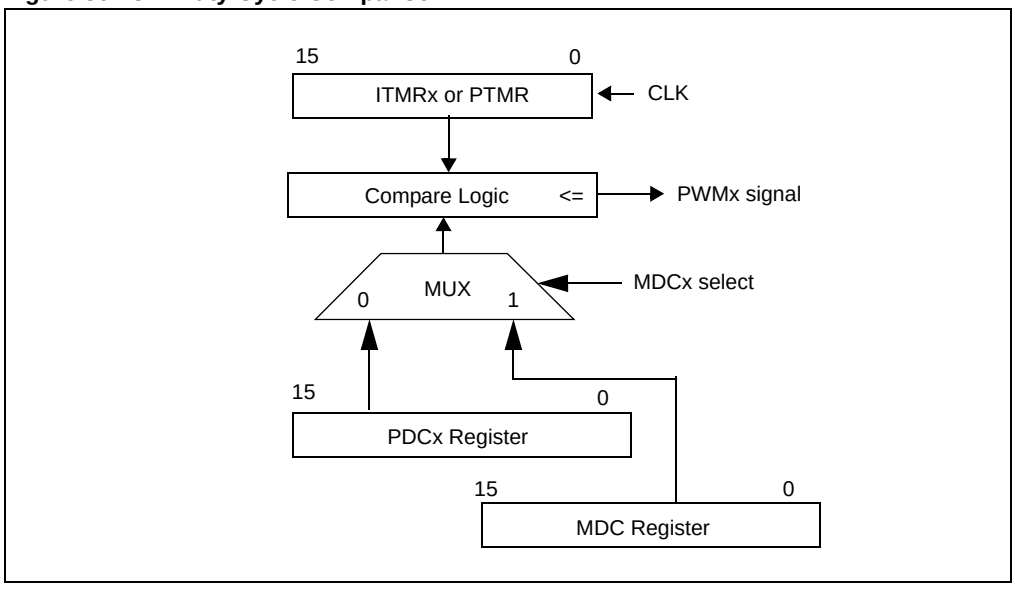
- Master Duty Cycle (MDC) register
- PWM Generator Duty Cycle (PDC1, PDC2, PDC3 and PDC4) registers

Each PWM generator has its own duty cycle register (PDCx), and there is a Master Duty Cycle (MDC) register. The MDC register can be used instead of individual duty cycle registers. The MDC register enables multiple PWM generators to share a common duty cycle register to reduce the CPU overhead required in updating multiple duty cycle registers. Multi-phase power converters are an application where the use of the MDC feature saves valuable processor time.

The value in each duty cycle register determines the amount of time that the PWM output is in an active state. The PWM time base counters are 13 bits wide and increment twice per instruction cycle. The PWM output is asserted when the timer/counter is less than or equal to the Most Significant 13 bits of the duty cycle register value. Each of the duty cycle registers allows a 16-bit duty cycle to be specified. The Least Significant 3 bits of the duty cycle registers are sent to additional logic for further adjustment of the PWM signal edge.

Figure 30-18 is a block diagram of a duty cycle comparison unit; however, the additional circuitry for resolving the Least Significant 3 bits is not shown.

Figure 30-18: Duty Cycle Comparison



The duty cycle values can be updated any time. The updated duty cycle values can be held until the next rollover of the primary time base before becoming active. Optionally, by setting the Immediate Update Enable (IUE) bit in the PWM Control (PWMCONx<0>) register, the updated duty cycle values can be used immediately.

30.12 COMPLEMENTARY PWM OUTPUTS

The Complementary PWM Output mode provides true and inverted PWM outputs on the pair of PWM output pins (PWMxH and PWMxL). The complementary PWM signal is generated by inverting the active PWM signal. Complementary outputs are normally available with all PWM modes except the Push-Pull PWM mode and Independent PWM Output mode.

30.13 INDEPENDENT PWM OUTPUTS

The Independent PWM Output mode simply replicates the active PWM output signal on both output pins (PWMxH and PWMxL) associated with a PWM generator.

Note:	The Independent PWM Output mode is different from the Independent Time Base mode. Here, each PWM generator has its own time base but can produce a complementary or push-pull output.

30.14 DUTY CYCLE LIMITS

The duty cycle generators are limited to the range of allowable values. A value of 0x0008 is the minimum duty cycle value that will produce an output pulse. This value represents 8.4 ns at 30 MIPS. This minimum range limitation is not a problem in real world applications because of the slew-rate limitation of the PWM output buffers, external FET drivers and power transistors. The application control loop requires larger duty cycle values to achieve minimum transistor ON times. The maximum duty cycle value is also limited to 0xFFEF.

Note:	The user-assigned application is responsible for limiting the duty cycle values to the allowable range of 0x0008 to 0xFFEF.

30.15 DEAD-TIME GENERATION

Dead-time refers to a programmable period of time (specified by the Dead-Time Register (DTRx) or the Alternate Dead-Time (ALTDTRx) register) that prevents a PWM output from being asserted until its complementary PWM signal has been deasserted for the specified time. Figure 30-19 shows the insertion of dead-time in a complementary pair of PWM outputs. Figure 30-20 shows the four Dead-Time Control Units, each having their own dead-time value. Dead-time generation can be provided when any of the PWM I/O pin pairs are operating in any output mode.

Many power-converter circuits require dead-time because the power transistors cannot switch instantaneously. To prevent current shoot-through, some amount of time must be provided between the turn-off event of one PWM output and the turn-on event of the other PWM output in a complementary pair.

The PWM module can also provide negative dead-time. Negative dead-time is the forced overlap of the PWMxH and PWMxL signals. Certain converter techniques require a limited amount of current shoot-through.

The Dead-Time feature can be disabled for each PWM generator. The Dead-Time functionality is controlled by the Dead-Time Control (DTC<1:0>) bits in the PWM Control (PWMCONx<7:6>) register.

Note:	If zero dead-time is required, the Dead-Time feature must be explicitly disabled in the DTC<1:0> bits in the PWMCONx register.

Figure 30-19: Dead-Time Insertion for Complementary PWM

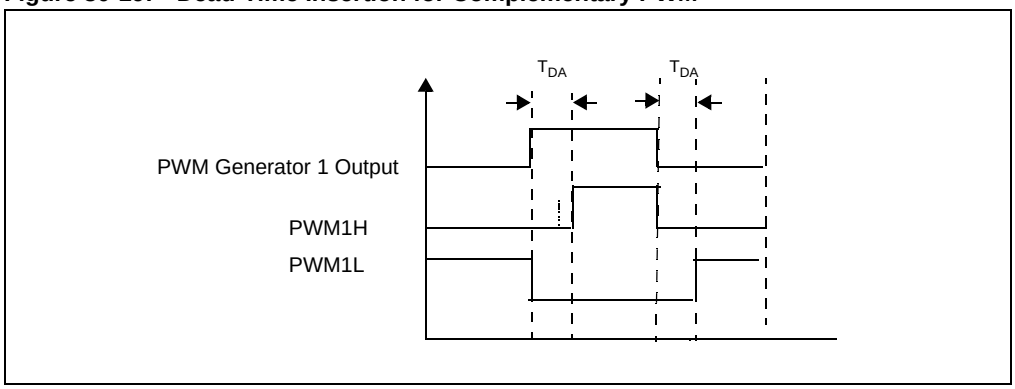
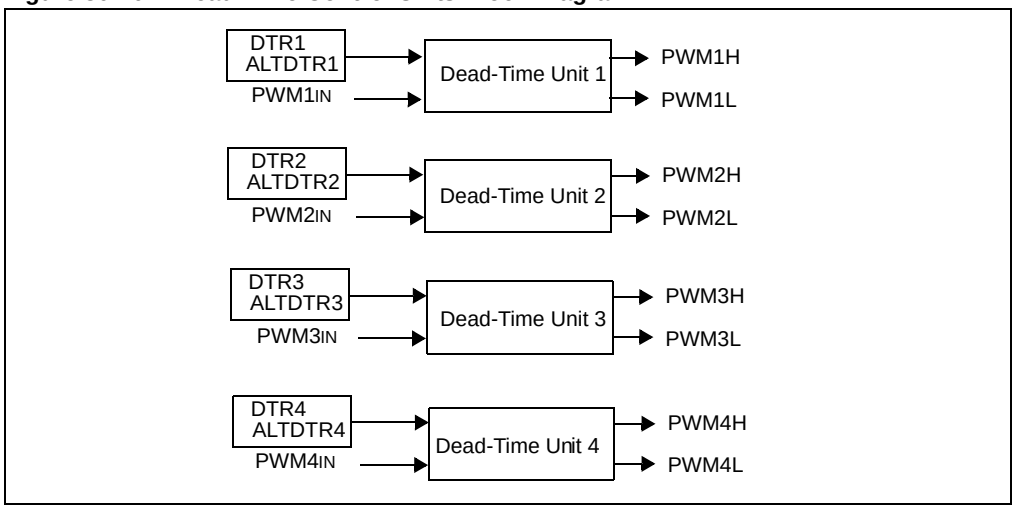


Figure 30-20: Dead-Time Control Units Block Diagram



30.15.1 Dead-Time Generators

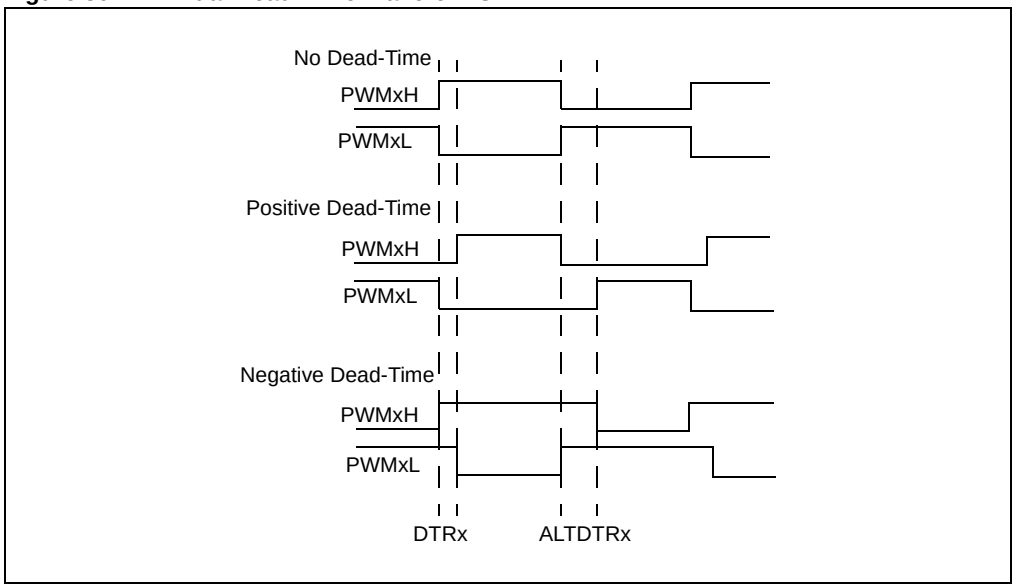
Each complementary output pair for the PWM module has 12-bit down counters to produce the dead-time insertion. Each dead-time unit has a rising and falling edge detector connected to the duty cycle comparison output.

Depending on whether the edge is rising or falling, one of the transitions on the complementary outputs is delayed until the associated timer counts down to zero. A timing diagram indicating the dead-time insertion for one pair of PWM outputs is shown in Figure 30-19.

30.15.2 Alternate Dead-Time Source

The alternate dead-time source refers to the dead-time specified by the ALTDTRx register that is applied to the complementary PWM output. Figure 30-21 shows a dual dead-time insertion using the ALTDTRx register.

