Section 14. High-Speed PWM

HIGHLIGHTS

This section of the manual contains the following major topics:

Note: This family reference manual section is meant to serve as a complement to device data sheets. Depending on the device variant, this manual section may not apply to all dsPIC33E/PIC24E devices.

Please consult the note at the beginning of the **"High-Speed PWM"** chapter in the current device data sheet to check whether this document supports the device you are using.

Device data sheets and family reference manual sections are available for download from the Microchip Worldwide Web site at: http://www.microchip.com

14.1 INTRODUCTION

This section describes the High-Speed Pulse-Width Modulator (PWM) module and its associated operational modes. The High-Speed PWM module in the dsPIC33E/PIC24E device family supports a wide variety of PWM modes and is ideal for power conversion/motor control applications. Some of the common applications include:

- AC-to-DC converters
- DC-to-DC converters
- AC and DC motors
- Inverters
- Battery chargers
- Digital lighting
- Uninterrupted Power Supply (UPS)
- Power Factor Correction (PFC) (e.g., Interleaved PFC and Bridgeless PFC)

14.2 FEATURES

The High-Speed PWM module consists of the following major features:

- Up to seven PWM generators, each with an individual time base
- Two PWM outputs per PWM generator
- Individual period and duty cycle for each PWM output
- Duty cycle, dead time, phase shift and frequency resolution of 8.32 ns at 60 MIPS
- Independent fault and current-limit inputs for 14 PWM outputs
- Redundant Output mode
- True Independent Output mode
- Push-Pull Output mode
- Complimentary Output mode
- Center-Aligned PWM mode
- Output override control
- Special Event Trigger
- PWM capture feature
- Prescaler for input clock
- ADC triggering with PWM
- Independent PWM frequency, duty cycle and phase shift changes
- Leading-Edge Blanking (LEB) functionality
- Dead time compensation
- Output clock chopping

14.3 CONTROL REGISTERS

The following registers control the operation of the High-Speed PWM module:

- **PTCON: PWM Time Base Control Register**
  - Enables or disables the High-Speed PWM module
  - Sets the Special Event Trigger for the ADC
  - Enables or disables immediate period updates
  - Selects the synchronizing source for the master time base
  - Specifies synchronization settings
- **PTCON2: PWM Clock Divider Select Register**
  Provides the clock prescaler to the PWM master time base
- **PTPER: Primary Master Time Base Period Register**
  Provides the PWM time period value
- **STCON: PWM Secondary Master Time Base Control Register**
  - Enables or disables immediate period updates based on the secondary master time base
  - Selects the synchronization source for the secondary master time base
  - Specifies the synchronization setting for secondary master time base control
- **STCON2: PWM Secondary Clock Divider Select Register**
  Provides the clock prescaler to the PWM secondary master time base
- **STPER: Secondary Master Time Base Period Register**
  Provides the secondary master time base period value
- **MDC: PWM Master Duty Cycle Register**
  Provides the PWM master duty cycle value
- **SEVTCMP: PWM Special Event Compare Register**
  Provides the compare value that is used to trigger the ADC module
- **SSEVTCMP: PWM Secondary Special Event Compare Register**
  Provides the compare value that is used to trigger the ADC module based on the secondary master time base
- **CHOP: PWM Chop Clock Generator Register**
  - Provides the chop clock frequency
  - Enables or disables the chop clock generator
- **PWMKEY: PWM Unlock Register**
  Writes the unlock sequence to allow writes to the IOCONx and FCLCONx registers
- **PWMCONx: PWM Control Register**
  - Enables or disables fault interrupt, current-limit interrupt and primary trigger interrupt
  - Provides the interrupt status for fault interrupt, current-limit interrupt and primary trigger interrupt
  - Selects the type of time base (master time base or independent time base)
  - Selects the type of duty cycle (master duty cycle or independent duty cycle)
  - Controls Dead Time mode
  - Enables or disables Center-Aligned mode
  - Controls the external PWM Reset operation
  - Enables or disables immediate updates of the duty cycle, phase offset, independent time base period

- **IOCONx: PWM I/O Control Register**
  - Enables or disables PWM pin control feature (PWM control or GPIO)
  - Controls fault/current limit override values
  - Enables PWMxH and PWMxL pin swapping
  - Controls the PWMxH and PWMxL output polarity
  - Controls the PWMxH and PWMxL output if any of the following modes is selected:
    - Complementary mode
    - Push-Pull mode
    - True Independent mode
- **FCLCONx: PWM Fault Current-Limit Control Register**
  - Selects the current-limit control signal source
  - Selects the current-limit polarity
  - Enables or disables Current-Limit mode
  - Selects the fault control signal source
  - Configures the fault polarity
  - Enables or disables Fault mode
- **PDCx: PWM Generator Duty Cycle Register(1)**
  - Provides the duty cycle value for the PWMxH and PWMxL outputs, if master time base is selected
  - Provides the duty cycle value for the PWMxH output, if independent time base is selected
- **PHASEx: PWM Primary Phase Shift Register**
  - Provides the phase shift value for the PWMxH and PWMxL output, if master time base is selected
  - Provides the independent time base period for the PWMxH output, if independent time base is selected
- **SDCx: PWM Secondary Duty Cycle Register(1)**

  Provides the duty cycle value for the PWMxL output, if independent time base is selected
- **SPHASEx: PWM Secondary Phase Shift Register(1,2)**
  - Provides the phase shift for the PWMxL output, if the master time base is selected
  - Provides the independent time base period value for the PWMxL output, if the independent time base is selected
- **DTRx: PWM Dead Time Register**
  - Provides the dead time value for the PWMxH output, if positive dead time is selected
  - Provides the dead time value for the PWMxL output, if negative dead time is selected
- **ALTDTRx: PWM Alternate Dead Time Register**
  - Provides the dead time value for the PWMxL output, if positive dead time is selected
  - Provides the dead time value for the PWMxH output, if negative dead time is selected
- **TRIGx: PWM Primary Trigger Compare Value Register**

  Provides the compare value to generate the primary PWM trigger
- **TRGCONx: PWM Trigger Control Register**
  - Enables the PWMx trigger postscaler start event
  - Specifies the number of PWM cycles to skip before generating the first trigger
- **LEBCONx: Leading-Edge Blanking Control Register**
  - Selects the rising or falling edge of the PWM output for LEB
  - Enables or disables LEB for fault and current-limit inputs
- **LEBDLYx: Leading-Edge Blanking Delay Register**

  Provides leading-edge blanking delay for the fault and current-limit inputs

- **PWMCAPx: Primary PWM Time Base Capture Register**

  Provides the captured independent time base value when a leading edge is detected on the current-limit input, and when LEB processing on the current-limit input signal is completed

- **AUXCONx: PWM Auxiliary Control Register**
  - Selects PWM state blank and chop clock sources
  - Selects PWMxH and PWMxL output chopping functionality

Register 14-1: PTCON: PWM Time Base Control Register

R/W-0	U-0	R/W-0	HS/HC-0	R/W-0	R/W-0	R/W-0	R/W-0
PTEN	—	PTSIDL	SESTAT	SEIEN	EIPU(1)	SYNCPOL(1)	SYNCOEN(1)
bit 15							bit 8

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(1)	SYNCSRC<2:0>(1)			SEVTPS<3:0>(1)			
bit 7							bit 0

Legend:	HC = Cleared in Hardware	HS = Set in Hardware	
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

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)
1 = Active Period register is updated immediately
0 = Active Period register updates occur on PWM cycle boundaries

bit 9 **SYNCPOL:** Synchronize Input and Output Polarity bit(1)
1 = SYNCIx/SYNCO polarity is inverted (active-low)
0 = SYNCIx/SYNCO is active-high

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

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

bit 6-4 **SYNCSRC<2:0>:** Synchronous Source Selection bits(1)
111 = Reserved
•
•
•
010 = Reserved
001 = SYNCI2
000 = SYNCI1

bit 3-0 **SEVTPS<3:0>:** PWM Special Event Trigger Output Postscaler Select bits(1)
1111 = 1:16 postscaler generates Special Event trigger at every 16th compare match event
•
•
•
0001 = 1:2 postscaler generates Special Event trigger at every second compare match event
0000 = 1:1 postscaler generates Special Event trigger at every compare match event

Note 1: These bits should be changed only when PTEN = 0. In addition, when using the SYNCIx feature, the user application must program the period register with a value that is slightly larger than the expected period of the external synchronization input signal.

Register 14-2: PTCON2: PWM Clock Divider Select Register

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	PCLKSEL<2:0>(1)		
bit 7							bit 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

bit 15-3 **Unimplemented:** Read as ‘0’

bit 2-0 **PCLKSEL<2:0>:** PWM Input Clock Prescaler (Divider) Select bits(1)

111 = Reserved
110 = Divide by 64, maximum PWM resolution
101 = Divide by 32, maximum PWM resolution
100 = Divide by 16, maximum PWM resolution
011 = Divide by 8, maximum PWM resolution
010 = Divide by 4, maximum PWM resolution
001 = Divide by 2, maximum PWM resolution
000 = Divide by 1, maximum PWM resolution (power-on default)

Note 1: These bits should be changed only when PTEN = 0. Changing the clock selection during operation will yield unpredictable results.

Register 14-3: PTPER: Primary Master Time Base Period Register

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PTPER<15:8>(1)							
bit 15							bit 8

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0
PTPER<7:0>(1)							
bit 7							bit 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

bit 15-0 **PTPER<15:0>:** Primary Master Time Base (PMTMR) Period Value bits(1)

Note 1: 1 LSB = 1 Tosc. For example, 8.33 ns for 60 MIPS operation.

Register 14-4: STCON: PWM Secondary Master Time Base Control Register

U-0	U-0	U-0	HS/HC-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	SESTAT	SEIEN	EIPU(1)	SYNCPOL	SYNCOEN
bit 15							bit 8

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

Legend:	HC = Cleared in Hardware	HS = Set in Hardware	
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

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

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

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

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

bit 9 **SYNCPOL:** Synchronize Input and Output Polarity bit
1 = SYNCO2 output is active-low
0 = SYNCO2 output is active-high

bit 8 **SYNCOEN:** Secondary Master Time Base Sync Enable bit
1 = SYNCO2 output is enabled
0 = SYNCO2 output is disabled

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

bit 6-4 **SYNCSRC<2:0>:** Secondary Time Base Sync Source Selection bits
111 = Reserved
•
•
•
010 = Reserved
001 = SYNCI2
000 = SYNCI1

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

Note 1: This bit only applies to the secondary master time base period.

Register 14-5: STCON2: PWM Secondary Clock Divider Select Register

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	PCLKSEL<2:0>(1)		
bit 7							bit 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

bit 15-3 **Unimplemented:** Read as '0'

bit 2-0 **PCLKSEL<2:0>:** PWM Input Clock Prescaler (Divider) Select bits(1)
111 = Reserved
110 = Divide by 64, maximum PWM resolution
101 = Divide by 32, maximum PWM resolution
100 = Divide by 16, maximum PWM resolution
011 = Divide by 8, maximum PWM resolution
010 = Divide by 4, maximum PWM resolution
001 = Divide by 2, maximum PWM resolution
000 = Divide by 1, maximum PWM resolution (power-on default)

Note 1: These bits should be changed only when PTEN = 0. Changing the clock selection during operation will yield unpredictable results.

Register 14-6: STPER: Secondary Master Time Base Period Register

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
STPER<15:8>							
bit 15							bit 8

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
STPER<7:0>							
bit 7							bit 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

bit 15-0 **STPER<15:0>:** Secondary Master Time Base Period Value bits

Register 14-7: MDC: PWM Master Duty Cycle Register

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

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

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

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

Register 14-8: SEVTCMP: PWM Special Event Compare Register

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

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<7:0>							
bit 7							bit 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

bit 15-0 **SEVTCMP<15:0>:** Special Event Compare Count Value bits

Register 14-9: SSEVTCMP: PWM Secondary Special Event Compare Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SSEVTCMP<15:8>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SSEVTCMP<7:0>							
bit 7							bit 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

bit 15-0 **SSEVTCMP<15:0>:** Special Event Compare Count Value bits

The optional SSEVTCMP register and the optional secondary master time base provide an additional Special Event Trigger. The secondary special event trigger also has its own postscaler controlled by the SEVTPS<3:0> bits in the STCON register.

Register 14-10: CHOP: PWM Chop Clock Generator Register

R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
CHPCLKEN	—	—	—	—	—	CHOP<9:8>	
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CHOP<7:0>							
bit 7							bit 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

bit 15 **CHPCLKEN:** Enable Chop Clock Generator bit

1 = Chop clock generator is enabled

0 = Chop clock generator is disabled

bit 14-10 **Unimplemented:** Read as '0'

bit 9-0 **CHOP<9:0>:** Chop Clock Divider bits

Chop Frequency = (FP/PLKDIV) / (CHOP<9:0> + 1)

For devices running at 60 MIPS, a value of all zeros will yield a 60 MHz chop clock (period = 16.7 ns) with the PWM clock prescaler configured for fastest clock. A value of 0000000001 in the CHOP<9:0> bits will yield a 30 MHz chop clock with the PWM clock prescaler configured for fastest clock.

Note: The chop clock generator operates with the Primary PWM Clock Prescaler bits (PCLKSEL<2:0>) in the PTCON2 register.

Register 14-11: PWMKEY: PWM Unlock Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PWMKEY<15:8>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PWMKEY<7:0>							
bit 7							bit 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

bit 15-0 **PWMKEY<15:0>:** PWM Unlock bits

If the PWMLOCK Configuration bit is asserted (PWMLOCK = 1), the IOCONx and FCLCONx registers are writable only after the proper sequence is written to the PWMKEY register. If the PWMLOCK Configuration bit is deasserted (PWMLOCK = 0), the IOCONx and FCLCONx registers are writable at all times. Refer to **14.5.3 “Write Protection”** for further details about the unlock sequence.

Note: This register is implemented only in devices where the PWMLOCK Configuration bit is present in the FOSCSEL Configuration register.

Register 14-12: PWMCONx: PWM Control Register

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(1)	CLSTAT(1)	TRGSTAT	FLTIEN	CLIEN	TRGIEN	ITB(3)	MDCS(3)
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
DTC<1:0>		DTCP(5)	—	MTBS	CAM(2,3)	XPRES(4)	IUE(3)
bit 7							bit 0

Legend:	HC = Cleared in Hardware	HS = Set in Hardware	
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

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

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

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 is enabled
0 = Fault interrupt is disabled and the FLTSTAT bit is cleared

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

bit 9 **ITB:** Independent Time Base Mode bit(3)
1 = PHASEx/SPHASEx registers provide time base period for this PWM generator
0 = PTPER register provides timing for this PWM generator

bit 8 **MDCS:** Master Duty Cycle Register Select bit(3)
1 = MDC register provides duty cycle information for this PWM generator
0 = PDCx and SDCx registers provide duty cycle information for this PWM generator

Note 1: Software must clear the interrupt status here, and in the corresponding IFS bit in the Interrupt Controller.
2: The Independent Time Base mode (ITB = 1) must be enabled to use Center-Aligned mode. If ITB = 0, the CAM bit is ignored.
3: These bits should not be changed after the PWM is enabled (PTEN = 1).
4: To operate in External Period Reset mode, the ITB bit must be set to '1' and the CLMOD bit in the FCLCONx register must be set to '0'.
5: For DTCP to be effective, DTC<1:0> must be set to '11'; otherwise, DTCP is ignored.
6: Negative dead time is only implemented for Edge-Aligned mode (CAM = 0).