Figure 30-21: Dual Dead-Time Waveforms



30.15.3 Dead-Time Ranges

The amount of dead-time provided by each dead-time unit is selected by specifying a 12-bit unsigned value in the DTRx registers. The 12-bit dead-time counters clock at four times the instruction execution rate. The Least Significant 1 bit of the dead-time value is processed by the Fine Adjust PWM module.

Example 30-3 shows example dead-time ranges as a function of the device operating frequency.

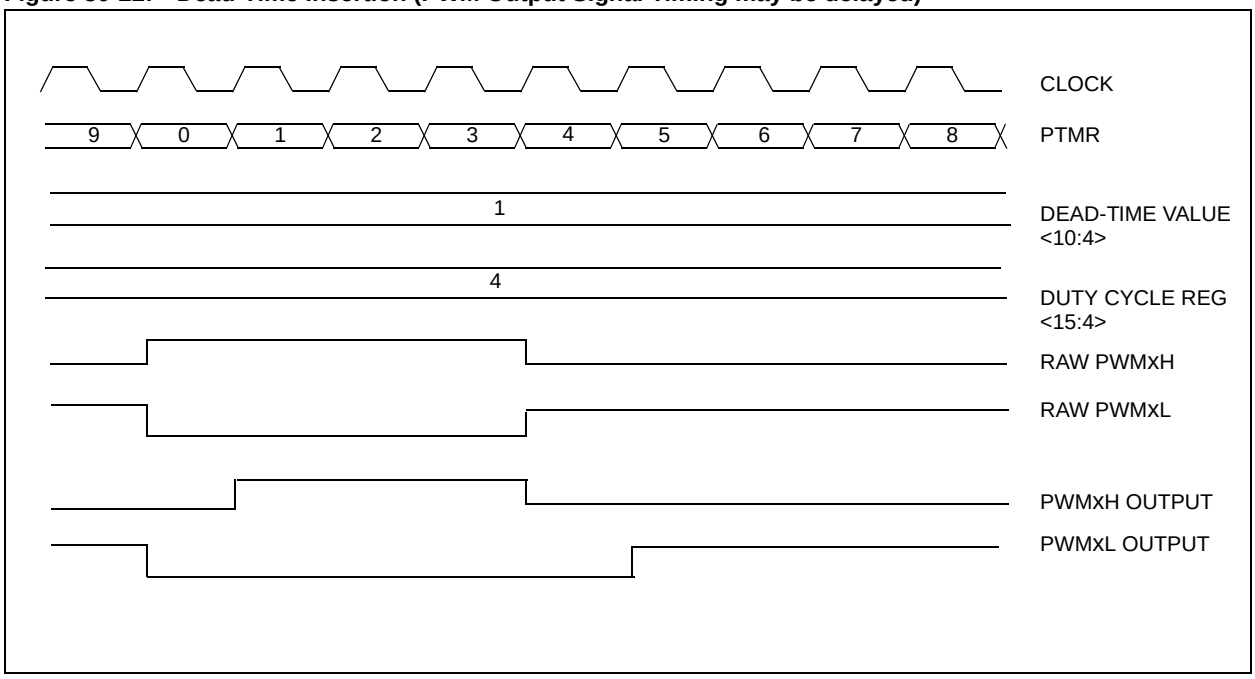
Example 30-3: Example Dead-Time Ranges

MIPS	Resolution	Dead-Time Range
30	4.16 ns	0-17.03 µs
20	6.25 ns	0-25.59 µs

30.15.4 Dead-Time Insertion Timing

Figure 30-22 shows how the dead-time insertion for complementary signals is accomplished.

Figure 30-22: Dead-Time Insertion (PWM Output Signal Timing may be delayed)



30.15.5 Dead-Time Distortion

For small PWM duty cycles, the ratio of dead-time to the active PWM time may become large. In this case, the inserted dead-time introduces distortion into waveforms produced by the PWM module. The user-assigned application can ensure that dead-time distortion is minimized by keeping the PWM duty cycle at least three times larger than the dead-time.

A similar effect occurs for duty cycles at or near 100%. The maximum duty cycle used in the application should be chosen such that the minimum inactive time of the signal is at least three times larger than the dead-time.

30.16 CONFIGURING A PWM CHANNEL

Example 30-4 shows a code example for configuring the PWM channel 1 to operate in Complementary PWM Output mode at 400 kHz with a dead-time value of approximately 64 ns. It is assumed that the dsPIC30F SMPS device is operating on the Internal Fast RC Oscillator with the PLL in the High Frequency Range (14.55 MHz input to the PLL).

Example 30-4: Code Example for Configuring PWM Channel 1

Note: This code example does not illustrate configuration of various fault modes for the PWM module. It is intended as a quick start guide for setting up the PWM Module.

```
mov #0x0400, w0                ; PWM module is disabled, continue operation in
mov w0, PTCON                  ; Idle mode, special event interrupt disabled,
                               ; immediate period updates enabled, no external
                               ; synchronization

; Set the PWM Period
mov #0x094D, w0                ; Select period to be approximately 2.5µs
mov w0, PTPER                  ; PLL Frequency is ~480MHz. This equates to a
                               ; clocke period of 2.1ns. The PWM period and
                               ; duty cycle registers are triggered on both +ve
                               ; and -ve edges of the PLL clock. Therefore,
                               ; one count of the PTPER and PDCx registers
                               ; equals 1.05ns.
                               ; So, to achieve a PWM period of 2.5µs, we
                               ; choose PTPER = 0x094D

mov #0x0000, w0                ; no phase shift for this PWM Channel
mov w0, PHASE1                 ; This register is used for generating variable
                               ; phase PWM

; Select individual Duty Cycle Control
mov #0x0001, w0                ; Fault interrupt disabled, Current Limit
mov w0, PWMCON1                ; interrupt disabled, trigger interrupt,
                               ; disabled, Primary time base provides timing,
                               ; DC1 provides duty cycle information, positive
                               ; dead time applied, no external PWM reset,
                               ; Enable immediate duty cycle updates

; Code for PWM Current Limit and Fault Inputs
mov #0x0003, w0
mov w0, FCLCON1                ; Disable current limit and fault inputs


; Code for PWM Output Control
mov #0xC000, w0                ; PWM1H and PWM1L is controlled by PWM module
mov w0, IOCON1                 ; Output polarities are active high, override
                               ; disabled

; Duty Cycle Setting
mov #0x04A6, w0                ; To achieve a duty cycle of 50%, we choose
mov w0, PDC1                   ; the PDC1 value = 0.5*(PWM Period)
                               ; The ON time for the PWM = 1.25µs
                               ; The Duty Cycle Register will provide
                               ; positive duty cycle to the PWMxH outputs
                               ; when output polarities are active high
                               ; (see IOCON1 register)

; Dead-Time Setting
mov #0x0040, w0                ; Dead time ~ 67ns
mov w0, DTR1                   ; Hex(40) = decimal(64)
                               ; So, Dead-time = 64*1.05ns = 67.2ns
                               ; Note that the last two bits are unimplemented,
                               ; therefore the dead-time register can achieve a
                               ; a resolution of about 4ns.
mov w0, ALTDTR1                ; Load the same value in ALTDTR1 register

bset PTCON, #15                ; turn ON PWM module
```

30.17 SPEED LIMITS OF PWM OUTPUT CIRCUITRY

The PWM output I/O buffers, and any attached circuits such as FET drivers and power FETs, have limited slew-rate capability. For very small PWM duty cycles, the PWM output signal is low-pass filtered; no pulse makes it through all of the circuitry.

A similar effect happens for duty cycle values near 100%. Before 100% duty cycle is reached, the output PWM signal appears to saturate at 100%. Such behavior should be taken into account in the user-assigned applications. In some power conversion applications, the duty cycle values nearing 0% or 100% are avoided, because, at these values, the user-assigned application is operating in a discontinuous mode or in a saturated mode, where the control loop may be non-functional.

30.18 PWM SPECIAL EVENT TRIGGER

The PWM module has a Special Event Trigger that allows Analog-to-Digital (A/D) conversions to be synchronized to the PWM time base. The A/D sampling and conversion time can be programmed to occur at any point within the PWM period. The Special Event Trigger allows the user-assigned application to minimize the delay between the time the A/D conversion results are acquired and the time the duty cycle value is updated.

The Special Event Trigger is based on the primary PWM time base. The PWM Special Event Trigger has one register (SEVTCMP) and four additional bits (SEVTPS<3:0> in PTCON) to control its operation. The PTMR value that causes a Special Event Trigger is loaded into the SEVTCMP register.

30.18.1 Special Event Trigger Enable

The PWM module always produces Special Event Trigger pulses. This signal can optionally be used by the Analog-to-Digital Converter (ADC) module. If the Special Event Interrupt Enable (SEIEN) bit is set in the PTCON (PTCON<11>) register, a PWM Special Event interrupt request is generated.

30.18.2 Special Event Trigger Postscaler

The PWM Special Event Trigger has a postscaler that allows a 1:1 to 1:16 postscale ratio. The postscaler is configured by writing the SEVTPS<3:0> control bits in the PTCON register.

The special event output postscaler is cleared on the following events:

- Any write to the SEVTCMP register
- Any device Reset

30.19 INDIVIDUAL PWM TRIGGERS

The PWM module also features an additional ADC trigger output for each PWM generator. This feature is very useful when the PWM generators are operating in Independent Time Base mode.

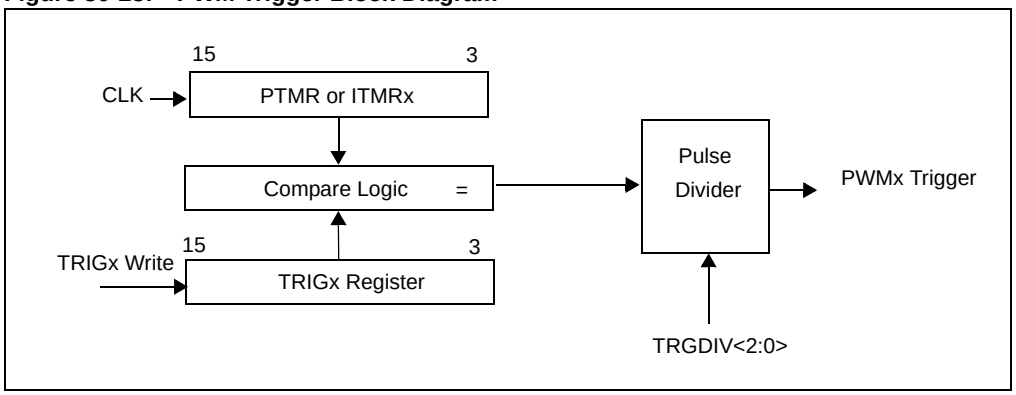
A block diagram of a trigger circuit is shown in Figure 30-23. The user-assigned application specifies a match value in the TRIGx register. When the local time base counter value matches the TRIGx value, an ADC trigger signal is generated (if configured to do so).

Trigger signals are always generated regardless of the TRIGx value as long as the TRIGx value is less than or equal to the PWM period value for the local time base. If the Trigger Interrupt Enable (TRGIEN) bit is set in the PWM Control (PWMCONx <10>) register, an interrupt request is generated.

The individual trigger outputs can be divided per the TRGDIV<2:0> bits in the TRGCONx registers, which allows the trigger signals to the ADC to be generated once for every 1, 2, 3, …, 8 trigger events.

The trigger divider allows the user-assigned application to tailor the ADC sample rates to the requirements of the control loop.

Figure 30-23: PWM Trigger Block Diagram



30.20 PWM INTERRUPTS

The PWM module can generate interrupts based on internal timing or on external signals through the current-limit and fault inputs. The Primary Time Base module can generate an interrupt request when a specified event occurs. Each PWM generator module has its own interrupt request signal to the interrupt controller. The interrupt for each PWM generator is an OR of the trigger event interrupt request, the current-limit input event or the fault input event for that module.