Register 14-12: PWMCONx: PWM Control Register (Continued)

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

bit 5 **DTCP:** Dead Time Compensation Polarity bit(5)
1 = If DTCMPx pin = 0, PWMxL is shortened, and PWMxH is lengthened
If DTCMPx pin = 1, PWMxH is shortened, and PWMxL is lengthened
0 = If DTCMPx pin = 0, PWMxH is shortened, and PWMxL is lengthened
If DTCMPx pin = 1, PWMxL is shortened, and PWMxH is lengthened

bit 4 **Unimplemented:** Read as '0'

bit 3 **MTBS:** Master Time Base Select bit
1 = PWM generator uses the secondary master time base for synchronization and the clock source for the PWM generation logic (if secondary time base is available)
0 = PWM generator uses the primary master time base for synchronization and the clock source for the PWM generation logic

bit 2 **CAM:** Center-Aligned Mode Enable bit(2,3)
1 = Center-Aligned mode is enabled
0 = Edge-Aligned mode is enabled

bit 1 **XPRES:** External PWM Reset Control bit(4)
1 = Current-limit source resets primary local 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(3)
1 = Updates to the active MDC/PDCx/SDCx/DTx/ALTDTRx/PHASEx/SPHASEx registers are immediate
0 = Updates to the active MDC/PDCx/SDCx/DTx/ALTDTRx/PHASEx/SPHASEx registers are synchronized to the PWM time base

Note 1: Software must clear the interrupt status here, and in the corresponding IFS bit in the Interrupt Controller.
2: The Independent Time Base mode (ITB = 1) must be enabled to use Center-Aligned mode. If ITB = 0, the CAM bit is ignored.
3: These bits should not be changed after the PWM is enabled (PTEN = 1).
4: To operate in External Period Reset mode, the ITB bit must be set to '1' and the CLMOD bit in the FCLCONx register must be set to '0'.
5: For DTCP to be effective, DTC<1:0> must be set to '11'; otherwise, DTCP is ignored.
6: Negative dead time is only implemented for Edge-Aligned mode (CAM = 0).

Register 14-13: IOCONx: PWM I/O Control Register

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

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OVRDAT<1:0>		FLTDAT<1:0>(1,2)		CLDAT<1:0>		SWAP	OSYNC
bit 7							bit 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

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>:** PWM # I/O Pin Mode bits
11 = PWM I/O pin pair is in True Independent PWM Output mode
10 = PWM I/O pin pair is in Push-Pull Output mode
01 = PWM I/O pin pair is in Redundant Output mode
00 = PWM I/O pin pair is in Complementary Output mode

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>:** State(2) for PWMxH, PWMxL Pins if Override is Enabled bits
If OVERENH = 1, OVRDAT<1> provides data for PWMxH
If OVERENL = 1, OVRDAT<0> provides data for PWMxL

bit 5-4 **FLTDAT<1:0>:** State(2) for PWMxH and PWMxL Pins if FLTMOD is Enabled bits(1)

IFLTMOD (FCLCONx<15>) = 0: Normal Fault mode:
If fault is active, FLTDAT<1> provides the state for PWMxH.
If fault is active, FLTDAT<0> provides the state for PWMxL.

IFLTMOD (FCLCONx<15>) = 1: Independent Fault mode:
If current-limit is active, FLTDAT<1> provides the state for PWMxH.
If fault is active, FLTDAT<0> provides the state for PWMxL.

Note 1: These bits must not be changed after the PWM module is enabled (PTEN = 1).

2: State represents Active/Inactive state of the PWM, depending on the POLH and POLL bits. For example, if FLTDAT<1> is set to '1' and POLH is set to '1', the PWMxH pin will be at logic level 0 (active level) when a fault occurs.

Register 14-13: IOCONx: PWM I/O Control Register (Continued)

bit 3-2 **CLDAT<1:0>:** State(2) for PWMxH and PWMxL Pins if CLMOD is Enabled bits

IFLTMOD (FCLCONx<15>) = 0: Normal Fault mode:
If current-limit is active, CLDAT<1> provides the state for PWMxH.
If current-limit is active, CLDAT<0> provides the state for PWMxL.

IFLTMOD (FCLCONx<15>) = 1: Independent Fault mode:
The CLDAT<1:0> bits are ignored.

bit 1 **SWAP:** SWAP PWMxH and PWMxL Pins bit

1 = PWMxH output signal is connected to PWMxL pin; PWMxL output signal is connected to PWMxH pin
0 = PWMxH and PWMxL output signals pins are mapped to their respective pins

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 OVDDAT<1:0> bits occur on next CPU clock boundary

Note 1: These bits must not be changed after the PWM module is enabled (PTEN = 1).

2: State represents Active/Inactive state of the PWM, depending on the POLH and POLL bits. For example, if FLTDAT<1> is set to '1' and POLH is set to '1', the PWMxH pin will be at logic level 0 (active level) when a fault occurs.

Register 14-14: FCLCONx: PWM Fault Current-Limit Control Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IFLTMOD	CLSRC<4:0>(2,3)					CLPOL(1)	CLMOD
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FLTSRC<4:0>(2,3)					FLTPOL(1)	FLTMOD<1:0>	
bit 7							bit 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

bit 15 **IFLTMOD:** Independent Fault Mode Enable bit
1 = Independent Fault mode: Current-limit input maps FLTDAT<1> to PWMxH output, and fault input maps FLTDAT<0> to PWMxL output. The CLDAT<1:0> bits are not used for override functions.
0 = Normal Fault mode: Current-limit and fault inputs map CLDAT<1:0> and FLTDAT<1:0> to PWMxH and PWMxL outputs.

bit 14-10 **CLSRC<4:0>:** Current-Limit Control Signal Source Select bits for PWM Generator #(2,3)
These bits also specify the source for the dead-time compensation input signal, DTCMPx.
11111 = Reserved
•
•
•
01010 = Comparator 3
01001 = Comparator 2
01000 = Comparator 1
00111 = Reserved
00110 = Fault 7
00101 = Fault 6
00100 = Fault 5
00011 = Fault 4
00010 = Fault 3
00001 = Fault 2
00000 = Fault 1

bit 9 **CLPOL:** Current-Limit Polarity bit for PWM Generator #(1)
1 = The selected current-limit source is active-low
0 = The selected current-limit source is active-high

bit 8 **CLMOD:** Current-Limit Mode Enable bit for PWM Generator #
1 = Current-Limit mode is enabled
0 = Current-Limit mode is disabled

Note 1: These bits should be changed only when PTEN = 0. Changing the polarity selection during operation will yield unpredictable results.

2: When Independent Fault mode is enabled (IFLTMOD = 1) and Fault 1 is used for Fault mode (FLTSRC<4:0> = b0000), the Current-Limit Control Source Select bits (CLSRC<4:0>) should be set to an unused current-limit source to prevent the current-limit source from disabling both the PWMxH and PWMxL outputs.

3: When Independent Fault mode is enabled (IFLTMOD = 1), and Fault 1 is used for Current-Limit mode (CLSRC<4:0> = b0000), the Fault Control Source Select bits (FLTSRC<4:0>) should be set to an unused fault source to prevent Fault 1 from disabling both the PWMxL and PWMxH outputs.

Register 14-14: FCLCONx: PWM Fault Current-Limit Control Register (Continued)

bit 7-3 **FLTSRC<4:0>:** Fault Control Signal Source Select bits for PWM Generator #(2,3)
11111 = Reserved
•
•
•
01010 = Comparator 3
01001 = Comparator 2
01000 = Comparator 1
00111 = Reserved
00110 = Fault 7
00101 = Fault 6
00100 = Fault 5
00011 = Fault 4
00010 = Fault 3
00001 = Fault 2
00000 = Fault 1

bit 2 **FLTPOL:** Fault Polarity bit for PWM Generator #(1)
1 = The selected fault source is active-low
0 = The selected fault source is active-high

bit 1-0 **FLTMOD<1:0>:** Fault Mode bits for PWM Generator #
11 = Fault input is disabled
10 = Reserved
01 = The selected fault source forces PWMxH, PWMxL pins to FLTDAT values (cycle)
00 = The selected fault source forces PWMxH, PWMxL pins to FLTDAT values (latched condition)

Note 1: These bits should be changed only when PTEN = 0. Changing the polarity selection during operation will yield unpredictable results.

2: When Independent Fault mode is enabled (IFLTMOD = 1) and Fault 1 is used for Fault mode (FLTSRC<4:0> = b0000), the Current-Limit Control Source Select bits (CLSRC<4:0>) should be set to an unused current-limit source to prevent the current-limit source from disabling both the PWMxH and PWMxL outputs.

3: When Independent Fault mode is enabled (IFLTMOD = 1), and Fault 1 is used for Current-Limit mode (CLSRC<4:0> = b0000), the Fault Control Source Select bits (FLTSRC<4:0>) should be set to an unused fault source to prevent Fault 1 from disabling both the PWMxL and PWMxH outputs.

Register 14-15: PDCx: PWM Generator Duty Cycle Register(1)

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>							
bit 15							bit 8

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>							
bit 7							bit 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

bit 15-0 **PDCx<15:0>:** PWM Generator # Duty Cycle Value bits

Note 1: In Independent PWM mode, PMOD<1:0> (IOCONx<11:10>) = 11, the PDCx register controls the PWMxH duty cycle only. In Complementary, Redundant and Push-Pull PWM modes (PMOD<1:0> (IOCONx<11:0>) = 00, 01, or 10), the PDCx register controls the duty cycle of both the PWMxH and PWMxL.

Register 14-16: PHASEx: PWM Primary Phase Shift Register

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

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<7:0>							
bit 7							bit 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

bit 15-0 **PHASEx<15:0>:** PWM Phase Shift Value or Independent Time Base Period bits for the PWM Generator

Note 1: If the ITB bit = 0 (PWMCONx<9>), the following applies based on the mode of operation:

- Complementary, Redundant and Push-Pull Output mode (PMOD<1:0> (IOCONx<11:10>) = 00, 01, or 10) PHASEx<15:0> = Phase shift value for PWMxH and PWMxL outputs
- True Independent Output mode (PMOD<1:0> (IOCONx<11:10>) = 11) PHASEx<15:0> = Phase shift value for PWMxH only

2: If the ITB bit = 1(PWMCONx<9>), the following applies based on the mode of operation:

- Complementary, Redundant, and Push-Pull Output mode (PMOD<1:0> (IOCONx<11:10>) = 00, 01, or 10) PHASEx<15:0> = Independent time base period value for PWMxH and PWMxL
- True Independent Output mode (PMOD<1:0> (IOCONx<11:10>) = 11) PHASEx<15:0> = Independent time base period for PWMxH only

Register 14-17: SDCx: PWM Secondary Duty Cycle Register(1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SDCx<15:8>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SDCx<7:0>							
bit 7							bit 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

bit 15-0 **SDCx<15:0>:** Secondary Duty Cycle bits for PWMxL output pin

Note 1: The SDCx register is used in Independent PWM mode only (PMOD<1:0> (IOCONx<11:10>) = 11. When used in Independent PWM mode, the SDCx register controls the PWMxL duty cycle.

Register 14-18: SPHASEx: PWM Secondary Phase Shift Register(1,2)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SPHASEx<15:8>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SPHASEx<7:0>							
bit 7							bit 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

bit 15-0 **SPHASEx<15:0>:** Secondary Phase Offset bits for PWMxL Output Pin

Note 1: If the ITB bit = 0 (PWMCONx<9>), the following applies based on the mode of operation:

- Complementary, Redundant and Push-Pull Output mode (PMOD<1:0> (IOCONx<11:10>) = 00, 01, or 10) SPHASEx<15:0> = Not used
- True Independent Output mode (PMOD<1:0> (IOCONx<11:10>) = 11) SPHASEx<15:0> = Phase shift value for PWMxL only

2: If the ITB bit = 1 (PWMCONx<9>), the following applies based on the mode of operation:

- Complementary, Redundant and Push-Pull Output mode (PMOD<1:0> (IOCONx<11:10>) = 00, 01, or 10) SPHASEx<15:0> = Not used
- True Independent Output mode (PMOD<1:0> (IOCONx<11:10>) = 11) SPHASEx<15:0> = Independent time base period value for PWMxL only

Register 14-19: DTRx: PWM Dead Time Register

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

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DTRx<7:0>							
bit 7							bit 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

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

bit 13-0 **DTRx<13:0>:** Unsigned 14-bit Dead Time Value bits for PWMx Dead Time Unit

Register 14-20: ALTDTRx: PWM Alternate Dead Time Register

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

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ALTDTRx<7:0>							
bit 7							bit 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

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-0 **ALTDTRx<13:0>:** Unsigned 14-bit Dead Time Value bits for PWMx Dead Time Unit

Register 14-21: TRIGx: PWM Primary Trigger Compare Value Register

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

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<7:0>							
bit 7							bit 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

bit 15-0 **TRGCMP<15:0>:** Trigger Control Value bits

When the primary PWM functions in local time base, this register contains the compare values that can trigger the ADC module.

Register 14-22: TRGCONx: PWM Trigger Control Register

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
TRGDIV<3:0>				—	—	—	—
bit 15							bit 8

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

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

bit 15-12 **TRGDIV<3:0>:** Trigger # Output Divider bits

1111 = Trigger output for every 16th trigger event
1110 = Trigger output for every 15th trigger event
1101 = Trigger output for every 14th trigger event
1100 = Trigger output for every 13th trigger event
1011 = Trigger output for every 12th trigger event
1010 = Trigger output for every 11th trigger event
1001 = Trigger output for every 10th trigger event
1000 = Trigger output for every 9th trigger event
0111 = Trigger output for every 8th trigger event
0110 = Trigger output for every 7th trigger event
0101 = Trigger output for every 6th trigger event
0100 = Trigger output for every 5th trigger event
0011 = Trigger output for every 4th trigger event
0010 = Trigger output for every 3rd trigger event
0001 = Trigger output for every 2nd trigger event
0000 = Trigger output for every trigger event

bit 11-6 **Unimplemented:** Read as '0'

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

111111 = Wait 63 PWM cycles before generating the first trigger event after the module is enabled
•
•
•
000010 = Wait 2 PWM cycles before generating the first trigger event after the module is enabled
000001 = Wait 1 PWM cycles before generating the first trigger event after the module is enabled
000000 = Wait 0 PWM cycles before generating the first trigger event after the module is enabled

Register 14-23: LEBCONx: Leading-Edge Blanking Control Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
PHR	PHF	PLR	PLF	FLTLEBEN	CLLEBEN	—	—
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	BCH(1)	BCL(1)	BPHH	BPHL	BPLH	BPLL
bit 7							bit 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

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 9-6 **Unimplemented:** Read as '0'

bit 5 **BCH:** Blanking in Selected Blanking Signal High Enable bit(1)
1 = State blanking (of current-limit and/or fault input signals) when selected blanking signal is high
0 = No blanking when selected blanking signal is high

bit 4 **BCL:** Blanking in Selected Blanking Signal Low Enable bit(1)
1 = State blanking (of current-limit and/or fault input signals) when selected blanking signal is low
0 = No blanking when selected blanking signal is low

bit 3 **BPHH:** Blanking in PWMxH High Enable bit
1 = State blanking (of current-limit and/or fault input signals) when PWMxH output is high
0 = No blanking when PWMxH output is high

bit 2 **BPHL:** Blanking in PWMxH Low Enable bit
1 = State blanking (of current-limit and/or fault input signals) when PWMxH output is low
0 = No blanking when PWMxH output is low

bit 1 **BPLH:** Blanking in PWMxL High Enable bit
1 = State blanking (of current-limit and/or fault input signals) when PWMxL output is high
0 = No blanking when PWMxL output is high

bit 0 **BPLL:** Blanking in PWMxL Low Enable bit
1 = State blanking (of current-limit and/or fault input signals) when PWMxL output is low
0 = No blanking when PWMxL output is low

Note 1: The blanking signal is selected via the BLANKSEL<3:0> bits (AUXCONx<11:8>).