There are four Interrupt Request (IRQ) signals to the interrupt control, plus another interrupt request from the primary time base on special events.

The four IRQ coming from each PWM generator are called Individual PWM interrupts. The IRQ for each one of these individual interrupts can come from the PWM Individual Trigger, PWM Fault logic, or PWM Current-Limit logic. When an IRQ is generated from any of the above sources, the PWM interrupt flag (in the IFS1 register) corresponding to the chosen PWM generator will be set.

If more than one IRQ source is enabled, then the interrupt source must be resolved in software. This can be done by checking the TRGSTAT, FLTSTAT and CLSTAT bits in the PWMCONx register.

30.21 PWM TIME BASE INTERRUPTS

The PWM module can generate interrupts based on the primary time base and/or the individual time bases in each PWM generator. The interrupt timing is specified by the Special Event Comparison (SEVTCMP) register for the primary time base, and by the TRIGx registers for the individual time bases in the PWM generator modules.

The primary time base special event interrupt is enabled using the Special Event Interrupt Enable (SEIEN) bit in the PTCON (PTCON<11>) register. The individual time base interrupts generated by the trigger logic in each PWM generator are controlled by the Trigger Interrupt Enable (TRGIEN) bit in the PWM Control (PWMCONx<10>) registers.

30.22 PWM FAULT AND CURRENT-LIMIT PINS

The PWM module supports multiple fault pins for each PWM generator. These pins are labeled SFLTx (shared fault) or IFLTx (individual fault). The shared fault pins can be seen and used by any of the PWM generators. The individual fault pins can be used by specific PWM generators.

Each PWM generator can use one pin as a cycle-by-cycle current-limit, and another pin for use as a fault input. This fault input pin can be configured to reset the Fault state every PWM cycle, or have the Fault state remain latched until the Fault state has been reset in software.

30.23 LEADING-EDGE BLANKING (LEB)

Each PWM generator supports Leading-Edge Blanking (LEB) of the current-limit and fault inputs through the LEB<9:3> bits, and the PHR, PHF, PLR, PLF, FLTLEBEN and CLLEBEN bits in the LEBCONx registers. The purpose of LEB is to mask the transients that occur on the application printed circuit board when the power transistors are turned on and off.

The LEB bits support the blanking (ignoring) of the current-limit and fault inputs for a period of 0 ns to 1024 ns in 8.4 ns increments following any specified rising or falling edge of the coarse PWMxH and PWMxL signals. The coarse PWM signal (signal prior to the PWM fine tuning) has a resolution of 8.4 ns (at 30 MIPS), which is the same time resolution as the LEB counters.

The PHR, PHF, PLR and PLF bits select which edge of the PWMxH and PWMxL signals will start the blanking timer. If a new selected edge triggers the LEB timer while the timer is still active from a previously selected PWM edge, the timer is re-initialized and continues counting.

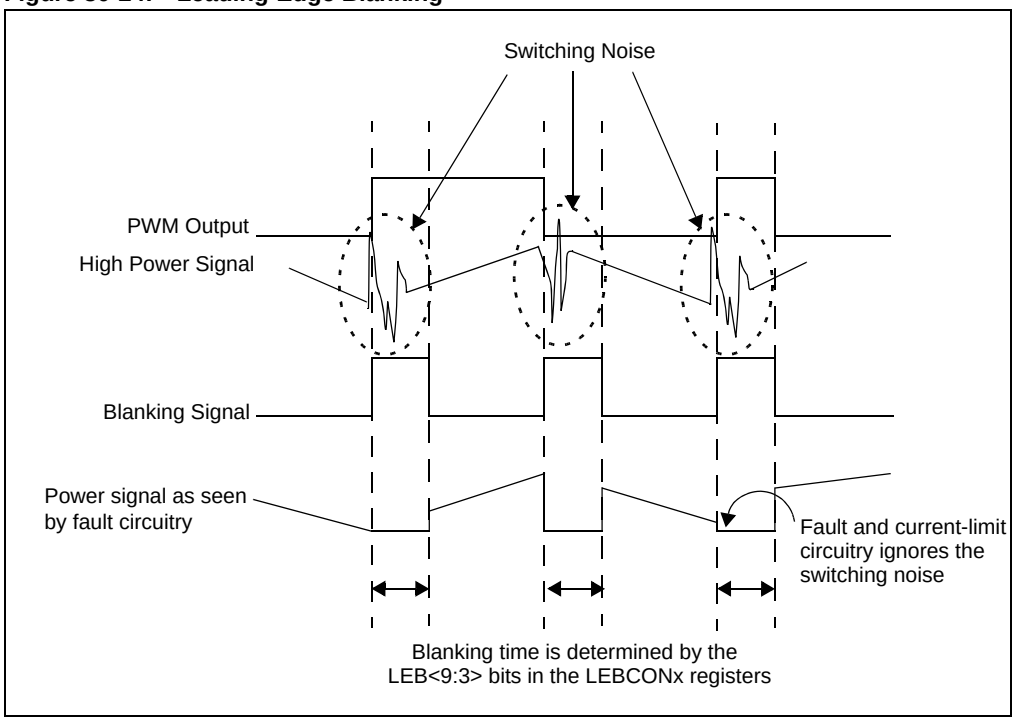
The FLTLEBEN and CLLEBEN bits enable the application of the blanking period to the selected fault and current-limit inputs.

The LEB duration at 30 MIPS = (LEB<9:3> + 1)/30 MHz * 4.

With a blanking period offset of 8.4 ns, a LEB<9:3> value of zero yields an effective blanking period of 8.4 ns. If a current-limit or fault input is active at the end of the previous PWM cycle, is still active at the start of the new PWM cycle and the dead-time is non-zero, the fault or current limit will be detected regardless of the LEB counter configuration.

Figure 30-24 illustrates the use of LEB in power conversion applications. The leading-edge Blanking (LEB) signal enables the PWM fault and current-limit circuitry to ignore noise generated by high-power switching.

Figure 30-24: Leading-Edge Blanking



30.24 PWM FAULT PINS

Each PWM generator can select its own fault input source from a selection of up to 10 fault or current-limit sources. In the Fault Current-Limit Control (FCLCONx) registers, each PWM generator has control (FLTSRC<3:0>) bits that specify the source for its fault input signal. Each PWM generator also has a Fault Interrupt Enable (FLTIEN) bit in the PWMCONx register that enables the generation of fault interrupt requests. Each PWM generator has an associated Fault Polarity (FLTPOL) bit in the FCLCONx register that selects the active level of the selected fault input.

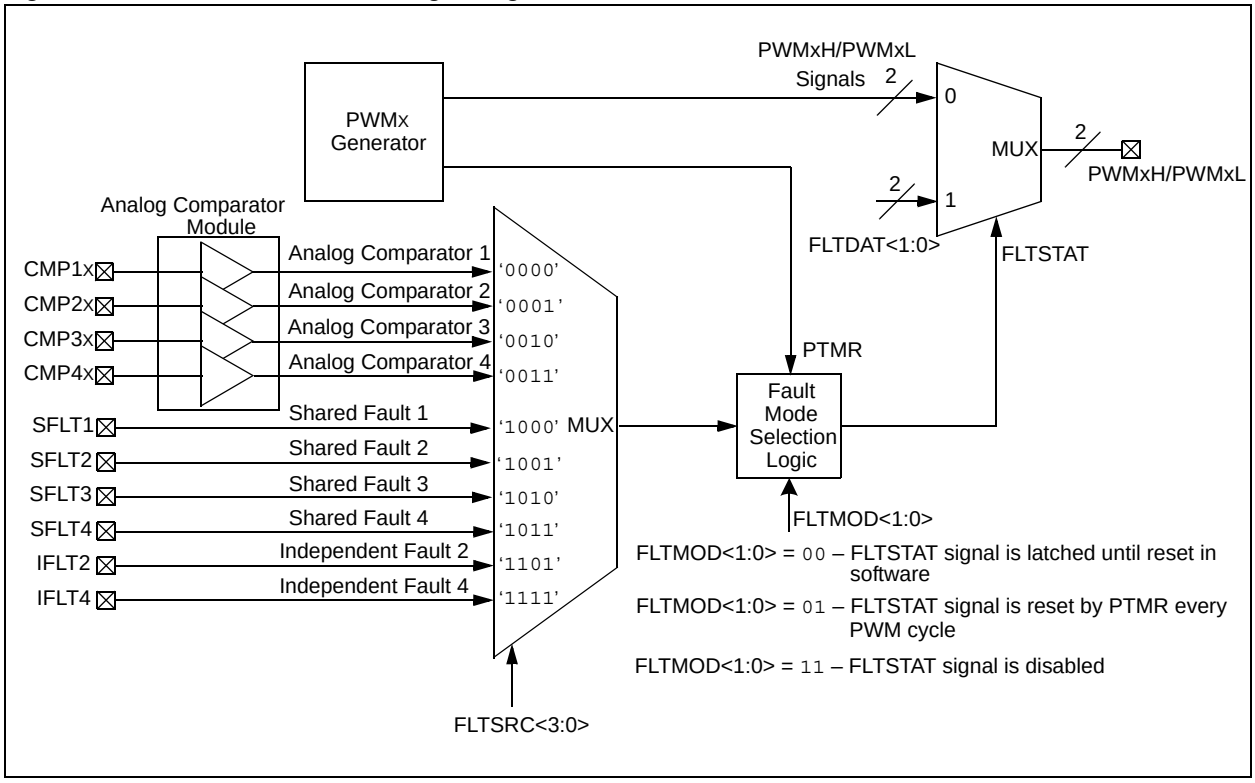
The fault pins serve two purposes. First is generation of fault overrides for the PWM outputs. The action of overriding the PWM outputs and generating an interrupt is performed asynchronously in hardware so that fault events are managed quickly. Second, the fault pin inputs can be used to implement either Current-Limit PWM mode or Current Reset mode.

PWM Fault states are available on the Fault Interrupt Status (FLTSTAT) bit in the PWMCONx registers. The FLTSTAT bit displays the fault IRQ latch if the FLTIEN bit is set. If fault interrupts are not enabled, the FLTSTAT bit displays the status of the selected FLTx input in positive logic format. When the fault input pins are not used in association with a PWM generator, these pins become general purpose I/O or interrupt input pins.

The FLTx pins are normally active-high. The FLTPOL bit in the FCLCONx registers, if set to '1', inverts the selected fault input signal so that it is an active-low.

The fault pins are also readable through the PORT I/O logic when the PWM module is enabled. This allows the user-assigned application to poll the state of the fault pins in software. Figure 30-25 provides a PWM fault control logic diagram.

Figure 30-25: PWM Fault Control Logic Diagram



30.24.1 Fault Interrupts

The FLTIENx bits in the PWMCONx registers determine whether an interrupt will be generated when the FLTx input is asserted high. The Fault mode for PWM Generator (FLTMOD) bits in the FCLCONx registers determine how the PWM generator and its outputs respond to the selected fault input pin. The FLTDAT<1:0> bits in the IOCONx registers supply the data values to be assigned to the PWMxH and PWMxL pins in the case of a fault.

The fault pin logic can operate separately from the PWM logic as an external interrupt pin. If the faults are disabled from affecting the PWM generators in the FCLCONx register, the fault pin can be used as a general purpose interrupt pin.

30.24.2 Fault States

The IOCONx registers have two bits (FLTDAT<1:0>) that determine the state of each PWMxH/PWMxL I/O pin when they are overridden by a fault input. When these bits are cleared, the PWM I/O pin is driven to the inactive state. If the bit is set, the PWM I/O pin is driven to the active state. The active and inactive states are referenced to the polarity defined for each PWM I/O pin (POLH and POLL polarity control bits).

30.24.3 Fault Input Modes

The fault input pin has two modes of operation: Latched mode and Cycle-by-Cycle mode.

- **Latched mode:** When the fault pin is asserted in Latched mode, the PWM outputs go to the states defined in the FLTDAT bits in the IOCONx registers. The PWM outputs remain in this state until the fault pin is deasserted and the corresponding interrupt flag has been cleared in software. When both of these actions have occurred, the PWM outputs return to normal operation at the beginning of the next PWM cycle boundary. If the FLTSTAT bit is cleared before the Fault condition ends, the PWM module waits until the fault pin is no longer asserted to restore the outputs. Software can clear the FLTSTAT bit by writing a zero to the FLTIEN bit.
- **Cycle-by-Cycle mode:** When the fault input pin is asserted in Cycle-by-Cycle mode, the PWM outputs remain in the deasserted PWM state for as long as the fault pin is asserted. For Complementary PWM Output mode, PWMxH is low (deasserted) and PWMxL is high (asserted). After the fault pin is driven high, the PWM outputs return to normal operation at the beginning of the following PWM cycle.

The operating mode for each fault input pin is selected using the FLTMOD<1:0> control bits in the FCLCONx register.

30.24.4 Fault Entry

The response of the PWM pins to the fault input pins is always asynchronous with respect to the device clock signals. That is, the PWM outputs should immediately go to the states defined in the FLTDAT register bits without any interaction from the dsPIC30F SMPS device or software. Refer to **30.29 “Fault and Current-Limit Override Issues with Dead-Time Logic”** for information regarding data sensitivity and behavior in response to current-limit or Fault events.

30.24.5 Fault Exit

After a Fault condition has ended, the PWM signals must be restored at a PWM cycle boundary to ensure proper synchronization of PWM signal edges and manual signal overrides. The next PWM cycle begins when the PTMR value is zero.

If Cycle-by-Cycle Fault mode is selected, the fault is automatically reset on every PWM cycle. No additional coding is needed to exit the fault condition.

For the Latched Fault mode, however, the following sequence must be followed to exit the fault condition:

1. Poll the PWM Fault Source to determine whether the fault signal has been deasserted.
2. If the PWM Fault interrupt is *not* enabled, skip the following sub-steps and proceed to step 3. If the PWM Fault interrupt is enabled, perform the following sub-steps and then proceed to step 4.
   a) Complete the PWM fault interrupt service routine (ISR).
   b) Disable the PWM fault interrupt by clearing the FLTIEN bit in the PWMCONx register.
   c) Enable the PWM fault interrupt by setting FLTMOD<1:0> = `0b00` in the FCLCONx register.
3. Disable PWM faults by setting FLTMOD<1:0> = `0b00` in the FCLCONx register.
4. Enable the latched PWM Fault mode by setting FLTMOD<1:0> = `0b00` in the FCLCONx register.

30.24.6 Fault Exit with PTMR Disabled

There is a special case for exiting a Fault condition when the PWM time base is disabled (PTEN = `0`). When a fault input is programmed for Cycle-by-Cycle mode, the PWM outputs are immediately restored to normal operation when the fault input pin is deasserted. The PWM outputs should return to their default programmed values (With the time base disabled, there is no reason to wait for the beginning of the next PWM cycle). When a fault input is programmed for Latched mode, the PWM outputs are restored immediately when the fault input pin is deasserted and the FLTSTAT bit has been cleared in software.

30.24.7 Fault Pin Software Control

The fault pin can be controlled manually in software. Since the fault input is shared with a PORT I/O pin, the PORT pin can be configured as an output by clearing the corresponding TRIS bit. When the PORT bit for the pin is cleared, the fault input will be activated.

Note:	Use caution when controlling the fault inputs in software. If the TRIS bit for the fault pin is cleared and the PORT bit is set high, the fault input cannot be driven externally.

30.25 PWM CURRENT-LIMIT PINS

Each PWM generator can select its own current-limit input source from up to 10 current-limit/fault sources. In the FCLCONx registers, each PWM generator has control bits (CLSRC<3:0>) that specify the source for its current-limit input signal. Additionally, each PWM generator has a Current-Limit Interrupt Enable (CLIEN) bit in the PWM Control (PWMCONx<11>) register that enables the generation of current-limit interrupt requests. Each PWM generator has an associated Current-Limit Polarity for PWM Generator bit (CLPOL) in the PWM Fault Current-Limit Control (FCLCONx<8>) register.

The current-limit pins serve two different purposes. They can be used to implement either Current-Limit PWM mode or Current Reset PWM mode.

When the CLIEN bit is set in the PWMCONx registers, the PWMxH and PWMxL outputs are forced to the values specified by the CLDAT<1:0> bits in the IOCONx register, if the selected current-limit input signal is asserted. This is called Current-Limit mode.

When the Current-Limit Mode Enable for PWM Generator (CLMODE) bit is zero, the External PWM Reset Control (XPRES) bit in the PWMCONx register is '1', and the PWM generator is in Independent Time Base mode (ITB = 1), then a current-limit signal resets the time base for the affected PWM generator. This behavior is called Current Reset mode, which is used in some PFC applications.

30.25.1 Current-Limit Interrupts

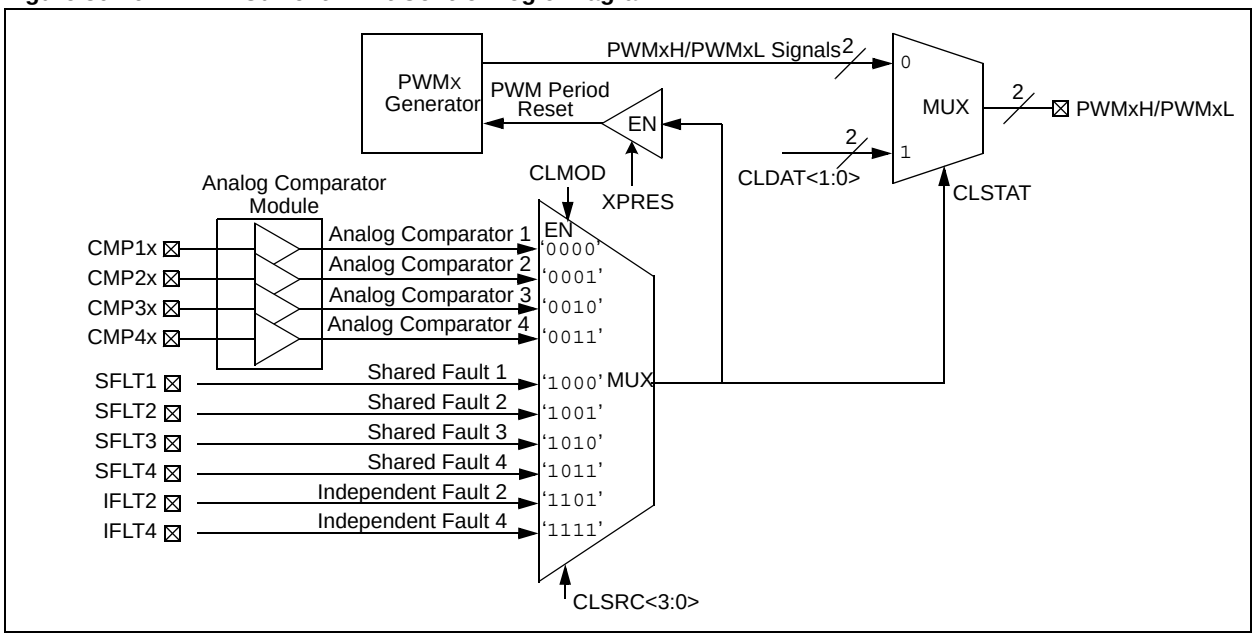
The state of the PWM current-limit conditions is available on the Current-Limit Interrupt Status (CLSTAT) bits in the PWM Control (PWMCONx<14>) registers. The CLSTAT bits display the current-limit IRQ flag, if the CLIEN bit is set. If current-limit interrupts are not enabled, the CLSTAT bits display the status of the selected current-limit inputs in positive logic format. When the current-limit input pin associated with a PWM generator is not used, these pins become general purpose I/O or interrupt input pins.

The current-limit pins are normally active-high. If set to '1', the CLPOL bit in the FCLCONx registers inverts the selected current-limit input signal to active-low.

The interrupts generated by the selected current-limit signals are combined to create a single interrupt request signal to the interrupt controller, which has its own interrupt vector, interrupt flag bit, interrupt enable bit, and interrupt priority bits associated with it.

The fault pins are also readable through the PORT I/O logic when the PWM module is enabled. This capability allows the user-assigned application to poll the state of the fault pins in software. Figure 30-26 provides a PWM current-limit control logic diagram.

Figure 30-26: PWM Current-Limit Control Logic Diagram



30.26 SIMULTANEOUS PWM FAULTS AND CURRENT LIMITS

The current-limit override function, if enabled and active, forces the PWMxH and PWMxL pins to the values specified by the CLDAT<1:0> bits in the IOCONx registers, unless the Fault function is enabled and active. If the selected fault input is active, the PWMxH and PWMxL outputs assume the values specified by the FLTDAT<1:0> bits in the IOCONx registers.

30.27 PWM FAULT AND CURRENT-LIMIT TRIGGER OUTPUTS TO ADC

The fault and current-limit source selection fields in the FCLCONx registers (FLTSRC<3:0> and CLSRC<3:0>) control multiplexers in each PWM Generator module. The control multiplexers select the desired fault and current-limit signals for their respective modules. The selected fault and current-limit signals are also available to the ADC module as trigger signals that initiate ADC sampling and conversion operations.

30.28 PWM OUTPUT OVERRIDE PRIORITY

If the PWM module is enabled, the priority of PWMxH/PWMxL pin ownership is as follows:

1. PWM Generator (lowest priority)
2. Output Override
3. Current-Limit Override
4. Fault Override
5. PENx (GPIO/PWM) ownership (highest priority)

If the PWM module is disabled, the GPIO module controls the PWMxH/PWMxL pins.

Example 30-5: PWM Output Pin Assignment

```
/* PWM Output pin control assigned to PWM generator */
IOCON1bits.PENH = 1;
IOCON1bits.PENL = 1;
```

Example 30-6: PWM Output Pin State Selection

```
/* High and Low switches set to active-high state */
IOCON1bits.POLH = 0;
IOCON1bits.POLL = 0;
```

Example 30-7: Enabling the Power Supply PWM Module

```
/* Enable High-Speed PWM module */

PTCONbits.PTEN = 1;
```

30.28.1 PWM Output Override Logic

The PWM Output Override feature is used to drive the individual PWM outputs to a desired state based on the system requirements. The output can be driven to both states: active state and inactive state. The Power Supply PWM module Override feature has the priority as assigned in the list presented in the previous section. All control bits associated with the PWM output override function are contained in the IOCONx register. If the PENH (IOCONx<15>) and PENL (IOCONx<14>) bits are set, the Power Supply PWM module controls the PWMxH/PWMxL output pins. The PWM Output Override bits allow the user application to manually drive the PWM I/O pins to specified logic states, independent of the duty cycle comparison units.

The OVRDAT<1:0> bits in the PWM I/O Control (IOCONx<7:6>) registers determine the state of the PWM I/O pins when a particular output is overridden by the OVRENH (IOCONx<9>) and OVRENL (IOCONx<8>) bits.

The OVRENH and OVRENL bits are active-high control bits. When these bits are set, the corresponding OVRDAT bit overrides the PWM output from the PWM generator.