Register 14-24: LEBDLYx: Leading-Edge Blanking Delay Register

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	LEB<11:8>			
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
LEB<7:0>							
bit 7							bit 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

bit 15-12 **Unimplemented:** Read as ‘0’

bit 11-0 **LEB<11:0>:** Leading-Edge Blanking Delay bits for Current-Limit and Fault Inputs

Register 14-25: PWMCAPx: Primary PWM Time Base Capture Register

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PWMCAP<15:8>(1,2)							
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PWMCAP<7:0>(1,2)							
bit 7							bit 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

bit 15-0 **PWMCAP<15:0>:** Captured PWM Time Base Value bits(1,2)

The value in this register represents the captured PWM time base value when a leading edge is detected on the current-limit input.

Note 1: The capture feature is only available on the primary output (PWMxH).

2: The feature is only active after LEB processing on the current-limit input signal is complete.

Register 14-26: AUXCONx: PWM Auxiliary Control Register

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	BLANKSEL<3:0>			
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CHOPSEL<3:0>				CHOPHEN	CHOPLEN
bit 7							bit 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

bit 15-12 **Unimplemented:** Read as '0'

bit 11-8 **BLANKSEL<3:0>:** PWM State Blank Source Select bits

The selected state blank signal blocks the current-limit and/or fault input signals (if enabled via the BCH and BCL bits in the LEBCONx register).

1111 = Reserved
•
•
•
1000 = Reserved
0111 = PWM7H selected as state blank source
0110 = PWM6H selected as state blank source
0101 = PWM5H selected as state blank source
0100 = PWM4H selected as state blank source
0011 = PWM3H selected as state blank source
0010 = PWM2H selected as state blank source
0001 = PWM1H selected as state blank source
0000 = No state blanking

bit 7-6 **Unimplemented:** Read as '0'

bit 5-2 **CHOPSEL<3:0>:** PWM Chop Clock Source Select bits

The selected signal will enable and disable (CHOP) the selected PWM outputs.

1111 = Reserved
•
•
•
1000 = Reserved
0111 = PWM7H selected as CHOP clock source
0110 = PWM6H selected as CHOP clock source
0101 = PWM5H selected as CHOP clock source
0100 = PWM4H selected as CHOP clock source
0011 = PWM3H selected as CHOP clock source
0010 = PWM2H selected as CHOP clock source
0001 = PWM1H selected as CHOP clock source
0000 = Chop clock generator selected as CHOP clock source

bit 1 **CHOPHEN:** PWMxH Output Chopping Enable bit

1 = PWMxH chopping function is enabled
0 = PWMxH chopping function is disabled

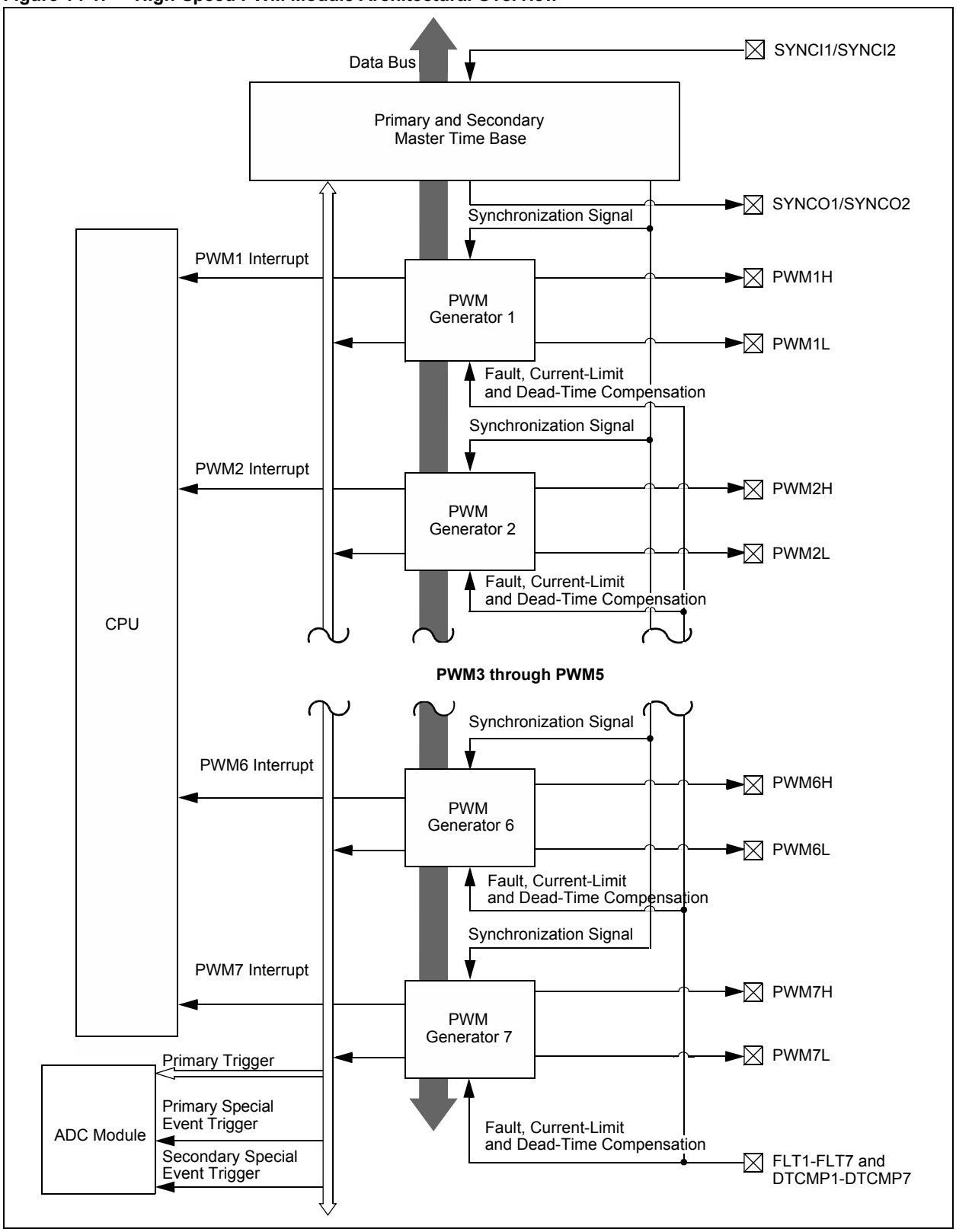
bit 0 **CHOPLEN:** PWMxL Output Chopping Enable bit

1 = PWMxL chopping function is enabled
0 = PWMxL chopping function is disabled

14.4 ARCHITECTURE OVERVIEW

Figure 14-1 illustrates the architectural overview of the High-Speed PWM module and its interconnection with the CPU and other peripherals.

Figure 14-1: High-Speed PWM Module Architectural Overview



The High-Speed PWM module contains up to seven PWM generators. Each PWM generator provides two PWM outputs: PWMxH and PWMxL. A master time base generator provides a synchronous signal as a common time base to synchronize the various PWM outputs. Each generator can operate independently or in sync with the master time base. The individual PWM outputs are available on the output pins of the device. The input fault signals and current-limit signals, when enabled, monitor and protect the system by placing the PWM outputs into a known “safe” state.

Each PWM generator can create a trigger to the ADC module to sample the analog signal at a specific instance during the PWM period. In addition, the High-Speed PWM module also generates a Special Event Trigger to the ADC module based on the master time base.

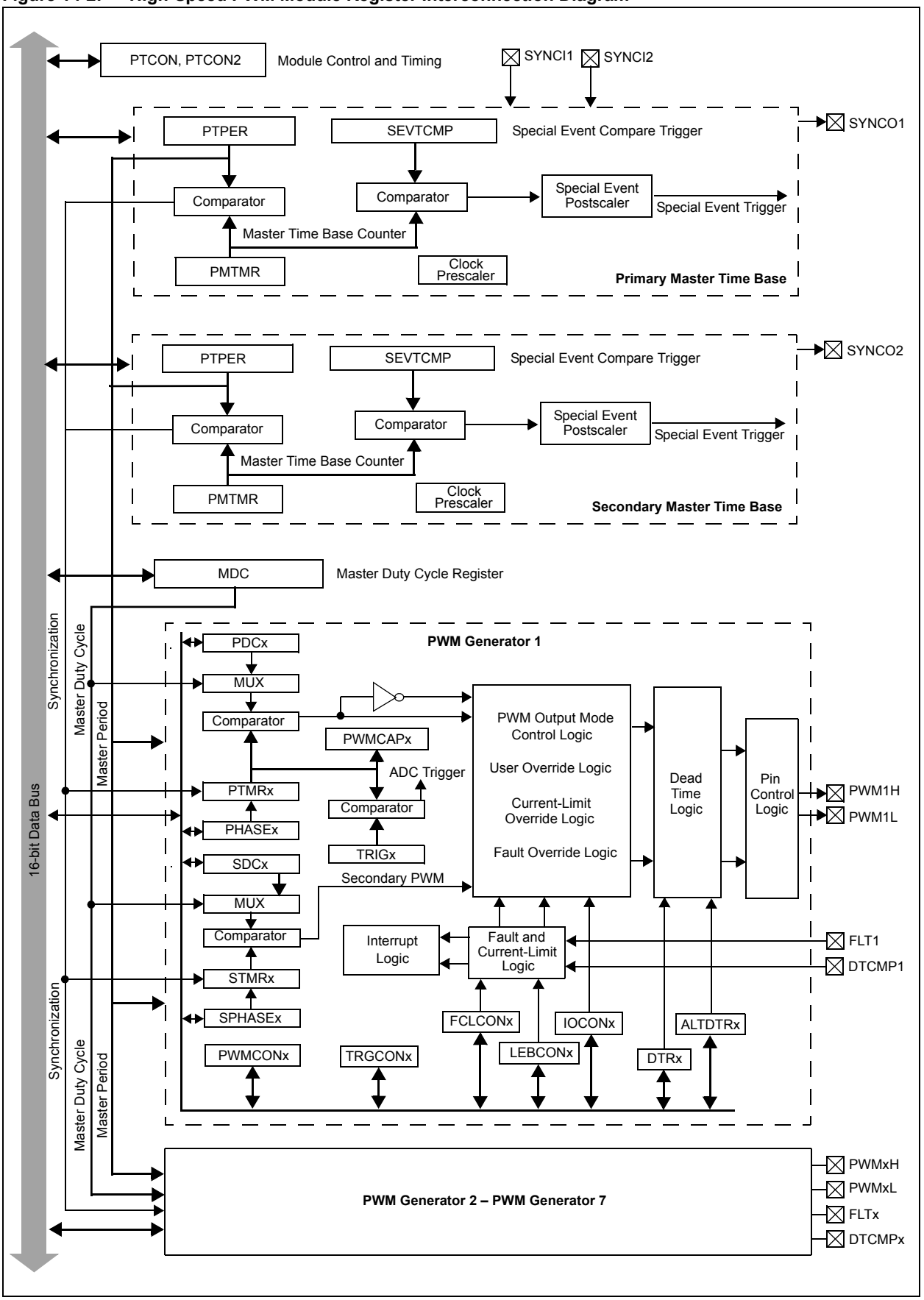
The High-Speed PWM module can synchronize itself with an external signal or can act as a synchronizing source to any external device. The SYNCI1 and SYNCI2 pins are the input pins, which can synchronize the High-Speed PWM module with an external signal. The SYNCO pin is an output pin that provides a synchronous signal to an external device.

The High-Speed PWM module can be used for a wide variety of power conversion/motor control applications that require:

- High operating frequencies with good resolution
- Ability to dynamically control PWM parameters such as duty cycle, period and dead time
- Ability to independently control each PWM
- Ability to synchronously control all PWMs
- Independent resource allocation for each PWM generator
- Fault handling capability
- CPU load staggering to execute multiple control loops

Each function of the High-Speed PWM module is described in detail in subsequent sections. Figure 14-2 illustrates the interconnections between various registers in the High-Speed PWM module.

Figure 14-2: High-Speed PWM Module Register Interconnection Diagram



14.5 MODULE DESCRIPTION

14.5.1 PWM Clock Selection

The system clock is used to generate the clock for the High-Speed PWM module internally. The maximum time resolution for this module is 8.33 ns when operating at 60 MIPS, since the PWM module is clocked at FOSC = FCY * 2 (FCY Max = 60 MIPS).

14.5.2 Time Base

Each PWM output in a PWM generator can use the master time base or an independent time base. The input clock of the High-Speed PWM module has prescaler (divider) options of 1:1 to 1:64, which can be selected using the PWM Input Clock Prescaler (Divider) Select bits (PCLKSEL) in the PWM Clock Divider Select register (PTCON2<2:0>). The prescaled value will also reflect the PWM resolution, which helps to reduce the power consumption of the High-Speed PWM module. The prescaled clock is the input to the PWM clock control logic block. The maximum clock rate provides a duty cycle and period resolution of TOSC.

For example:

- If a prescaler option of 1:2 is selected, the PWM duty cycle and period resolution can be set at TOSC * 2. Thereby, the power consumption of the High-Speed PWM module would be reduced by approximately 50 percent of the maximum speed operation.
- If a prescaler option of 1:4 is selected, the PWM duty cycle and period resolution can be set at TOSC * 4. Thereby, the power consumption of the High-Speed PWM module would be reduced by approximately 75 percent of the maximum speed operation.

The High-Speed PWM module can operate in the standard edge-aligned or center-aligned time base.

14.5.3 Write Protection

Certain devices incorporate a write protection feature for the IOCONx and FCLCONx registers, which prevents any inadvertent writes to these registers. This feature can be controlled by the PWMLOCK Configuration bit (FOSCSEL<6>). The default state of the write protection feature is enabled (PWMLOCK = 1). Refer to the specific device data sheet for more information regarding the Configuration bits.

To gain write access to the locked registers, the user application must write two consecutive values of 0xABCD and 0x4321 to the PWMKEY register. The write access to the IOCONx or FCLCONx registers must be the next SFR access following the unlock sequence; there can be no other SFR accesses during the unlock process and subsequent write access. To write to both the IOCONx and FCLCONx registers requires two unlock operations.