When the PWM is in Complementary PWM Output mode, the dead-time generator is still active with overrides. The output overrides and fault overrides generate control signals used by the dead time unit to set the outputs as requested. Dead-time insertion can be performed when the PWM channels are overridden manually.

30.28.2 Override Priority

When the PENH and PENL bits are set, the following priorities apply to the PWM output:

1. If a fault is active, the Fault Override data (FLTDAT<1:0>) bits override all other potential sources and set the PWM outputs.
2. If a fault is not active, but a current-limit event is active, the CLDAT<1:0> (IOCONx<3:2>) bits are selected as the source to set the PWM outputs.
3. If neither a fault nor a current-limit event is active, and a user Override Enable bit is set to OVRENH, OVRENL, the associated OVRDAT<1:0> (IOCONx<7:6>) register bits set the PWM output.
4. If no override conditions are active, the PWM signals generated by the time base and duty cycle comparator logic are the sources that set the PWM outputs.

30.28.3 Override Synchronization

If the OSYNC bit in the PWM I/O Control (IOCONx<0>) register is set, the output overrides performed by the OVRENH (IOCONx<9>), OVRENL (IOCONx<8>) and OVRDAT<1:0> (IOCONx<7:6>) bits are synchronized to the PWM time base. Synchronous output overrides occur when the time base is zero. If PTEN = 0, meaning the PWM timer is not running, writes to IOCONx take effect on the next TCY boundary.

30.28.4 PENx (GPIO/PWM) Ownership

Most of the PWM output pins are normally mixed with other GPIO pins. When the Debugger halts the device, the PWM pins will take the GPIO characteristics that is multiplexed on that pin. For example, if the PWM1L and PWM1H pins are multiplexed with REϕ and RE1, the configuration of GPIO pins will decide the PWM output status when halted by the Debugger.

Example 30-8: Code Example

```
// the PWM output will be pulled to low when the device is halted by the
debugger
TRISE = 0x0000; REϕ and RE1 configured for an output
LATE = 0x0000; REϕ and RE1 configured as Low output
// the PWM output will be pulled to high when the device is halted by the
debugger
TRISE = 0x0000; REϕ and RE1 configured for an output
LATE = 0x0003; REϕ and RE1 configured as high output
// the PWM output will be in tristate when the device is halted by the
debugger
TRISE = 0x0003; REϕ and RE1 configured for an input
```

30.29 FAULT AND CURRENT-LIMIT OVERRIDE ISSUES WITH DEAD-TIME LOGIC

In the event of a Fault of Current-Limit condition, the data in the FLTDAT<1:0> bits or CLDAT<1:0> bits determine the state of the PWM I/O pins.

If any of the FLTDAT<1:0> or CLDAT<1:0> bits are '0', then PWMxH and/or PWMxL outputs are immediately driven low (deasserted), bypassing the dead-time logic. This behavior enables the PWM outputs to turn off immediately without any additional delays, as many power conversion applications require a fast response to fault shutdown signals for accurate control and/or to limit circuitry damage.

If any of the FLTDAT<1:0> or CLDAT<1:0> bits are '1', then PWMxH and/or PWMxL outputs pass through the dead-time logic and, therefore, will be delayed by the specified dead-time value. So, in this case, dead-time will be inserted even if a Fault or Current-Limit condition occurs.

30.30 ASSERTING OUTPUTS VIA CURRENT LIMIT

It is possible to use the CLDAT bits to assert the PWMxH and PWMxL outputs in response to a Current-Limit event. Such behavior could be used as a current "force" feature in response to an external current or voltage measurement that indicates a sudden sharp increase in the load on the power-converter output. Forcing the PWM to an ON state can be considered a "Feed-Forward" action that allows quick system response to unexpected load increases without waiting for the digital control loop to respond.

30.31 PWM IMMEDIATE UPDATE

For high-performance PWM control-loop applications, it may be desirable to force the duty cycle updates to occur immediately. Setting the IUE bit in the PWMCONx registers enables this feature.

In a closed-loop control application, any delay between the sensing of a system's state and the subsequent output of PWM control signals that drive the application reduces the loop stability. Setting the IUE bit minimizes the delay between writing the duty cycle registers and the response of the PWM generators to that change.

30.32 PWM OUTPUT OVERRIDE

All control bits associated with the PWM output override function are contained in the IOCONx register. If the PENH, PENL bits are set, the PWM module controls the PWMxH/PWMxL output pins.

The PWM output override bits allow the user to manually drive the PWM I/O pins to specified logic states independent of the duty cycle comparison units.

The OVRDAT<1:0> bits in the IOCONx registers determine the state of the PWM I/O pins when a particular output is overridden by the OVRENH and OVRENL bits.

The OVRENH and OVRENL bits are active-high control bits. When these bits are set, the corresponding OVRDAT bit overrides the PWM output from the PWM generator.

30.32.1 Complementary PWM Output Mode

When the PWM is in Complementary PWM Output mode, the dead-time generator is still active with overrides. The output overrides and fault overrides generate control signals used by the dead-time unit to set the outputs as requested, including dead-time. Dead-time insertion can be performed when the PWM channels are overridden manually.

30.32.2 Override Synchronization

If the OSYNC bit in the IOCONx register is set, the output overrides performed by the OVRENH, OVRENL and OVRDAT<1:0> bits are synchronized to the PWM time base. Synchronous output overrides occur when the time base is zero. If PTEN = 0, meaning the PWM timer is not running, writes to IOCONx take effect on the next TCY boundary.

30.33 FUNCTIONAL EXCEPTIONS

30.33.1 Power-on Reset Conditions

Upon a Power-on Reset, all registers associated with the PWM module are reset to the states as provided in Table 30-3. On a device Reset, the PWM output pins are tri-stated.

30.33.2 Sleep Mode

The selected fault input pin has the ability to wake the CPU from Sleep mode. The PWM module should generate an asynchronous interrupt, if any of the selected fault pins is driven low while in Sleep mode.

It is recommended that the user-assigned application disable the PWM outputs prior to entering Sleep mode. If the PWM module is controlling a power conversion application, the action of putting the device into Sleep mode causes any control loops to be disabled, and most applications will likely experience issues unless they are explicitly designed to operate in an Open-Loop mode.

30.33.3 CPU Idle Mode

The PWM module has a PTSIDL control bit in the PTCON register. This bit determines whether the PWM module continues to operate or stops when the device enters Idle mode. Stopped Idle mode functions like Sleep mode, and fault pins are asynchronously active.

- PTSIDL = 1 (Stop module when in Idle mode)
- PTSIDL = 0 (Do not stop module when in Idle mode)

It is recommended that the user-assigned application disable the PWM outputs prior to entering Idle mode. If the PWM module is controlling a power-conversion application, the action of putting the device into Idle mode causes any control loops to be disabled, and most applications will likely experience issues unless they are explicitly designed to operate in an Open-Loop mode.

30.34 REGISTER BIT ALIGNMENT

Table 30-3 shows the registers for the Power Supply PWM module. All time-based data for the module is always bit-aligned with respect to time. For example, bit 3 in the period register, the duty cycle registers, the dead-time registers, the trigger registers and the phase registers always represent a value of 8.4 ns, assuming 30 MIPS operation. Unused portions of registers are always read as zero.

The use of data alignment makes it easier to write software because it eliminates the need to shift time values to fit into registers. It also eases the computation and understanding of time allotment within a PWM cycle.

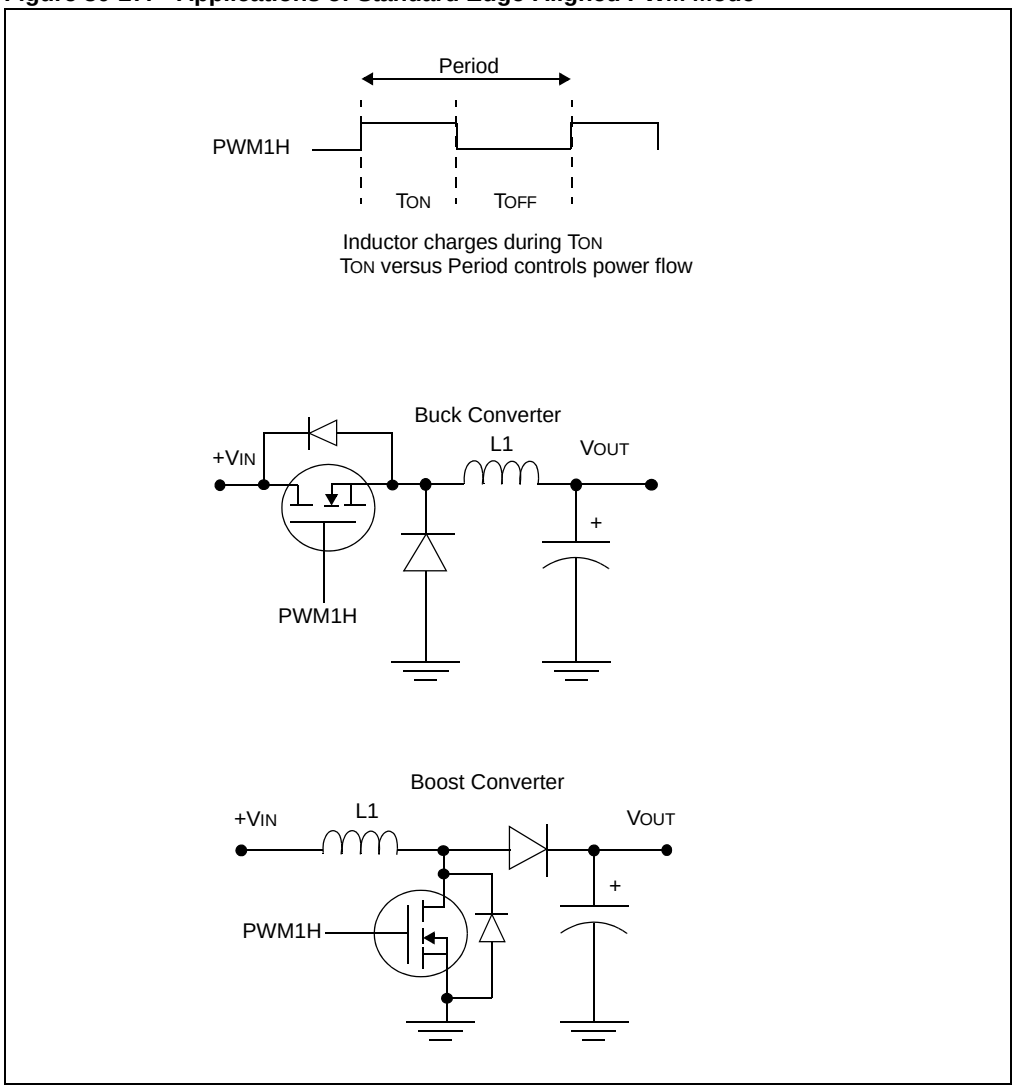
30.35 APPLICATION EXAMPLES

The following sections describe application examples for each of the available PWM modes.

30.35.1 Standard Edge-Aligned PWM Mode

In Standard Edge-Aligned PWM mode, the PWM output is typically connected to a single transistor, which charges an inductor, as shown in Figure 30-27. Buck and Boost converters typically use Standard Edge-Aligned PWM mode.