The correct unlocking sequence is described in Example 14-1.

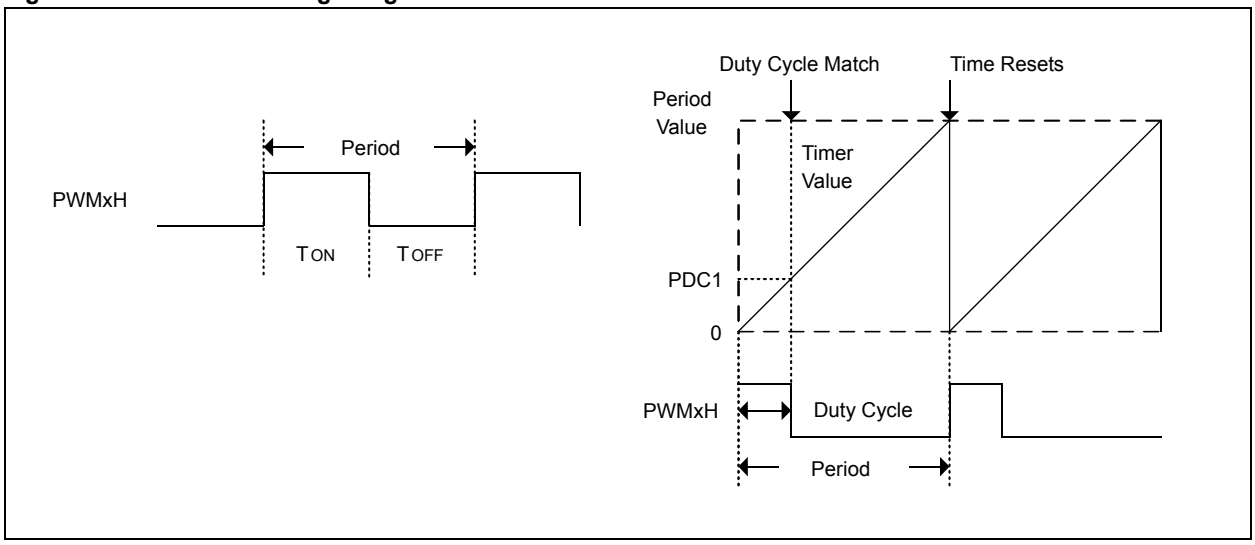
Example 14-1: PWM Write-Protected Register Unlock Sequence

```
; FLT32 pin must be pulled high externally to clear and disable the fault
; Writing to FCLCON1 register requires unlock sequence
mov #0xabcd,w10        ;Load first unlock key to w10 register
mov #0x4321,w11        ;Load second unlock key to w11 register
mov #0x0000,w0         ;Load desired value of FCLCON1 register in w0
mov w10, PWMKEY        ;Write first unlock key to PWMKEY register
mov w11, PWMKEY        ;Write second unlock key to PWMKEY register
mov w0,FCLCON1         ;Write desired value to FCLCON1 register

; Set PWM ownership and polarity using the IOCON1 register
; Writing to IOCON1 register requires unlock sequence
mov #0xabcd,w10        ;Load first unlock key to w10 register
mov #0x4321,w11        ;Load second unlock key to w11 register
mov #0xF000,w0         ;Load desired value of IOCON1 register in w0
mov w10, PWMKEY        ;Write first unlock key to PWMKEY register
mov w11, PWMKEY        ;Write second unlock key to PWMKEY register
mov w0,IOCON1          ;Write desired value to IOCON1 register
```

14.5.4 Standard Edge-Aligned PWM

The standard edge-aligned PWM waveforms are illustrated in Figure 14-3. To create the edge-aligned PWM, a timer or counter circuit counts upward from zero to a specified maximum value, called Period. Another register contains the duty cycle value, which is constantly compared with the timer value. When the timer or 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 value is greater than or equal to the period value, the timer resets itself and the process repeats.

Figure 14-3: Standard Edge-Aligned PWM Mode



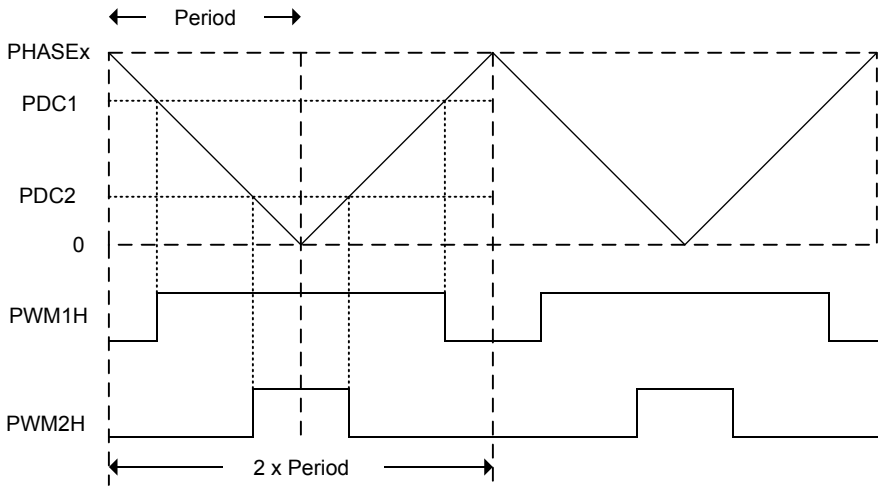
14.5.5 Center-Aligned PWM

The center-aligned PWM waveforms illustrated in Figure 14-4, align the PWM signals with respect to a reference point so that half of the PWM signal occurs before the reference point and the remaining half of the signal occurs after the reference point. The Center-Aligned mode is enabled when the CAM bit (PWMCONx<2>) is set.

When operating in Center-Aligned mode, the effective PWM period is twice the value specified in the PHASEx registers, because the independent time base counter in the PWM generator is counting up and then counting down during the cycle. The up/down count sequence doubles the effective PWM cycle period. This mode is used in many motor control applications.

Note:	The Independent Time Base mode (ITB = 1) must be enabled to use the Center-Aligned mode. If ITB = 0, the CAM bit is ignored.

Figure 14-4: Center-Aligned PWM Mode



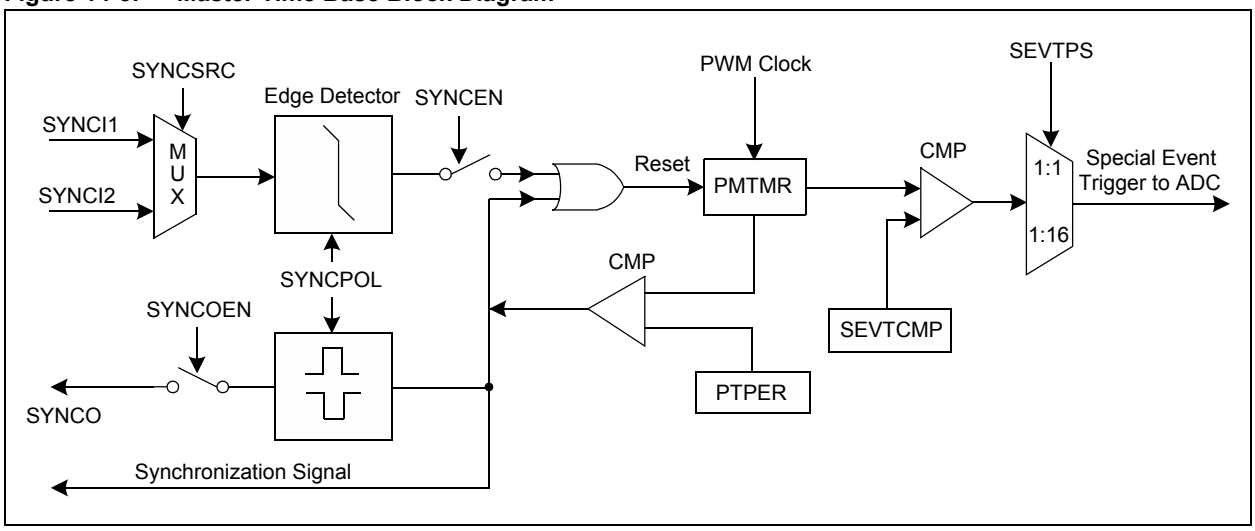
Example 14-2: Edge-Aligned or Center-Aligned Mode Selection

```
/* Select Edge-Aligned PWM Time Base */

PWMCON1bits.CAM = 0;       /* For Edge-Aligned mode */

/* Select Center-Aligned PWM Time Base */

PWMCON1bits.CAM = 1;       /* For Center-Aligned mode */
PWMCON1bits.ITB = 1;       /* Must be set for Center-Aligned mode */
```

14.5.6 Master Time Base/Synchronous Time Base

Figure 14-5 illustrates the PWM functionality in the master time base.

Figure 14-5: Master Time Base Block Diagram



Some of the common tasks of the master time base are as follows:

- Generates time reference for all the PWM generators
- Generates special event ADC trigger and interrupt
- Supports synchronization with the external SYNC signal (SYNCI1/SYNCI2)
- Supports synchronization with external devices using SYNCO signal

The master time base for a PWM generator is set by loading a 16-bit value into the Primary Master Time Base Period register (PTPER). In Master Time Base mode, the value in the PHASEx and SPHASEx registers provides phase shift between the PWM outputs. The clock for the PWM timer (PMTMR) is derived from the system clock.

14.5.7 Time Base Synchronization

The master time base can be synchronized with the external synchronization signal via the master time base synchronization signal (SYNCI1/SYNCI2). The synchronization source (SYNCI1 and SYNCI2) can be selected using the SYNCSRC<1:0> bits (PTCON<5:4>). The SYNCPOL bit (PTCON<9>) selects the rising or falling edge of the synchronization pulse, which resets the timer (PMTMR). The external synchronization feature can be enabled or disabled with the SYNCEN bit (PTCON<7>). The pulse-width of the external synchronization signal (SYNCI1/SYNCI2) must be more than the period of the post-scaled input clock to ensure reliable detection by the master time base.

An external device can also be synchronized with the master time base using the Synchronization Output (SYNCO) signal. The SYNCO signal is generated when the PTPER register resets the PMTMR register. The SYNCO signal pulse is 130 ns to ensure other devices reliably sense the signals. The polarity of the SYNCO signal is determined by the SYNCPOL bit (PTCON<9>). The SYNCO signal can be enabled or disabled by selecting the SYNCOEN bit (PTCON<8>).

Note 1: The SYNCI pulse period should be smaller than the PWM period value.

2: The SYNCI pulse should be continuous with a minimum pulse-width of 200 ns.

3: The PWM cycles are expected to be distorted for the first two SYNCI pulses.

4: The period value should be a multiple of 8 (Least Significant 3 bits set to '0') for the external synchronization to work in the Push-Pull mode.

5: There is a delay from the input of a SYNC signal until the internal time base counter is Reset. This will be approximately 2.5 * prescale clock period.

6: The External Time Base Synchronization must not be used with phase shifted PWM as the synchronization signal may not maintain the phase relationships between the multiple PWM channels.

7: The External Time Base Synchronization can not be used in Independent Time Base mode.

The advantage of synchronization is that it ensures the beat frequencies are not generated when multiple power controllers are in use.

Example 14-3: Synchronizing Master Time Base with External Signal

```
/* Synchronizing Master Time Base with External Signal */

PTCONbits.SYNCSRC = 0;     /* Select SYNC1 input as synchronizing source */
PTCONbits.SYNCPOL = 0;     /* Rising edge of SYNC1 resets the PWM Timer */
PTCONbits.SYNCEN = 1;      /* Enable external synchronization */
```

Example 14-4: Synchronizing External Device with Master Time Base

```
/* Synchronizing External Device with Master Time Base */

PTCONbits.SYNCPOL = 0;     /* SYNCO output is active-high */
PTCONbits.SYNCOEN = 1;     /* Enable SYNCO output */
```

14.5.8 Special Event Trigger

The High-Speed PWM module has a master Special Event Trigger that can be used for synchronization of analog-to-digital conversions with the PWM time base. The analog-to-digital sampling and conversion time can be programmed to occur within the PWM period. The Special Event Trigger allows the user-assigned application to minimize the delay between the time the conversion results are acquired and the time the duty cycle value is updated. The Special Event Trigger is based on the master time base.

The master Special Event Trigger value is loaded into the PWM Special Event Compare register (SEVTCMP). In addition, the SEVTPS<3:0> bits (PTCON<3:0>), control the Special Event Trigger operation. To generate a trigger to the ADC module, the value in the PTPER register is compared with the value in the SEVTCMP register. The master Special Event Trigger has a postscaler that allows a 1:1 to 1:16 postscaler ratio. The postscaler is configured by writing to the SEVTPS<3:0> bits (PTCON<3:0>).

The Special Event Trigger pulses are always generated during the following instances:

- On a match condition regardless of the status of the Special Event Interrupt Enable bit (SEIEN)
- If the compare value in the SEVTCMP register is a value from zero to a maximum value of the PTPER register

The Special Event Trigger output postscaler is cleared on these events:

- Any device Reset
- When PTEN = 0

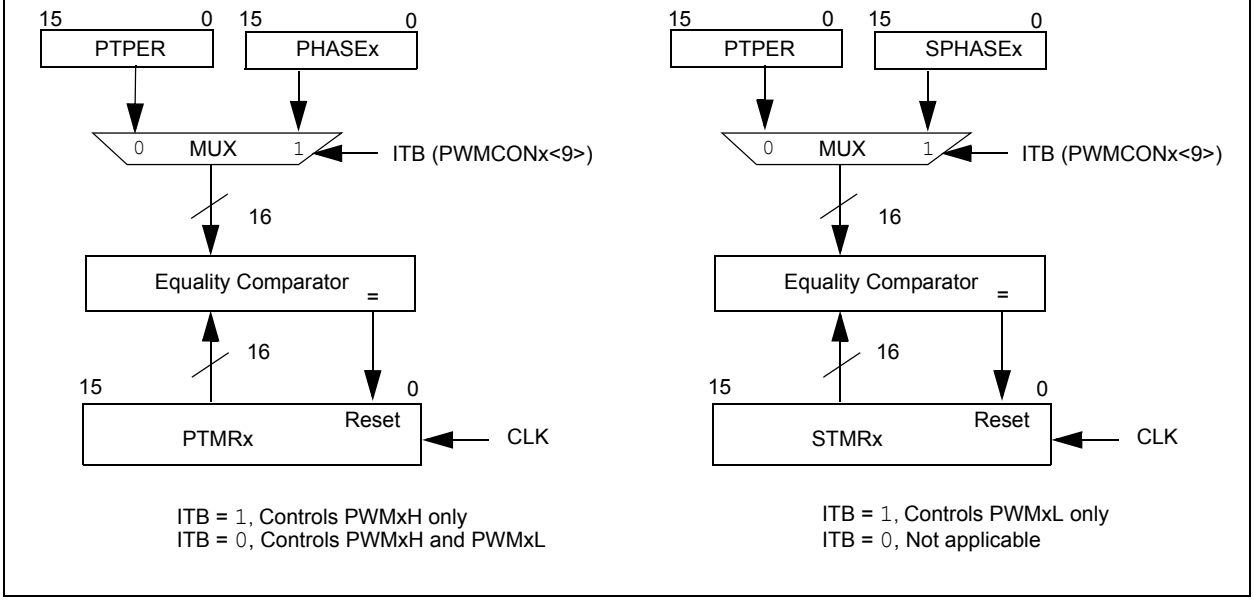
Example 14-5: ADC Special Event Trigger Configuration

```
/* ADC Special Event Trigger Configuration */

SEVTCMP = 1248;           /* Special Event Trigger value set at 25% of the period value (4999)*/

PTCONbits.SEVTPS = 0;     /* Special Event Trigger output postscaler set to 1:1 selection */

PTCONbits.SEIEN = 1;      /* Special event interrupt is enabled */

while (PTCONbits.SESTAT == 1);    /* Wait for special event interrupt status change */
```

14.5.9 Independent PWM Time Base

Figure 14-6 illustrates the PWM functionality in the independent time base.

Figure 14-6: Independent Time Base Block Diagrams



Each PWM generator can operate on:

- A shared time base for both the primary (PWMxH) and secondary (PWMxL) outputs

  The independent time base periods for both PWM outputs (PWMxH and PWMxL) are provided by the value in PWM Primary Phase Shift register (PHASEx).

- A dedicated time base for each of the primary (PWMxH) and secondary (PWMxL) outputs

  The independent time base period for PWMxH output is provided by the value in the PWM Primary Phase Shift register (PHASEx). The independent time base period for PWMxL output is provided by the value in the PWM Secondary Phase Shift (SPHASEx) register.

The PHASEx and SPHASEx registers provide the time period value for the PWMx outputs (PWMxH and PWMxL) in Independent Time Base mode.

Note:	The PTMRx and STMRx values are not readable to the user-assigned application.

14.6 PWM OPERATING MODES

The High-Speed PWM module supports the following operation modes:

- Push-Pull Output mode
- Complementary Output mode
- Redundant Output mode
- Independent Output mode

These operating modes can be selected using the PMOD<1:0> bits (IOCONx<11:10>).

In the following sections, figures and examples are provided, which show the PWM outputs in multiple operating modes. Table 14-1 provides a list of the available modes and settings, with references to the figures by number.

Table 14-1: Mode and Code Cross-reference Table

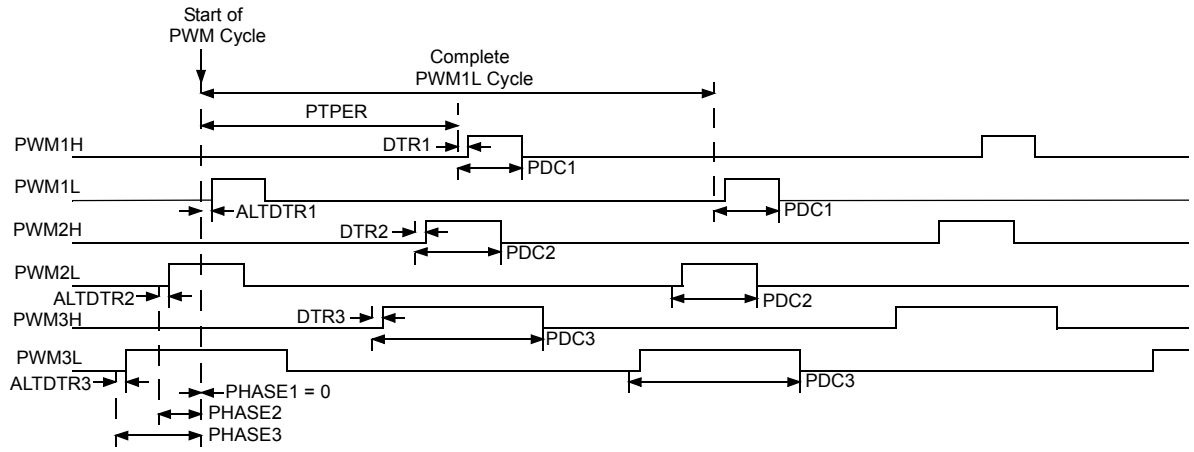
PWM Mode	Mode Settings	Related Figure
Push-Pull	Independent Duty Cycle and Phase, Fixed Primary Period, Edge-Aligned	14-7
	Independent Duty Cycle and Phase, Fixed Secondary Period, Edge-Aligned	14-8
	Master Duty Cycle and Independent Phase, Fixed Primary Period, Edge-Aligned	14-9
	Master Duty Cycle and Independent Phase, Fixed Secondary Period, Edge-Aligned	14-10
	Independent Duty Cycles and Periods, No Phase-Shifting, Edge-Aligned	14-11
	Master Duty Cycles and Independent Periods, No Phase-Shifting, Edge-Aligned	14-12
	Independent Duty Cycles and Periods, No Phase-Shifting, Center-Aligned Mode	14-13
	Master Duty Cycles and Independent Periods, No Phase-Shifting, Center-Aligned Mode	14-14
Complementary	Independent Duty Cycle and Phase, Fixed Primary Period, Edge-Aligned	14-15
	Independent Duty Cycle and Phase, Fixed Secondary Period, Edge-Aligned	14-16
	Master Duty Cycle and Independent Phase, Fixed Primary Period, Edge-Aligned	14-17
	Master Duty Cycle and Independent Phase, Fixed Secondary Period, Edge-Aligned	14-18
	Independent Duty Cycles and Periods, No Phase-Shifting, Edge-Aligned	14-19
	Master Duty Cycles and Independent Periods, No Phase-Shifting, Edge-Aligned	14-20
	Independent Duty Cycles and Periods, No Phase-Shifting, Center-Aligned	14-21
	Master Duty Cycles and Independent Periods, No Phase-Shifting, Center-Aligned	14-22
Redundant	Independent Duty Cycle and Phase, Fixed Primary Period, Edge-Aligned	14-23
	Independent Duty Cycle and Phase, Fixed Secondary Period, Edge-Aligned	14-24
	Master Duty Cycle and Variable Phase, Fixed Primary Period, Edge-Aligned	14-25
	Master Duty Cycle and Variable Phase, Fixed Secondary Period, Edge-Aligned	14-26
	Independent Duty Cycles and Periods, No Phase-Shifting, Edge-Aligned	14-27
	Master Duty Cycles and Independent Periods, No Phase-Shifting, Edge-Aligned	14-28
	Independent Duty Cycles and Periods, No Phase-Shifting, Center-Aligned	14-29
	Master Duty Cycles and Independent Periods, No Phase-Shifting, Center-Aligned	14-30
Independent	Independent Duty Cycle and Phase, Fixed Primary Period, Edge-Aligned	14-31
	Independent Duty Cycle and Phase, Fixed Secondary Period, Edge-Aligned	14-32
	Master Duty Cycle, Variable Phase, Fixed Primary Period, Edge-Aligned	14-33
	Master Duty Cycle, Variable Phase, Fixed Secondary Period, Edge-Aligned	14-34
	Independent Duty Cycles and Periods, No Phase-Shifting, Edge-Aligned	14-35
	Master Duty Cycles, Independent Periods, No Phase-Shifting, Edge-Aligned	14-36
	Independent Duty Cycles and Periods, No Phase-Shifting, Center-Aligned	14-37
	Master Duty Cycles, Independent Periods, No Phase-Shifting, Center-Aligned	14-38

14.6.1 Push-Pull PWM Mode

In Push-Pull mode, the PWM outputs are alternately available on the PWMxH and PWMxL pins. Some typical applications of Push-Pull mode are provided in **14.16 "Application Information"**.

Figure 14-7 through Figure 14-14 and Example 14-6 through Example 14-13 show the PWM outputs for Push-Pull PWM mode in multiple operating modes.

Figure 14-7: Push-Pull PWM Mode - Independent Duty Cycle and Phase, Fixed Primary Period, Edge-Aligned



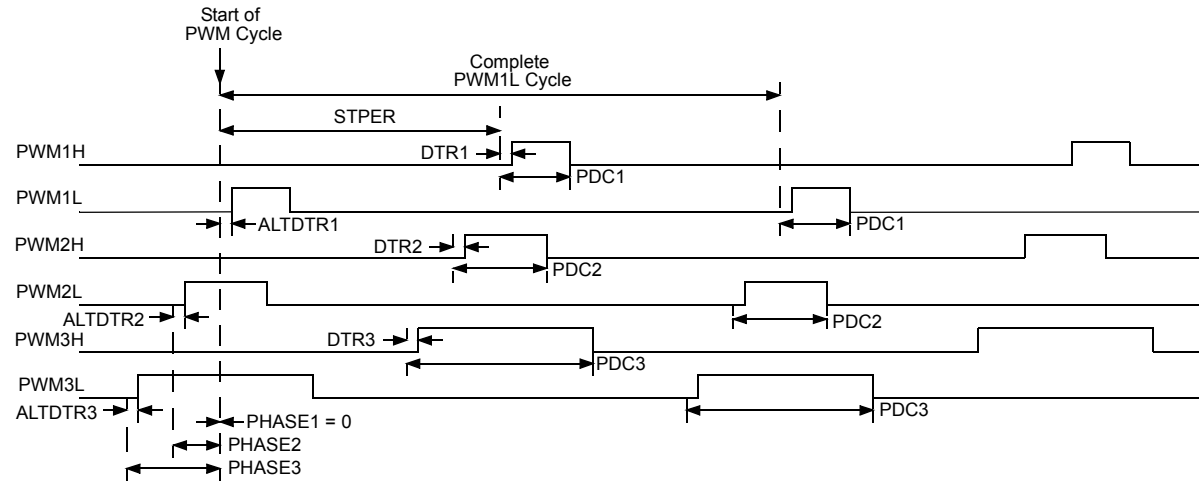
Where:

PHASEx	Phase of PWMxH and PWMxL
PDCx	Duty Cycle of PWMxH and PWMxL
PTPER	Period of PWMxH and PWMxL
DTRx	Dead Time for PWMxH Rising Edge
ALTDTRx	Dead Time for PWMxL Rising Edge

Example 14-6: Push-Pull PWM Mode - Independent Duty Cycle and Phase, Fixed Primary Period, Edge-Aligned

```
/* Set PWM Period on Primary Time Base */
PTPER = 1000;

/* Set Phase Shift */
PHASE1 = 0;
PHASE2 = 100;
PHASE3 = 200;

/* Set Duty Cycles */
PDC1 = 150;
PDC2 = 200;
PDC3 = 400;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 25;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 25;

/* Set PWM Mode to Push-Pull */
IOCON1 = IOCON2 = IOCON3 = 0xC800;

/* Set Primary Time Base, Edge-Aligned Mode and Independent Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0000;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

Figure 14-8: Push-Pull PWM Mode - Independent Duty Cycle and Phase, Fixed Secondary Period, Edge-Aligned



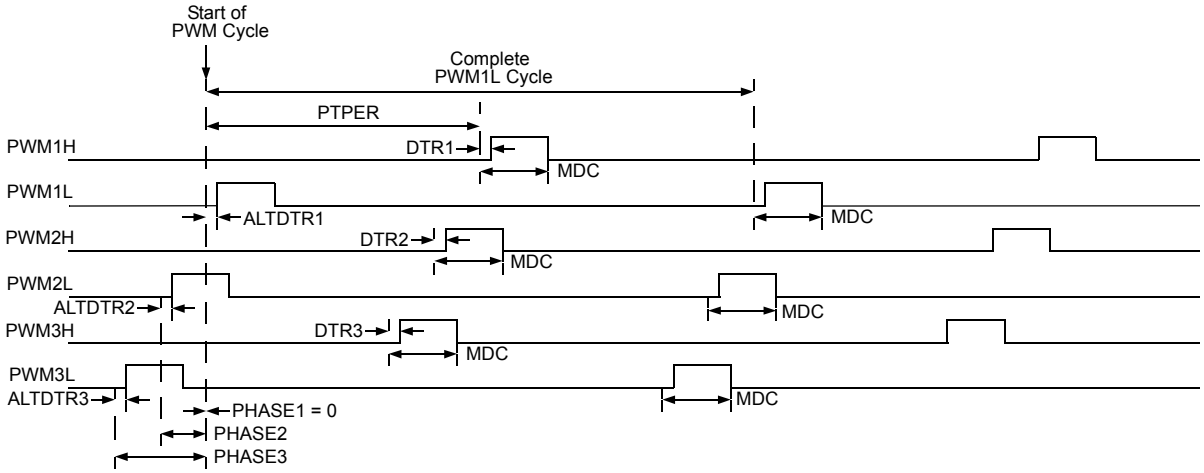
Where:

PHASEx	Phase of PWMxH and PWMxL
PDCx	Duty Cycle of PWMxH and PWMxL
STPER	Period of PWMxH and PWMxL
DTRx	Dead Time for PWMxH Rising Edge
ALTDTRx	Dead Time for PWMxL Rising Edge

Example 14-7: Push-Pull PWM Mode - Independent Duty Cycle and Phase, Fixed Secondary Period, Edge-Aligned

```
/* Set PWM Period on Secondary Time Base */
STPER = 1000;

/* Set Phase Shift */
PHASE1 = 0;
PHASE2 = 100;
PHASE3 = 200;

/* Set Duty Cycles */
PDC1 = 150;
PDC2 = 200;
PDC3 = 400;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 25;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 25;

/* Set PWM Mode to Push-Pull */
IOCON1 = IOCON2 = IOCON3 = 0xC800;

/* Set Secondary Time Base, Edge-Aligned Mode and Independent Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0008;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

Figure 14-9: Push-Pull PWM Mode - Master Duty Cycle and Independent Phase, Fixed Primary Period, Edge-Aligned



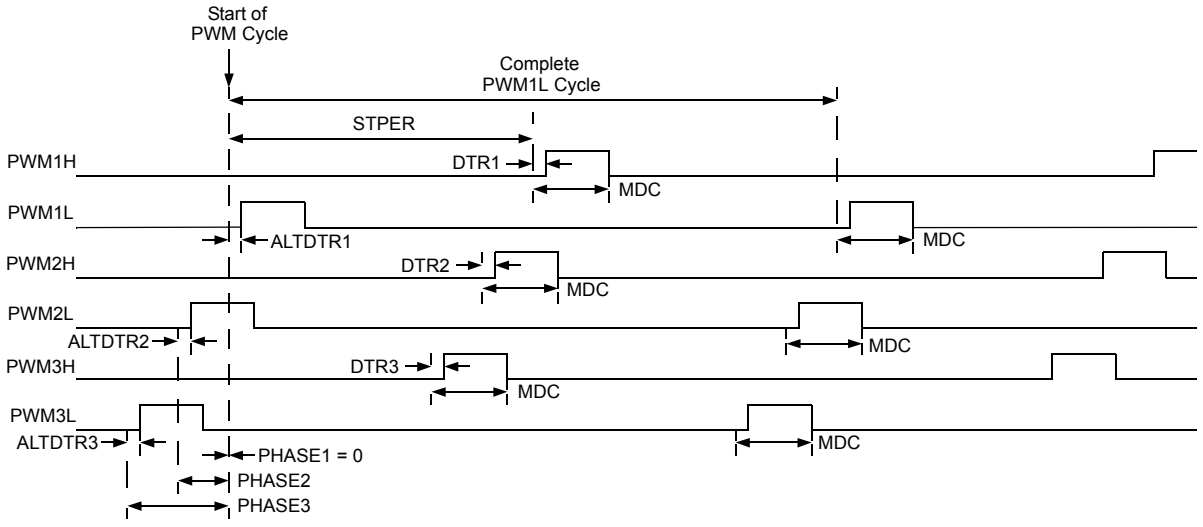
Where:

PHASEx	Phase of PWMxH and PWMxL
MDC	Duty Cycle of PWMxH and PWMxL
PTPER	Period of PWMxH and PWMxL
DTRx	Dead Time for PWMxH Rising Edge
ALTDTRx	Dead Time for PWMxL Rising Edge

Example 14-8: Push-Pull PWM Mode - Master Duty Cycle and Independent Phase, Fixed Primary Period, Edge-Aligned

```
/* Set PWM Period on Primary Time Base*/
PTPER = 1000;

/* Set Phase Shift */
PHASE1 = 0;
PHASE2 = 100;
PHASE3 = 200;

/* Set Duty Cycles */
MDC = 200;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 25;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 25;

/* Set PWM Mode to Push-Pull */
IOCON1 = IOCON2 = IOCON3 = 0xC800;

/* Set Primary Time Base, Edge-Aligned Mode and Master Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0100;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

Figure 14-10: Push-Pull PWM Mode - Master Duty Cycle and Independent Phase, Fixed Secondary Period, Edge-Aligned



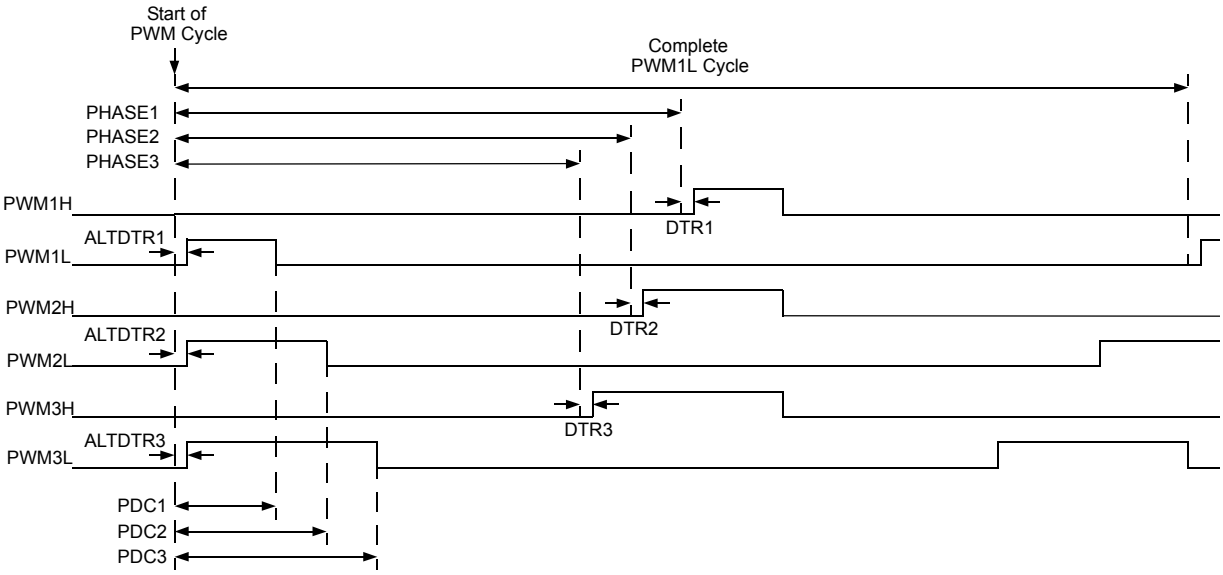
Where:

PHASEx	Phase of PWMxH and PWMxL
MDC	Duty Cycle of PWMxH and PWMxL
STPER	Period of PWMxH and PWMxL
DTRx	Dead Time for PWMxH Rising Edge
ALTDTRx	Dead Time for PWMxL Rising Edge

Example 14-9: Push-Pull PWM Mode - Master Duty Cycle and Independent Phase, Fixed Secondary Period, Edge-Aligned

```
/* Set PWM Period on Secondary Time Base*/
STPER = 1000;

/* Set Phase Shift */
PHASE1 = 0;
PHASE2 = 100;
PHASE3 = 200;

/* Set Duty Cycles */
MDC = 200;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 25;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 25;

/* Set PWM Mode to Push-Pull */
IOCON1 = IOCON2 = IOCON3 = 0xC800;

/* Set Secondary Time Base, Edge-Aligned Mode and Master Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0108;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

Figure 14-11: Push-Pull PWM Mode - Independent Duty Cycles and Independent Periods, No Phase-Shifting, Edge-Aligned



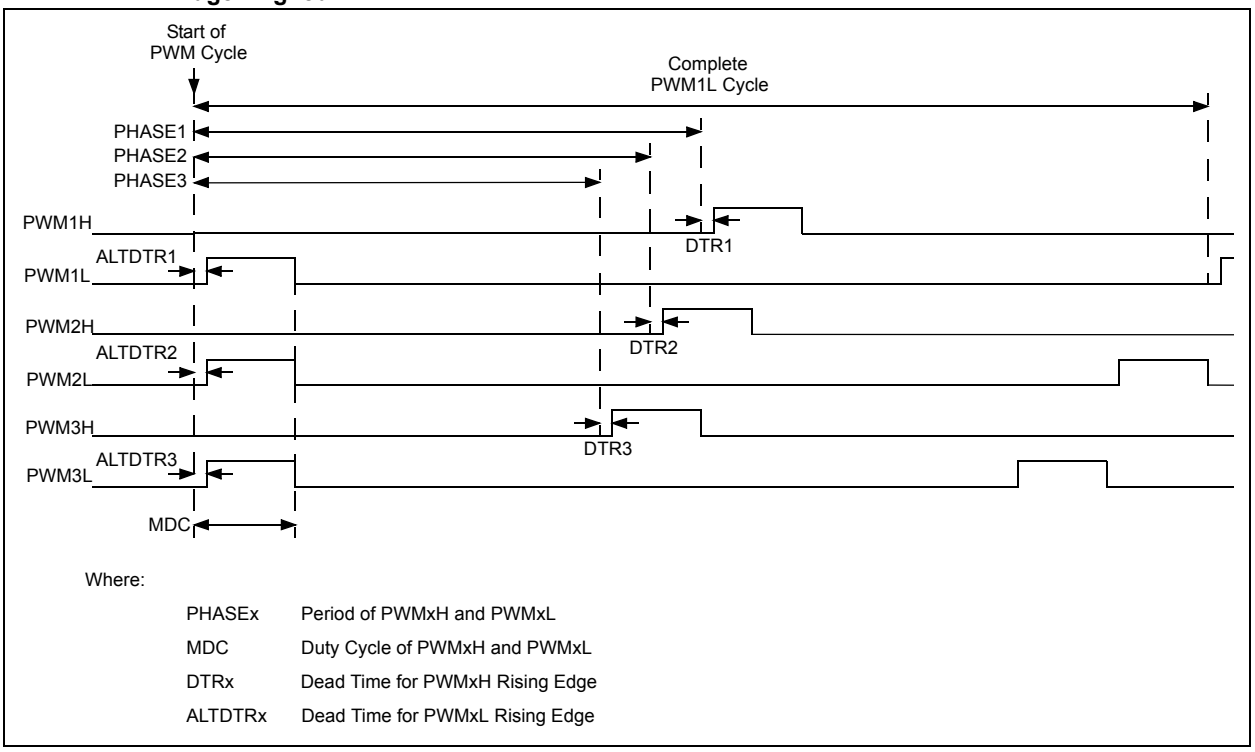
Where:

PHASEx	Period of PWMxH and PWMxL
PDCx	Duty Cycle of PWMxH and PWMxL
DTRx	Dead time for PWMxH Rising Edge
ALTDTRx	Dead time for PWMxL Rising Edge

Example 14-10: Push-Pull PWM Mode - Independent Duty Cycles and Independent Periods, No Phase-Shifting, Edge-Aligned

```
/* Set PWM Periods on PHASEx Registers */
PHASE1 = 1000;
PHASE2 = 900;
PHASE3 = 800;

/* Set Duty Cycles */
PDC1 = 200;
PDC2 = 300;
PDC3 = 400;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 25;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 25;

/* Set PWM Mode to Push-Pull */
IOCON1 = IOCON2 = IOCON3 = 0xC800;

/* Set Independent Time Bases, Edge-Aligned Mode and Independent Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0200;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

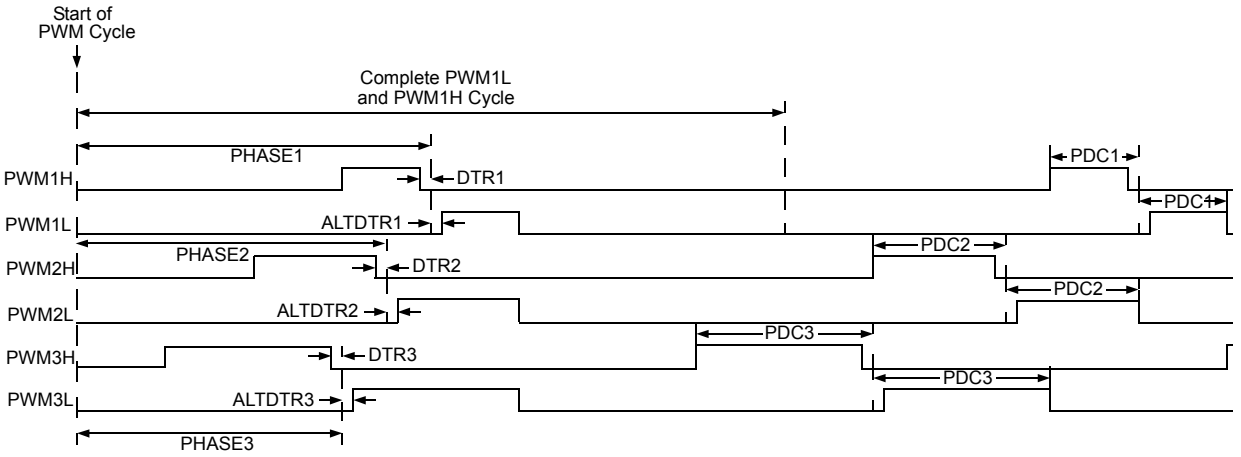
Figure 14-12: Push-Pull PWM Mode - Master Duty Cycles and Independent Periods, No Phase-Shifting, Edge-Aligned



Example 14-11: Push-Pull PWM Mode - Master Duty Cycles and Independent Periods, No Phase-Shifting, Edge-Aligned

```
/* Set PWM Periods on PHASEx Registers */
PHASE1 = 1000;
PHASE2 = 900;
PHASE3 = 800;

/* Set Duty Cycles */
MDC = 200;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 25;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 25;

/* Set PWM Mode to Push-Pull */
IOCON1 = IOCON2 = IOCON3 = 0xC800;

/* Set Independent Time Bases, Edge-Aligned Mode and Master Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0300;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

Figure 14-13: Push-Pull PWM Mode - Independent Duty Cycles and Independent Periods, No Phase-Shifting, Center-Aligned Mode



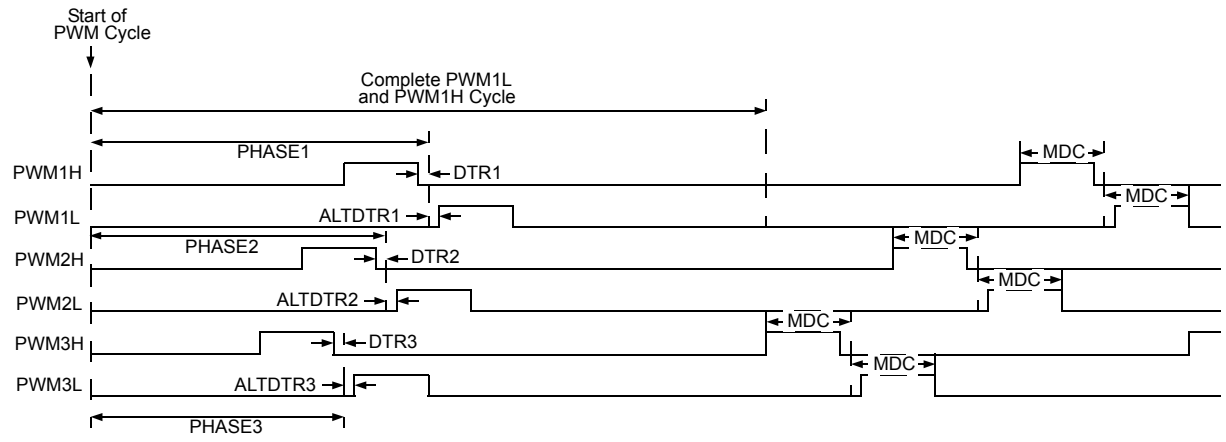
Where:

PHASEx	Period of PWMxH and PWMxL
PDCx	Duty Cycle of PWMxH and PWMxL
DTRx	Dead Time before PWMxH Falling Edge
ALTDTRx	Dead Time before PWMxL Falling Edge

Example 14-12: Push-Pull PWM Mode - Independent Duty Cycles and Independent Periods, No Phase-Shifting, Center-Aligned Mode

```
/* Set PWM Periods on PHASEx Registers */
PHASE1 = 1000;
PHASE2 = 900;
PHASE3 = 800;

/* Set Duty Cycles */
PDC1 = 200;
PDC2 = 300;
PDC3 = 400;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 25;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 25;

/* Set PWM Mode to Push-Pull */
IOCON1 = IOCON2 = IOCON3 = 0xC800;

/* Set Independent Time Bases, Center-Aligned Mode and Independent Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0204;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

Figure 14-14: Push-Pull PWM Mode - Master Duty Cycles and Independent Periods, No Phase-Shifting, Center-Aligned Mode



Where:

PHASEx	Period of PWMxH and PWMxL
MDC	Duty Cycle of PWMxH and PWMxL
DTRx	Dead Time before PWMxH Falling Edge
ALTDTRx	Dead Time before PWMxL Falling Edge

Example 14-13: Push-Pull PWM Mode - Master Duty Cycles and Independent Periods, No Phase-Shifting, Center-Aligned Mode

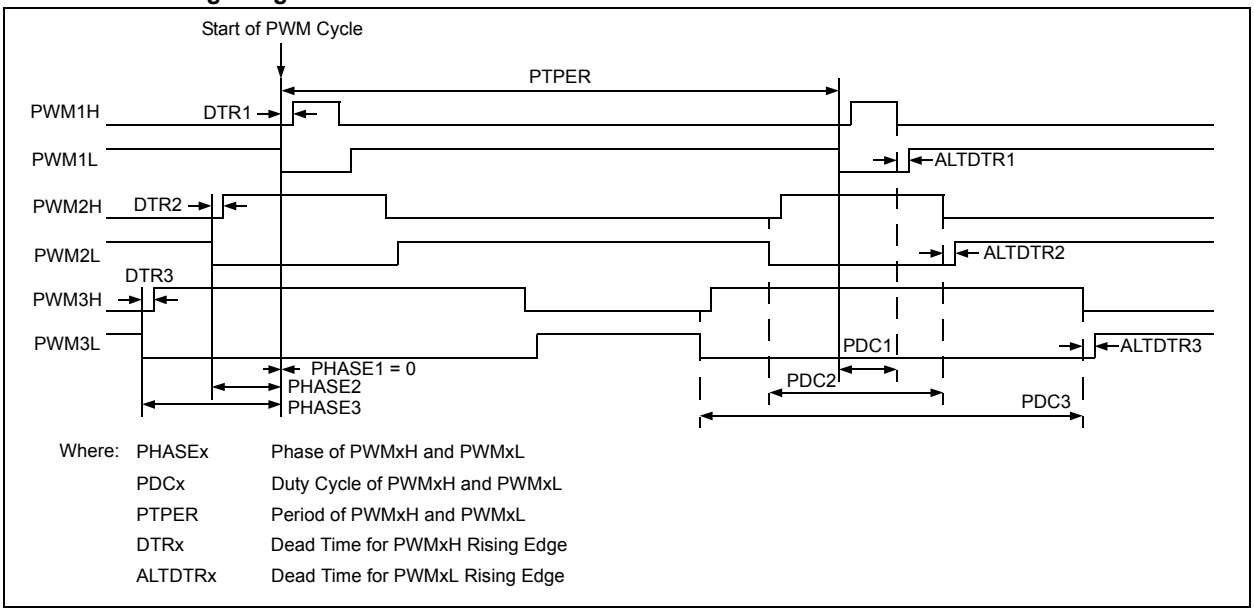
```
/* Set PWM Periods on PHASEx Registers */
PHASE1 = 1000;
PHASE2 = 900;
PHASE3 = 800;

/* Set Duty Cycles */
MDC = 200;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 25;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 25;

/* Set PWM Mode to Push-Pull */
IOCON1 = IOCON2 = IOCON3 = 0xC800;

/* Set Independent Time Bases, Center-Aligned Mode and Master Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0304;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

14.6.2 Complementary PWM Mode

In Complementary PWM mode, the PWM output PWMxH is the complement of the PWMxL output. Some typical applications of Complementary PWM mode are provided in **14.16 "Application Information"**.