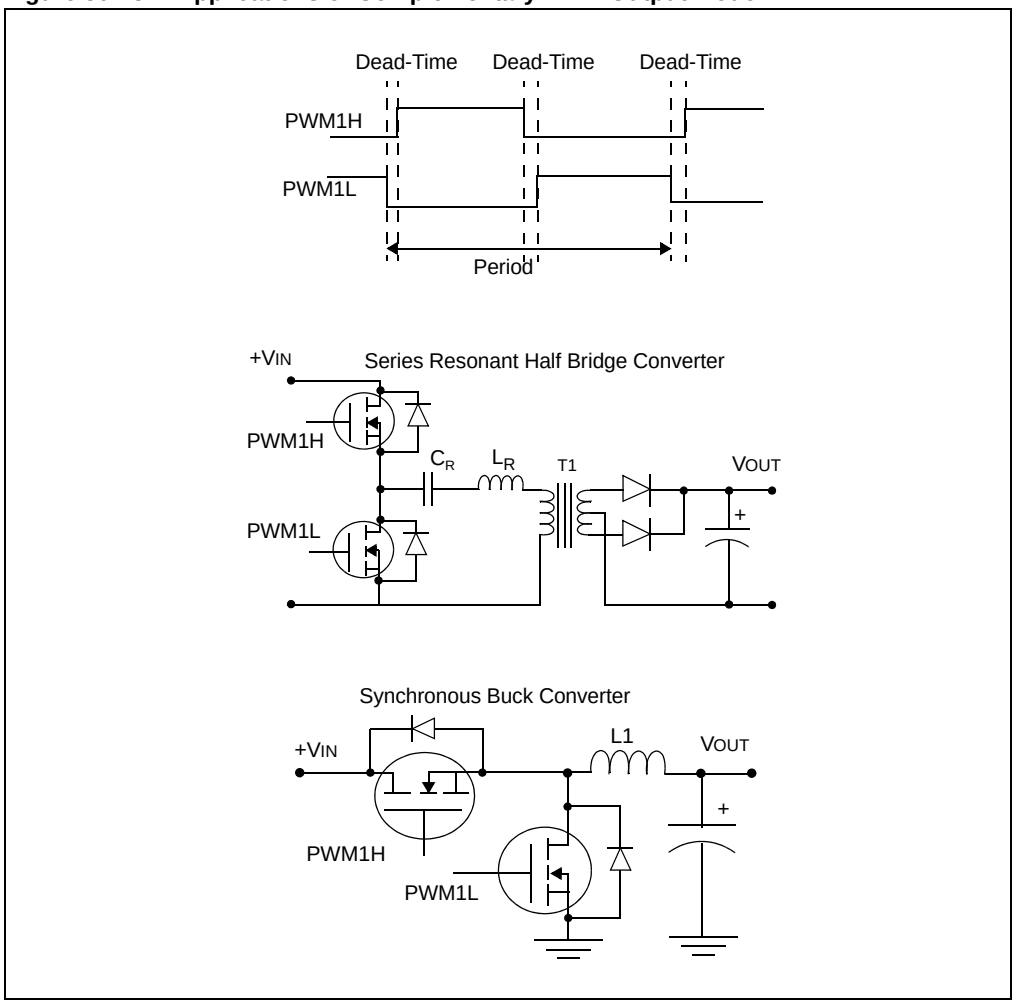
Figure 30-27: Applications of Standard Edge-Aligned PWM Mode



30.35.2 Complementary PWM Output Mode

The Complementary PWM Output mode shown in Figure 30-28 is often used in circuits that use two transistors in a bridge configuration where transformers are not used. If transformers are used, then some means must be provided to ensure that no net DC currents flow through the transformer, which would cause core saturation.

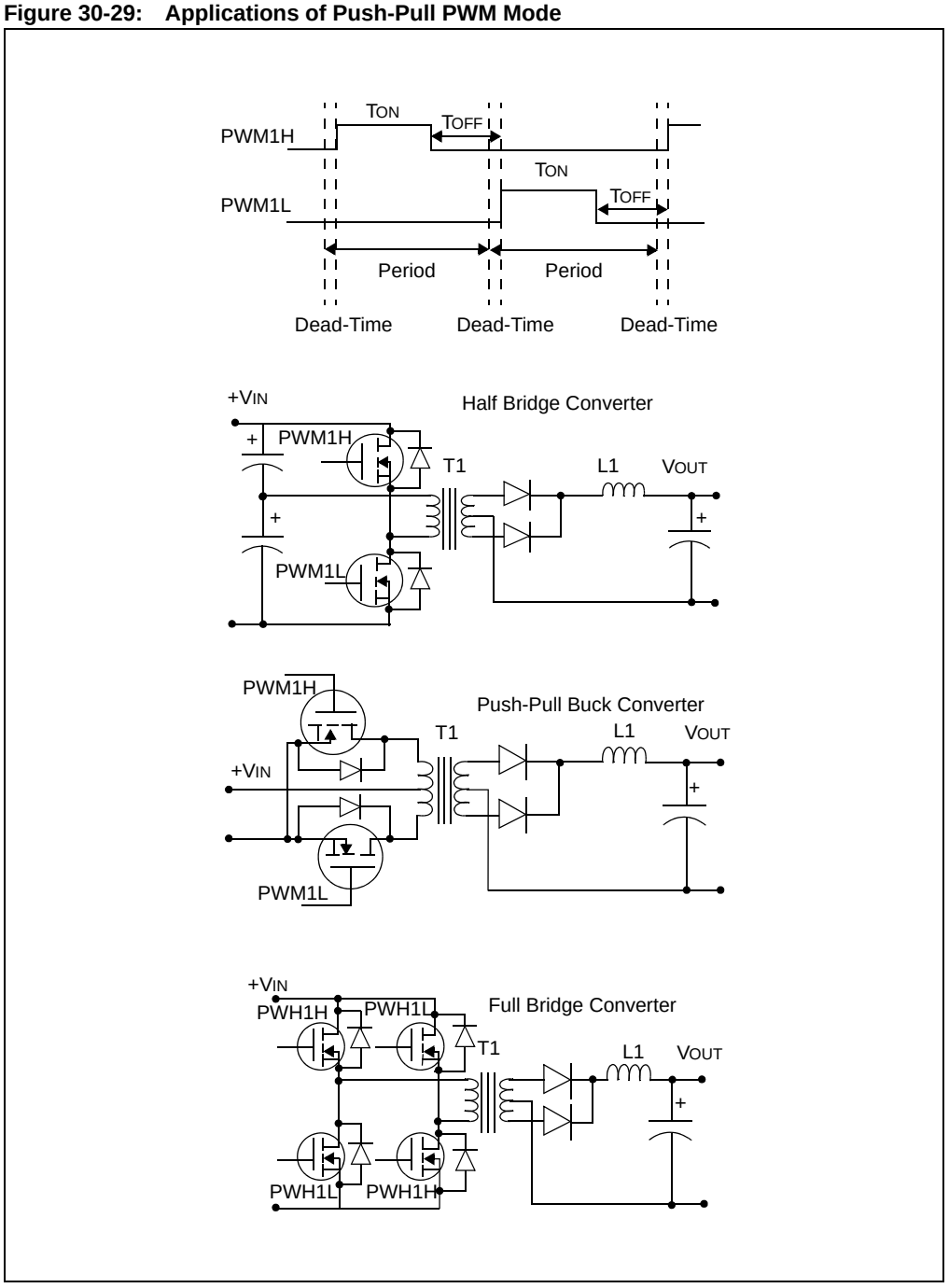
Figure 30-28: Applications of Complementary PWM Output Mode



30.35.3 Push-Pull PWM Mode

The Push-Pull PWM mode is typically used in transformer coupled circuits to ensure that no net DC currents flow through the transformer. Push-Pull mode ensures that the same duty cycle PWM pulse is applied to the transformer windings in alternate directions, as shown in Figure 30-29.

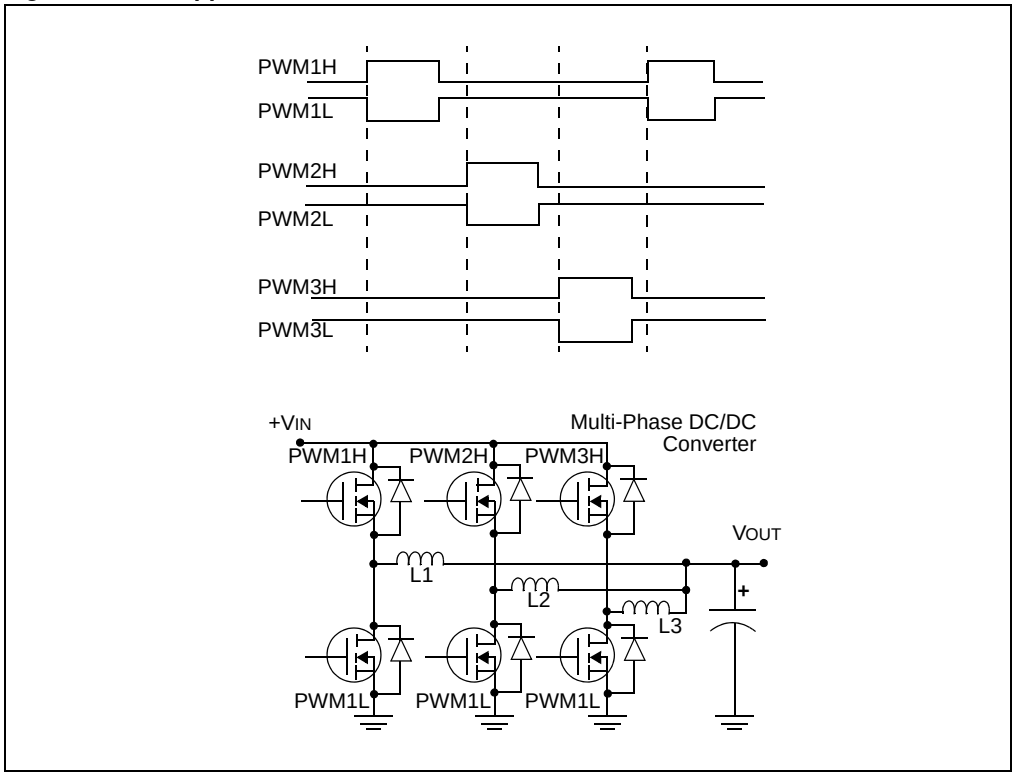
Figure 30-29: Applications of Push-Pull PWM Mode



30.35.4 Multi-Phase PWM Mode

The Multi-Phase PWM mode is often used in DC/DC converters that must handle very fast load current transients and fit into tight spaces. A multi-phase converter is essentially a parallel array of buck converters that are operated slightly out of phase of each other, as shown in Figure 30-30. The multiple phases create an effective switching speed equal to the sum of the individual converters. If a single phase is operating with a 333 kHz PWM frequency, the effective switching frequency for the circuit is 1 MHz. This high switching frequency greatly reduces output capacitor size requirements and improves load transient response.