Figure 14-15 through Figure 14-22 and Example 14-14 through Example 14-21 show the PWM outputs in multiple operating modes when the module operates in Complementary PWM mode.

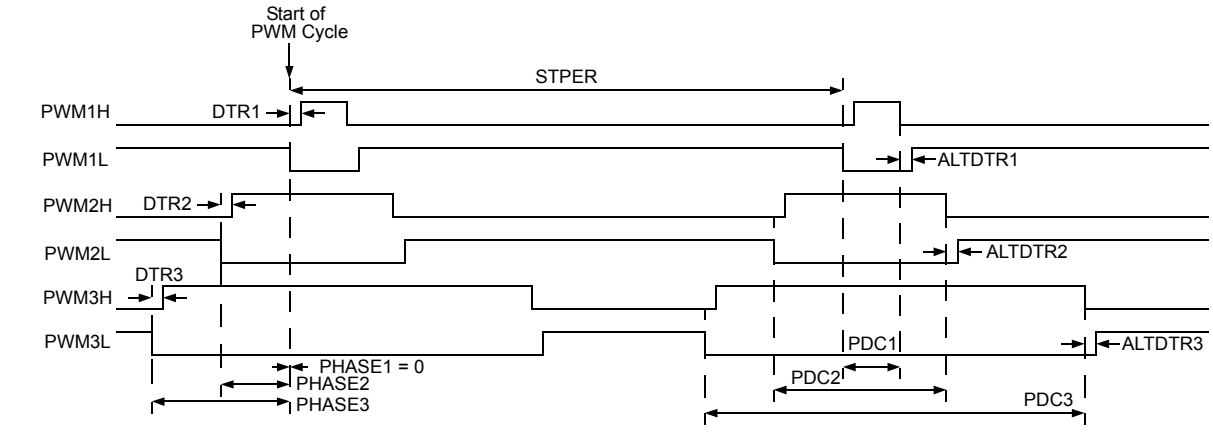
Figure 14-15: Complementary PWM Mode - Independent Duty Cycle and Phase, Fixed Primary Period, Edge-Aligned



Example 14-14: Complementary PWM Mode, Independent Duty Cycle and Phase, Fixed Primary Period, Edge-Aligned

```
/* Set PWM Period on Primary Time Base */
PTPER = 1000;

/* Set Phase Shift */
PHASE1 = 0;
PHASE2 = 100;
PHASE3 = 200;

/* Set Duty Cycles */
PDC1 = 150;
PDC2 = 200;
PDC3 = 400;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 25;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 25;

/* Set PWM Mode to Complementary */
IOCON1 = IOCON2 = IOCON3 = 0xC000;

/* Set Primary Time Base, Edge-Aligned Mode and Independent Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0000;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

Figure 14-16: Complementary PWM Mode - Independent Duty Cycle and Phase, Fixed Secondary Period, Edge-Aligned



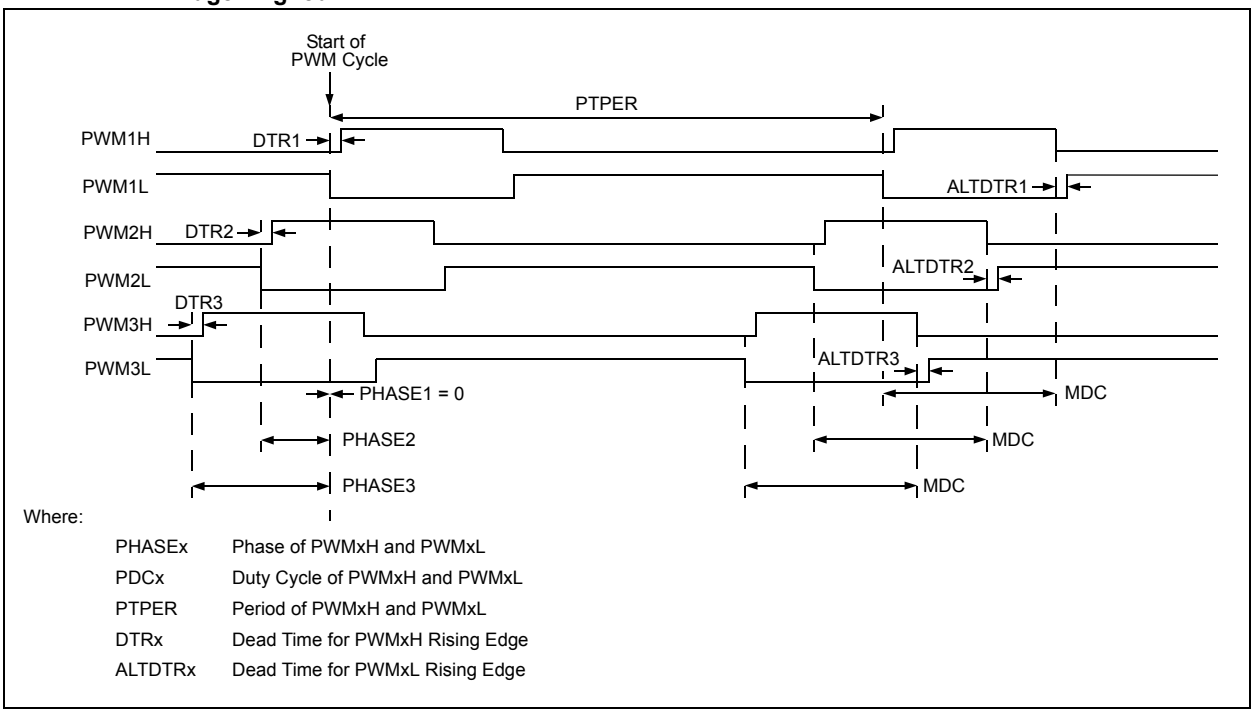
Where:

PHASEx	Phase of PWMxH and PWMxL
PDCx	Duty Cycle of PWMxH and PWMxL
STPER	Period of PWMxH and PWMxL
DTRx	Dead Time for PWMxH Rising Edge
ALTDTRx	Dead Time for PWMxL Rising Edge

Example 14-15: Complementary PWM Mode - Independent Duty Cycle and Phase, Fixed Secondary Period, Edge-Aligned

```
/* Set PWM Period on Secondary Time Base */
STPER = 1000;

/* Set Phase Shift */
PHASE1 = 0;
PHASE2 = 100;
PHASE3 = 200;

/* Set Duty Cycles */
PDC1 = 150;
PDC2 = 200;
PDC3 = 400;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 25;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 25;

/* Set PWM Mode to Complementary */
IOCON1 = IOCON2 = IOCON3 = 0xC000;

/* Set Secondary Time Base, Edge-Aligned Mode and Independent Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0008;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

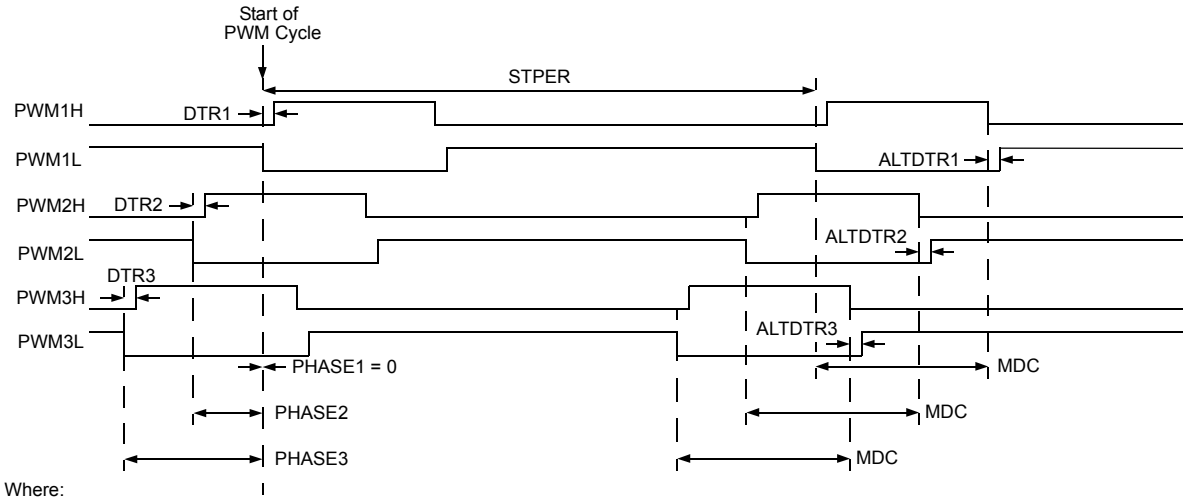
Figure 14-17: Complementary PWM Mode - Master Duty Cycle and Independent Phase, Fixed Primary Period, Edge-Aligned



Example 14-16: Complementary PWM Mode - Master Duty Cycle and Independent Phase, Fixed Primary Period, Edge-Aligned

```
/* Set PWM Period on Primary Time Base*/
PTPER = 1000;

/* Set Phase Shift */
PHASE1 = 0;
PHASE2 = 100;
PHASE3 = 200;

/* Set Duty Cycles */
MDC = 200;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 25;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 25;

/* Set PWM Mode to Complementary */
IOCON1 = IOCON2 = IOCON3 = 0xC000;

/* Set Primary Time Base, Edge-Aligned Mode and Master Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0100;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

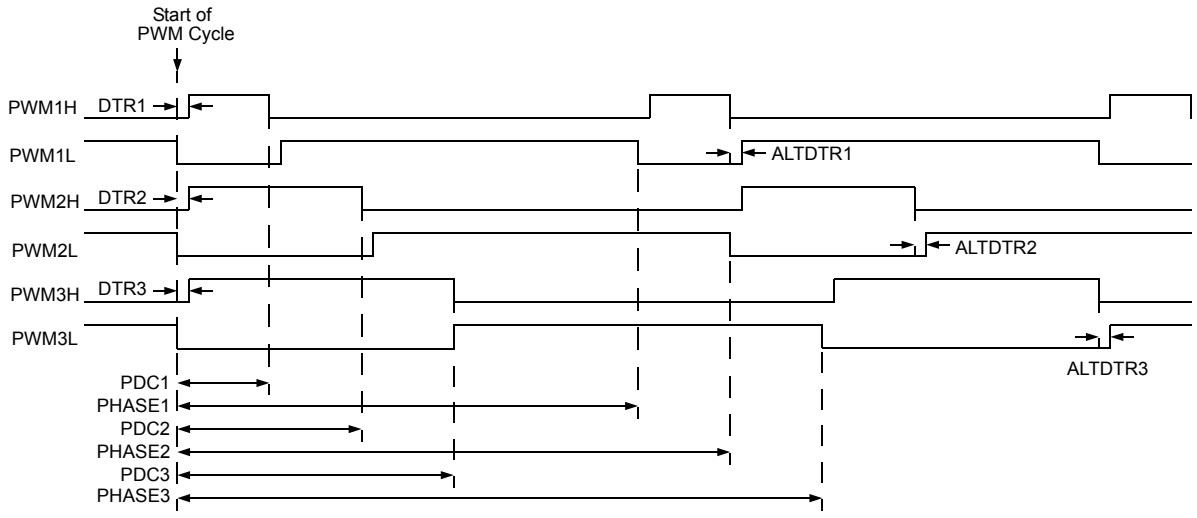
Figure 14-18: Complementary PWM Mode - Master Duty Cycle and Independent Phase, Fixed Secondary Period, Edge-Aligned



Example 14-17: Complementary PWM Mode - Master Duty Cycle and Independent Phase, Fixed Secondary Period, Edge-Aligned

```
/* Set PWM Period on Secondary Time Base*/
STPER = 1000;

/* Set Phase Shift */
PHASE1 = 0;
PHASE2 = 100;
PHASE3 = 200;

/* Set Duty Cycles */
MDC = 200;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 25;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 25;

/* Set PWM Mode to Complementary */
IOCON1 = IOCON2 = IOCON3 = 0xC000;

/* Set Secondary Time Base, Edge-Aligned Mode and Master Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0108;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

Figure 14-19: Complementary PWM Mode - Independent Duty Cycles and Independent Periods, No Phase-Shifting, Edge-Aligned



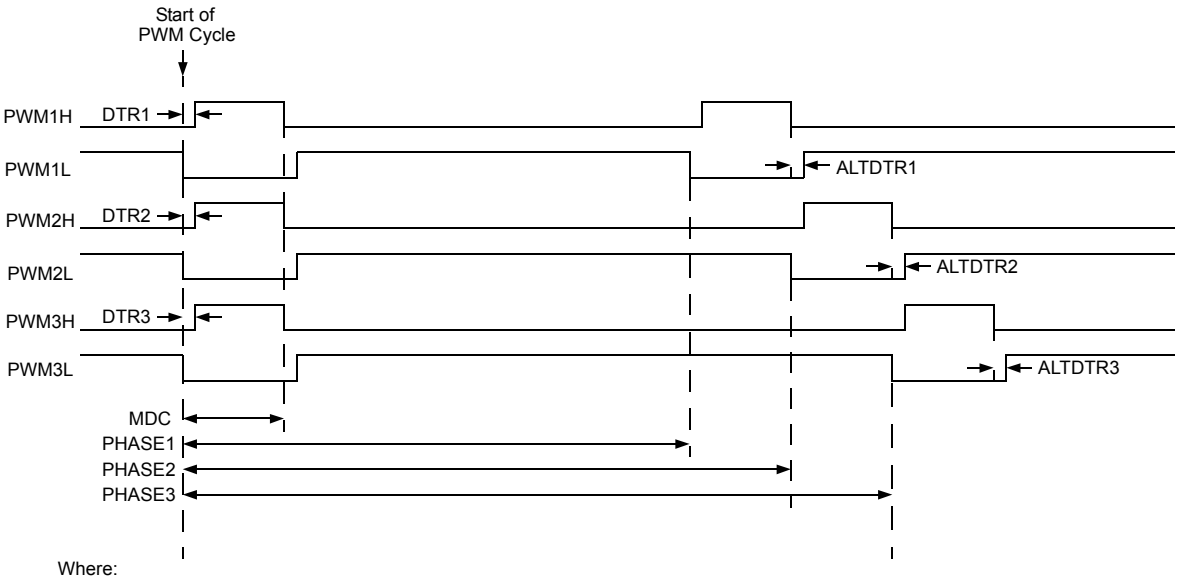
Where:

PHASEx	Period of PWMxH and PWMxL
PDCx	Duty Cycle of PWMxH and PWMxL
DTRx	Dead Time for PWMxH Rising Edge
ALTDTRx	Dead Time for PWMxL Rising Edge

Example 14-18: Complementary PWM Mode - Independent Duty Cycles and Independent Periods, No Phase-Shifting, Edge-Aligned

```
/* Set PWM Periods on PHASEx Registers */
PHASE1 = 800;
PHASE2 = 900;
PHASE3 = 1000;

/* Set Duty Cycles */
PDC1 = 200;
PDC2 = 300;
PDC3 = 400;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 25;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 25;

/* Set PWM Mode to Complementary */
IOCON1 = IOCON2 = IOCON3 = 0xC000;

/* Set Independent Time Bases, Edge-Aligned Mode and Independent Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0200;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

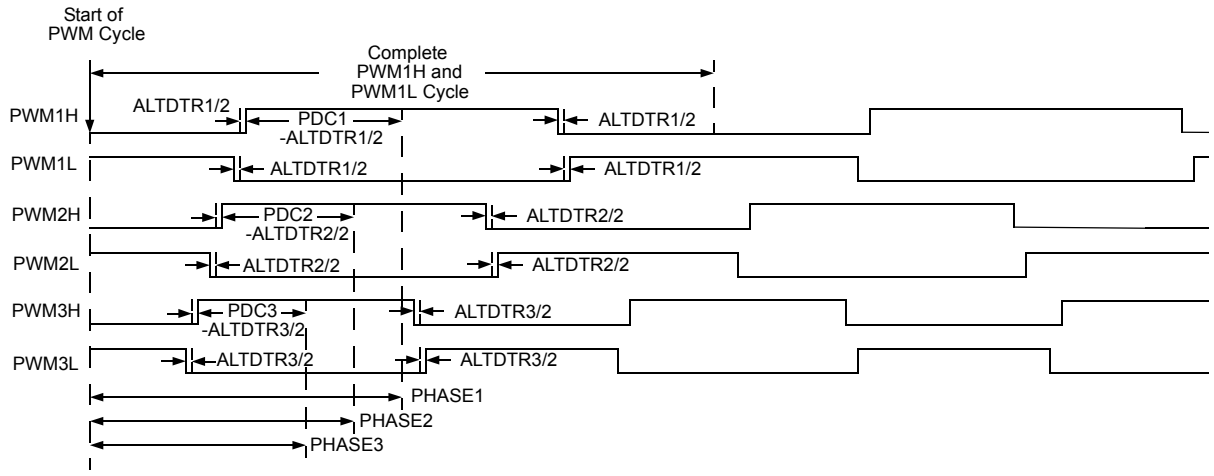
Figure 14-20: Complementary PWM Mode - Master Duty Cycles and Independent Periods, No Phase-Shifting, Edge-Aligned



Example 14-19: Complementary PWM Mode - Master Duty Cycles and Independent Periods, No Phase-Shifting, Edge-Aligned

```
/* Set PWM Periods on PHASEx Registers */
PHASE1 = 800;
PHASE2 = 900;
PHASE3 = 1000;

/* Set Duty Cycles */
MDC = 200;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 25;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 25;

/* Set PWM Mode to Complementary */
IOCON1 = IOCON2 = IOCON3 = 0xC000;

/* Set Independent Time Bases, Edge-Aligned Mode and Master Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0300;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

Figure 14-21: Complementary PWM Mode - Independent Duty Cycles and Independent Periods, No Phase-Shifting, Center-Aligned



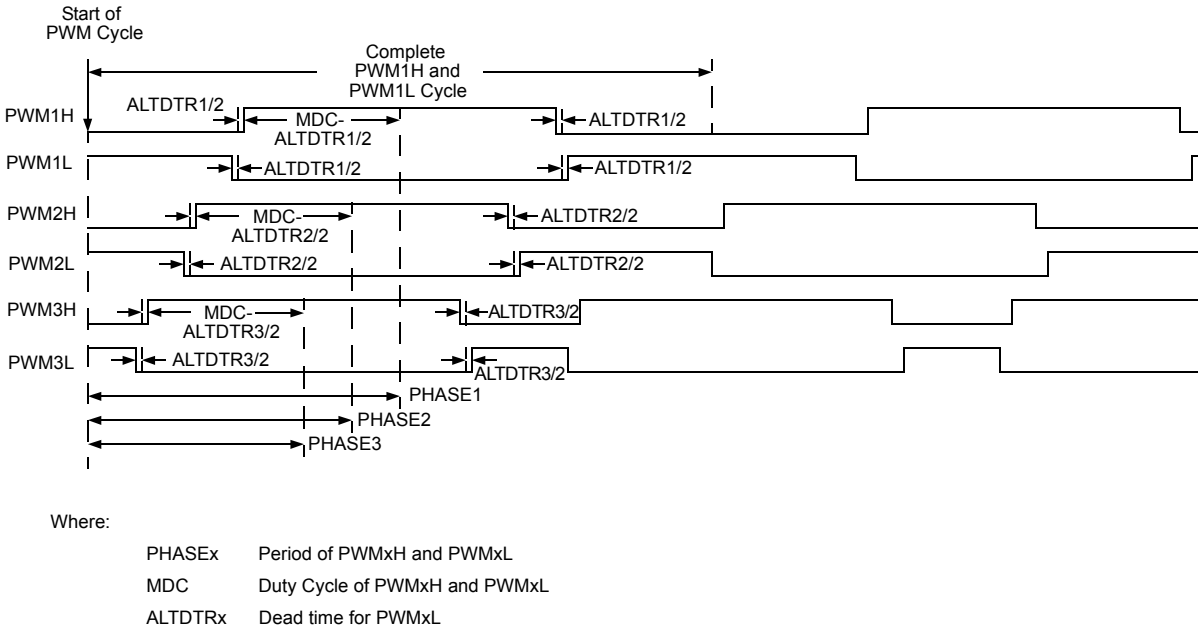
Where:

PHASEx	Period of PWMxH and PWMxL
PDCx	Duty Cycle of PWMxH and PWMxL
ALTDTRx	Dead time for PWMxL

Example 14-20: Complementary PWM Mode - Independent Duty Cycles and Independent Periods, No Phase-Shifting, Center-Aligned

```
/* Set PWM Periods on PHASEx Registers */
PHASE1 = 1000;
PHASE2 = 900;
PHASE3 = 800;

/* Set Duty Cycles */
PDC1 = 400;
PDC2 = 300;
PDC3 = 200;

/* Set Dead Time Values */
/* DTRx Registers are ignored in this mode */
DTR1 = DTR2 = DTR3 = 0;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 25;

/* Set PWM Mode to Complementary */
IOCON1 = IOCON2 = IOCON3 = 0xC000;

/* Set Independent Time Bases, Center-Aligned mode and
   Independent Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0204;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

Figure 14-22: Complementary PWM Mode - Master Duty Cycles and Independent Periods, No Phase-Shifting, Center-Aligned



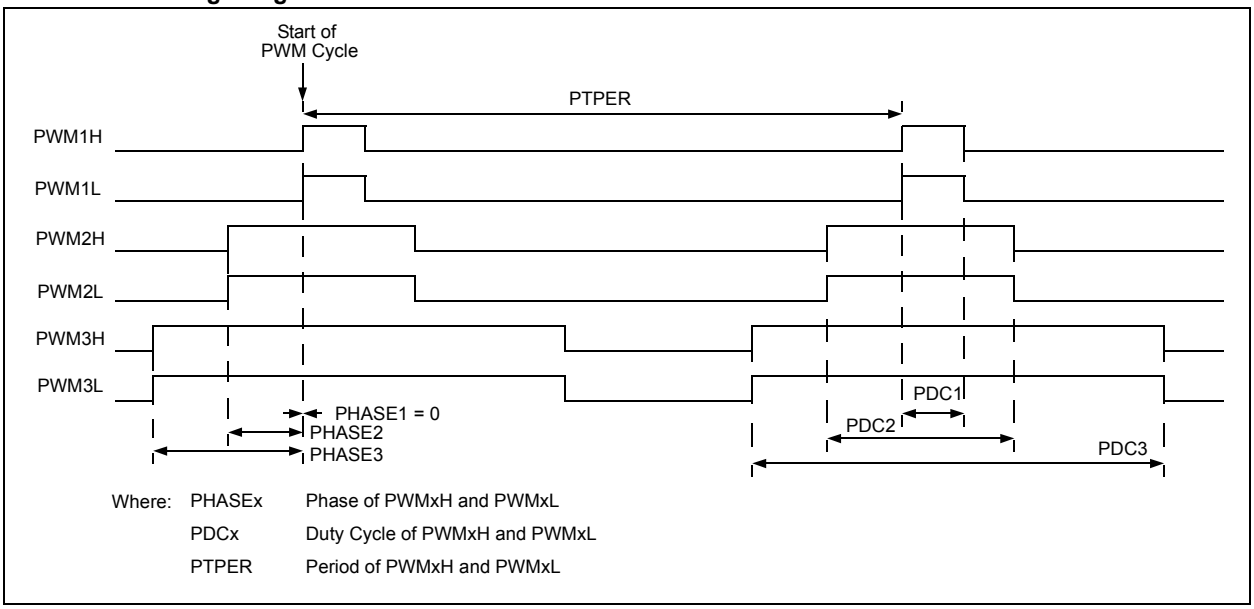
Example 14-21: Complementary PWM Mode - Master Duty Cycles and Independent Periods, No Phase-Shifting, Center-Aligned

```
/* Set PWM Periods on PHASEx Registers */
PHASE1 = 1000;
PHASE2 = 900;
PHASE3 = 800;

/* Set Duty Cycles */
MDC = 400;

/* Set Dead Time Values */
/* DTRx Registers are ignored in this mode */
DTR1 = DTR2 = DTR3 = 0;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 25;

/* Set PWM Mode to Complementary */
IOCON1 = IOCON2 = IOCON3 = 0xC000;

/* Set Independent Time Bases, Center-Aligned mode and
   Master Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0304;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

14.6.3 Redundant PWM Output Mode

In Redundant PWM Output mode, the High-Speed PWM module has the ability to provide two copies of a single-ended PWM output signal per PWM pin pair (PWMxH, PWMxL). This mode uses the PDCx register to specify the duty cycle. In this output mode, the two PWM output pins will provide the same PWM signal unless the user-assigned application specifies an override value. The following eight figures and examples show the Redundant PWM Output mode in multiple operating modes.

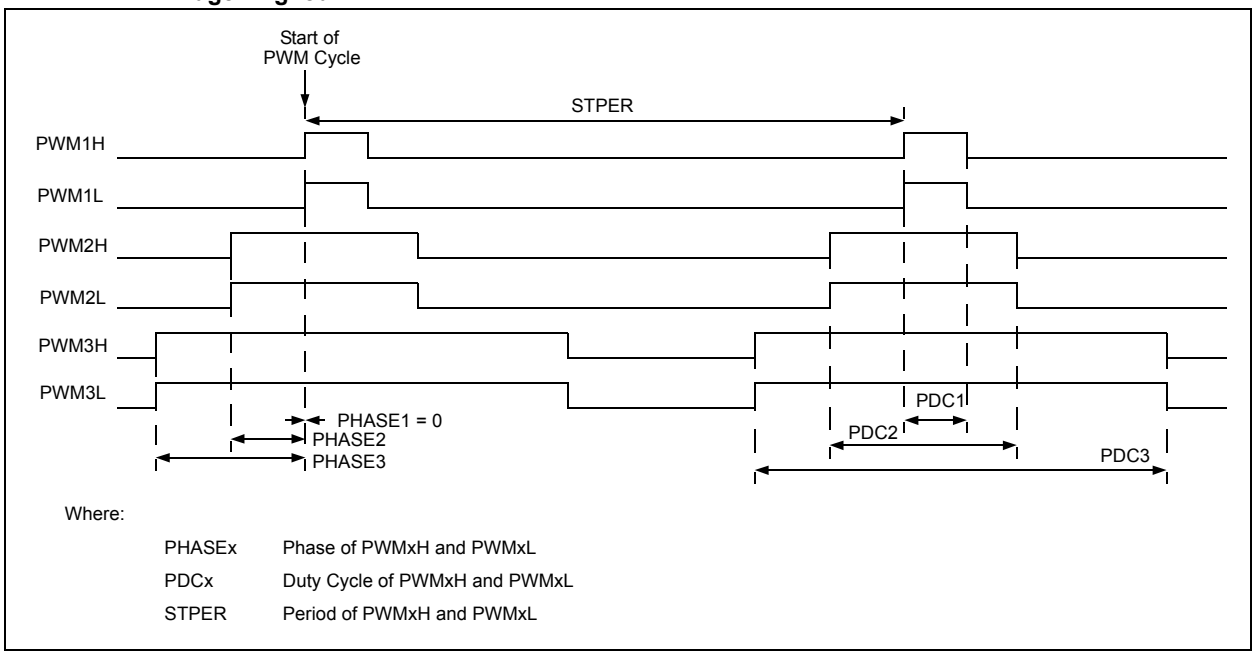
Figure 14-23: Redundant PWM Mode - Independent Duty Cycle and Phase, Fixed Primary Period, Edge-Aligned



Example 14-22: Redundant PWM Mode - Independent Duty Cycle and Phase, Fixed Primary Period, Edge-Aligned

```
/* Set PWM Period on Primary Time Base */
PTPER = 1000;

/* Set Phase Shift */
PHASE1 = 0;
PHASE2 = 100;
PHASE3 = 200;

/* Set Duty Cycles */
PDC1 = 150;
PDC2 = 200;
PDC3 = 400;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 0;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 0;

/* Set PWM Mode to Redundant */
IOCON1 = IOCON2 = IOCON3 = 0xC400;

/* Set Primary Time Base, Edge-Aligned Mode and Independent Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0000;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