Figure 30-30: Applications of Multi-Phase PWM Mode

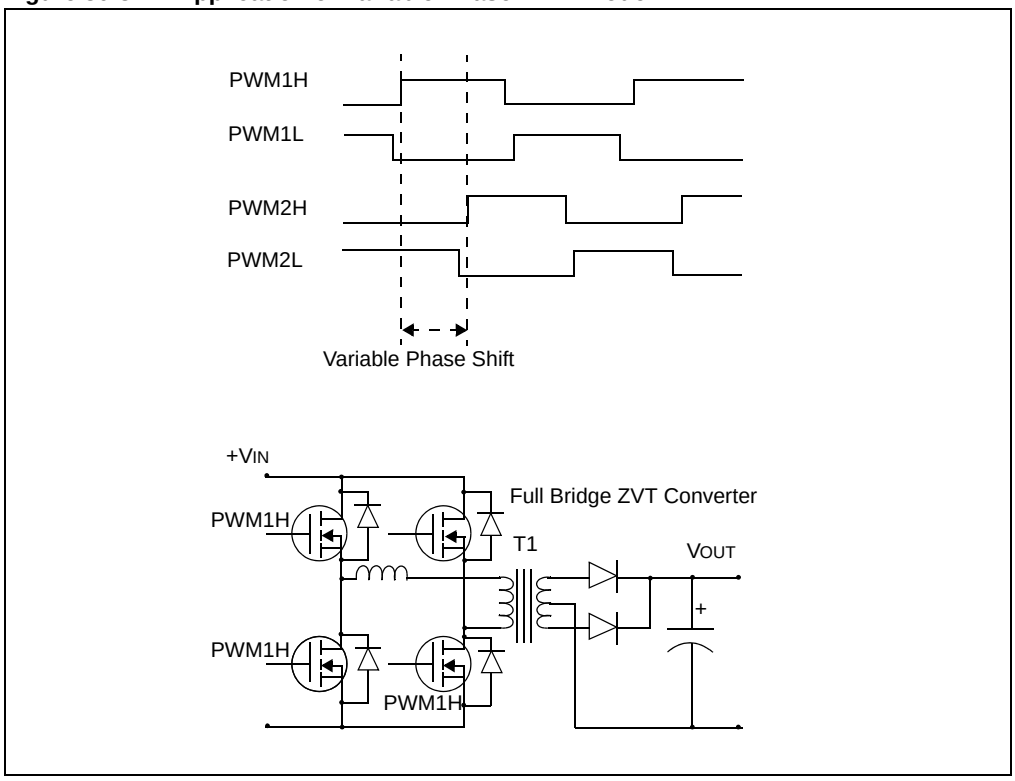


30.35.5 Variable Phase PWM Mode

The Variable Phase PWM mode is used in newer power conversion topologies that are designed to reduce switching losses. In standard PWM methods, any time a transistor switches between the conducting state and the non-conducting state (and vice versa), the transistor is exposed to the full current and voltage condition for the period of time it takes the transistor to turn on or off. The power loss (V * I * Tsw * FPWM) becomes appreciable at high frequencies. The Zero Voltage Switching (ZVS) and Zero Current Switching (ZCS) circuit topologies attempt to use quasi-resonant techniques to shift either the voltage or current waveforms relative to each other. This action makes either the voltage or the current zero at the time the transistor turns on or off. If either the current or the voltage is zero, no switching loss is generated.

In Variable Phase PWM modes, the duty cycle is fixed at 50%, and the power flow is controlled by varying the phase relationship between the PWM channels, as shown in Figure 30-31.

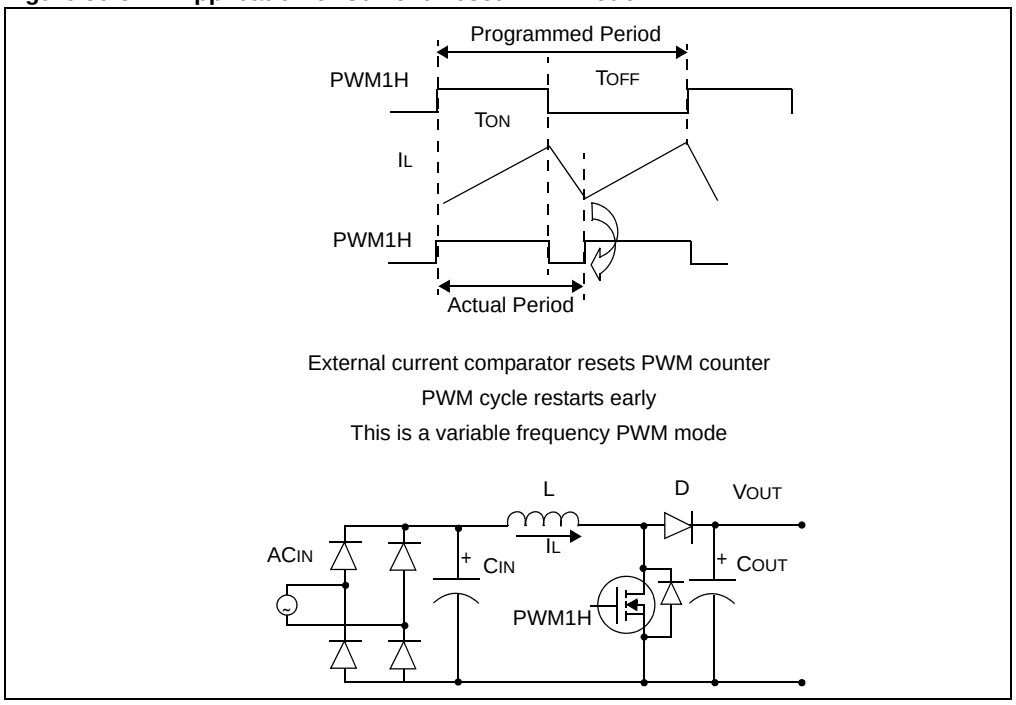
Figure 30-31: Application of Variable Phase PWM Mode



30.35.6 Current Reset PWM Mode

In Current Reset PWM mode, the PWM frequency varies with the load current. This mode differs from most PWM modes because the user-assigned application sets the maximum PWM period, but an external circuit measures the inductor current. When the inductor current falls below a specified value, the external current comparator circuit generates a signal that resets the PWM time base counter. The user-assigned application specifies a PWM “ON” time, and then, some time after the PWM signal becomes inactive, the inductor current falls below a specified value and the PWM counter is reset – earlier than the programmed PWM period. This mode is sometimes called Constant On-Time mode. This mode should not be confused with Cycle-by-Cycle Current-Limit PWM mode, where the PWM is asserted, an external circuit generates a current fault and the PWM signal is turned off before its programmed duty cycle would normally turn it off. In this mode as shown in Figure 30-32, the PWM frequency is fixed per the time base period.

Figure 30-32: Application of Current Reset PWM Mode



30.36 METHODS TO REDUCE EMI

The objective in reducing EMI is to move the PWM edges around in time to spread the EMI energy over a range of frequencies. This reduces the peak energy at any given frequency during the EMI measurement process, which measures long-term averages.

The EMI measurement process integrates the EMI energy into 9 kHz wide frequency bins. Assuming that the carrier (PWM) frequency is 150 kHz, a six percent dither will yield a 9 kHz wide dither.

30.36.1 Method 1: Programmable FRC Dither

This method dithers all the PWM outputs and the system clock. The advantage of this method is that no CPU resources are required. It is automatic once it is set up. The user-assigned application can periodically update these values to simulate a more random frequency pattern. Refer to **Section 29. "Oscillator"** (DS70268) in the "*dsPIC30F Family Reference Manual*" for more information on FRC Tuning and Dithering.

30.36.2 Method 2: Software Controlled Dither

This method uses software to dither individual PWM channels by scaling the duty cycle and period. This method consumes CPU resources as follows:

Assume:

- Four PWM channels updated @ 150 kHz rate
- 600 kHz x (5 clocks (2 mul, 1 tblrdl, 1 mov)) = 3 MIPS additional workload

30.36.3 Method 3: Software Scaling of Time Base Period

This method uses software to scale only the time base period. Assuming that the dither rate is relatively slow (about 250 Hz), the application control loop should be able to compensate for the changes in PWM period and adjust the duty cycle accordingly.

30.36.4 Method 4: Frequency Modulation

This method varies the frequency at which the PWM cycle is varied (dithered). The frequency modulation process is similar (mathematically speaking) to phase modulation when analyzed over a small time window.

The PWM module can phase modulate the PWM signals using the phase offset registers. Phase modulation has the advantage that the software is simpler and faster because multiple multiply operations (used for dithering frequency by scaling period and duty cycles) are replaced with fewer additions or simple updates of phase offset values into the phase registers.

This method also has the following advantages:

- Multi-phase and variable phase PWM modes can still be created.
- The PWM generators can still use the common time base, which simplifies determining when a "quiet time" is available for measuring current.

This method has one disadvantage: the phase modulation must be at a relatively high update rate to achieve usable frequency spreading.

30.36.5 Independent PWM Channel Dithering Issues

Multi-phase or variable phase designs using independent output dithering must consider the following issues:

- The phases are no longer phase aligned.
- Control of current sharing among phases is more difficult.

30.37 EXTERNAL SYNCHRONIZATION FEATURES

In large power conversion systems, it is often desirable to synchronize multiple power controllers to ensure that “beat frequencies” are not generated within the system, or as a means to ensure a “quiet” period during which time current and voltage measurements can be made.

dsPIC30F SMPS devices have input and/or output pins that provide the capability to either synchronize the dsPIC30F SMPS device with an external device or have external devices synchronized to the dsPIC30F SMPS device. These synchronizing features are enabled via the SYNCEN and SYNCOEN bits in the PTCON control register in the PWM module.

The SYNCPOL bit in the PTCON register selects whether the rising edge or falling edge of the SYNCI signal is the active edge. The SYNCPOL bit in the PTCON register also selects whether the SYNCO output pulse is active-low or active-high.

The SYNCSRC<2:0> bits in the PTCON register specify the source for the SYNCI signal.

If the SYNCI feature is enabled, the primary time base counter is reset when an active SYNCI edge is detected. If the SYNCO feature is enabled, an output pulse is generated when the primary time base counter rolls over at the end of a PWM cycle.

The recommended SYNCI pulse-width should be more than 100 ns. The expected SYNCO output pulse-width will be approximately 100 ns.

When using the SYNCI feature, it is recommended that the user-assigned application program the period register with a period value that is slightly longer than the expected period of the external synchronization input signal. This provides protection in case the SYNCI signal is not received due to noise or external component failure. With a reasonable period value programmed into the PTPER register, the local power conversion process should remain operational even if the global synchronization signal is not received.

Note:	The mismatch in the internal versus external synchronization period may produce unpredictable results on Multi-Phase PWM and Phase Shift PWM output configuration.

30.38 CPU LOAD STAGGERING

The dsPIC30F SMPS device has the ability to stagger the individual trigger comparison operations. This feature helps to level the processor’s workload to minimize situations where the processor is overloaded.

Assume a situation where four PWM channels are controlling four independent voltage outputs. Assume further that each PWM generator is operating at 500 kHz (2 µs period) and each control loop is operating at 125 kHz (8 µs).

The TRGDIV<2:0> bits in each TRGCONx register will be set to ‘011’, which specifies that every fourth trigger comparison match will generate a trigger signal to the ADC to capture data and begin a conversion process.

If the stagger-in-time feature did not exist, all of the requests from all of the PWM trigger registers might occur at the same time. If this “pile-up” were to happen, some data samples might become stale (outdated) by the time the data for all four channels can be processed.

With the Stagger-in-Time feature, the trigger signals are spaced out over time (during succeeding PWM periods) so that all of the data is processed in an orderly manner.

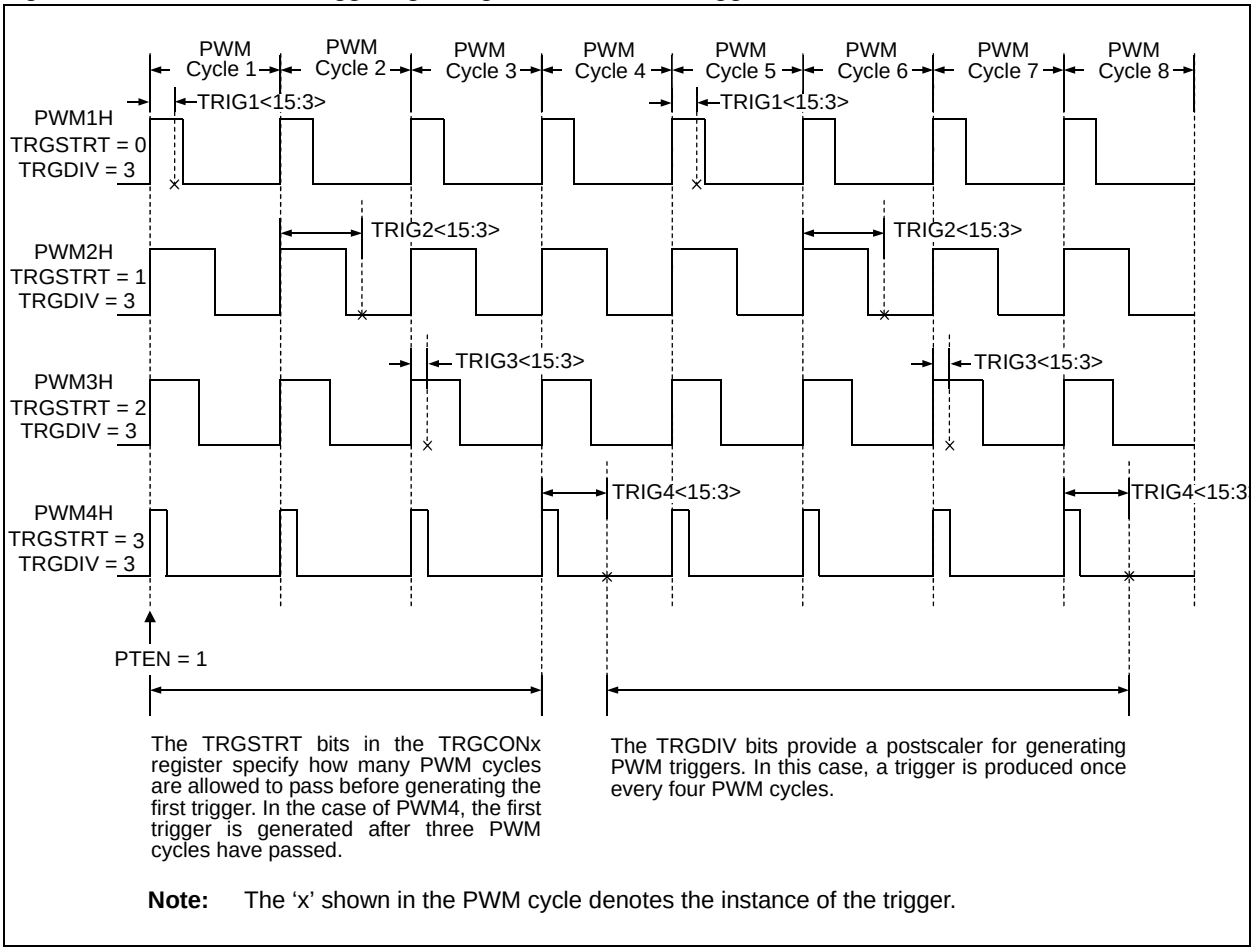
The ROLL counter is a counter connected to the primary time base counter and is incremented each time the primary time base counter reaches terminal count (period rollover).

The Stagger-in-Time feature is controlled by the TRGSTRT<5:0> bits in the TRGCONx registers. The TRGSTRT<5:0> bits specify the count value of the ROLL counter that must be matched before an individual trigger comparison module in each of the PWM generators can begin to count the trigger comparison events as specified by the TRGDIV<2:0> bits in the TRGCONx registers.

In the preceding example with the four PWM generators, the first PWM TRGSTRT<5:0> bits would be '000', the second PWM TRGSTRT bits would be set to '001', the third PWM TRGSTRT bits would be set to '010', and the fourth PWM TRGSTRT bits would be set to '011'. Therefore, over a total of four PWM cycles, the four separate control loops could be run, each with their own 2 μs time period.

Figure 30-33 illustrates the preceding example of CPU load staggering using the individual PWM triggers.

Figure 30-33: CPU Load Staggering Using Individual PWM Triggers



Note: The 'x' shown in the PWM cycle denotes the instance of the trigger.

30.39 EXTERNAL TRIGGER BLANKING

Using the LEB<9:3> bits in the LEBCONx registers, the PWM module has the capability to blank (ignore) the external current and fault inputs for a period of 0 ns to 1024 ns. This feature is useful, if the power transistor turn-on induced transients make current sensing difficult at the start of a PWM cycle.

30.40 REGISTER MAP

A summary of the registers associated with the PMP module is provided in Table 30-3.

Table 30-3: Power Supply PWM Register Map

File Name	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
PTCON	PTEN	—	PTSIDL	SESTAT	SEIEN	EIPU	SYNCPOL	SYNCOEN	SYNCEN	SYNCSRC<2:0>			SEVTPS<3:0>				0000
PTPER	PTPER<15:3>													—	—	—	FFF0
MDC	MDC<15:0>																0000
SEVTCMP	SEVTCMP<15:3>													—	—	—	0000
PWMCON1	FLTSTAT	CLSTAT	TRGSTAT	FLTIEN	CLIEN	TRGIEN	ITB	MDCS	DTC<1:0>		—	—	—	—	XPRES	IUE	0000
IOCON1	PENH	PENL	POLH	POLL	PMOD<1:0>		OVRENH	OVRENL	OVRDAT<1:0>		FLTDAT<1:0>		CLDAT<1:0>		—	OSYNC	0000
FCLCON1	—	—	—	CLSRC<3:0>				CLPOL	CLMODE	FLTSRC<3:0>				FLTPOL	FLTMOD<1:0>		0000
PDC1	PDC1<15:0>																0000
PHASE1	PHASE1<15:2>														—	—	0000
DTR1	—	—	DTR1<13:2>												—	—	0000
ALTDTR1	—	—	ALTDTR1<13:2>												—	—	0000
TRIG1	TRGCMP<15:3>													—	—	—	0000
TRGCON1	TRGDIV<2:0>			—-	—-	—-	—-	—-	—-	—-	TRGSTRT<5:0>						0000
LEBCON1	PHR	PHF	PLR	PLF	FLTLEBEN	CLLEBEN	LEB<9:3>							—	—	—	0000
PWMCON2	FLTSTAT	CLSTAT	TRGSTAT	FLTIEN	CLIEN	TRGIEN	ITB	MDCS	DTC<1:0>		—	—	—	—	XPRES	IUE	0000
IOCON2	PENH	PENL	POLH	POLL	PMOD<1:0>		OVRENH	OVRENL	OVRDAT<1:0>		FLTDAT<1:0>		CLDAT<1:0>		—	OSYNC	0000
FCLCON2	—	—	—	CLSRC<3:0>				CLPOL	CLMODE	FLTSRC<3:0>				FLTPOL	FLTMOD<1:0>		0000
PDC2	PDC2<15:0>																0000
PHASE2	PHASE2<15:2>														—	—	0000
DTR2	—	—	DTR2<13:2>												—	—	0000
ALTDTR2	—	—	ALTDTR2<13:2>												—	—	0000
TRIG2	TRGCMP<15:3>													—	—	—	0000
TRGCON2	TRGDIV<2:0>			—-	—-	—-	—-	—-	—-	—-	TRGSTRT<5:0>						0000
LEBCON2	PHR	PHF	PLR	PLF	FLTLEBEN	CLLEBEN	LEB<9:3>							—	—	—	0000
PWMCON3	FLTSTAT	CLSTAT	TRGSTAT	FLTIEN	CLIEN	TRGIEN	ITB	MDCS	DTC<1:0>		—	—	—	—	XPRES	IUE	0000
IOCON3	PENH	PENL	POLH	POLL	PMOD<1:0>		OVRENH	OVRENL	OVRDAT<1:0>		FLTDAT<1:0>		CLDAT<1:0>		—	OSYNC	0000
FCLCON3	—	—	—	CLSRC<3:0>				CLPOL	CLMODE	FLTSRC<3:0>				FLTPOL	FLTMOD<1:0>		0000
PDC3	PDC3<15:0>																0000
PHASE3	PHASE3<15:2>														—	—	0000
DTR3	—	—	DTR3<13:2>												—	—	0000
ALTDTR3	—	—	ALTDTR3<13:2>												—	—	0000
TRIG3	TRGCMP<15:3>													—	—	—	0000
TRGCON3	TRGDIV<2:0>			—-	—-	—-	—-	—-	—-	—-	TRGSTRT<5:0>						0000

Table 30-3: Power Supply PWM Register Map (Continued)

File Name	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
LEBCON3	PHR	PHF	PLR	PLF	FLTLEBEN	CLLEBEN	LEB<9:3>							—	—	—	0000
PWMCON4	FLTSTAT	CLSTAT	TRGSTAT	FLTIEN	CLIEN	TRGIEN	ITB	MDCS	DTC<1:0>		—	—	—				0000
IOCON4	PENH	PENL	POLH	POLL	PMOD<1:0>		OVRENH	OVRENL	OVRDAT<1:0>		FLTDAT<1:0>		CLDAT<1:0>		—	OSYNC	0000
FCLCON4	—	—	—	CLSRC<3:0>				CLPOL	CLMODE	FLTSRC<3:0>				FLTPOL	FLTMOD<1:0>		0000
PDC4	PDC4<15:0>																0000
PHASE4	PHASE4<15:2>														—	—	0000
DTR4	—	—	DTR4<13:2>												—	—	0000
ALTDTR4	—	—	ALTDTR4<13:2>												—	—	0000
TRIG4	TRGCMP<15:3>													—	—	—	0000
TRGCON4	TRGDIV<2:0>			—-	—-	—-	—-	—-	—-	—-	TRGSTRT<5:0>						0000
LEBCON4	PHR	PHF	PLR	PLF	FLTLEBEN	CLLEBEN	LEB<9:3>							—	—	—	0000
Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

30.41 RELATED APPLICATION NOTES

This section lists application notes that are related to this section of the manual. These application notes may not be written specifically for the dsPIC30F SMPS device product family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the Power Supply PWM module include the following:

Title	Application Note #
No related application notes at this time.	

Note: Please visit the Microchip web site (www.microchip.com) for additional Application Notes and code examples for the dsPIC30F SMPS device family.

30.42 REVISION HISTORY

Revision A (February 2007)

This is the initial released version of this document.

Revision B (February 2007)

Minor edits incorporated throughout the document.

Revision C (July 2009)

This revision includes the following updates:

- Examples:
  - Added Example 30-1 in **30.9 “PWM Period”** .
  - Added Example 30-2 in **30.10.1 “Phase Shift”** .
- Equations:
  - Updated Equation 30-1 in **30.9 “PWM Period”** .
  - Added Equation 30-2 for PWM duty cycle resolution in **30.10 “PWM Frequency and Duty Cycle Resolution”** .
  - Updated Equation 30-3 in **30.10 “PWM Frequency and Duty Cycle Resolution”** .
- Figures:
  - Added Figure 30-15, Figure 30-16 and Figure 30-17 in **30.10 “PWM Frequency and Duty Cycle Resolution”** .
- Notes:
  - Added a note on FRC clock selection in Equation 30-1.
  - Added a note on the mismatch in internal versus external synchronization period (see **30.37 “External Synchronization Features”**).
- Registers:
  - Added the HS and HC labels in the Legend section in Register 30-5.
- Sections:
  - Updated the PWM Period description (see **30.9 “PWM Period”**).
  - Updated **30.10 “PWM Frequency and Duty Cycle Resolution”** .
  - Added a sub-section **30.10.1 “Phase Shift”** in **30.10 “PWM Frequency and Duty Cycle Resolution”** .
  - Updated **30.24.5 “Fault Exit”** .
  - Added the following sub-sections: **30.28.1 “PWM Output Override Logic”** **30.28.2 “Override Priority”** , **30.28.3 “Override Synchronization”** and **30.28.4 “PENx (GPIO/PWM) Ownership”** .
- Tables:
  - Updated the MIPS values and PWM Frequency values in Table 30-2.
- Additional minor corrections such as language and formatting updates are incorporated throughout the document.

NOTES: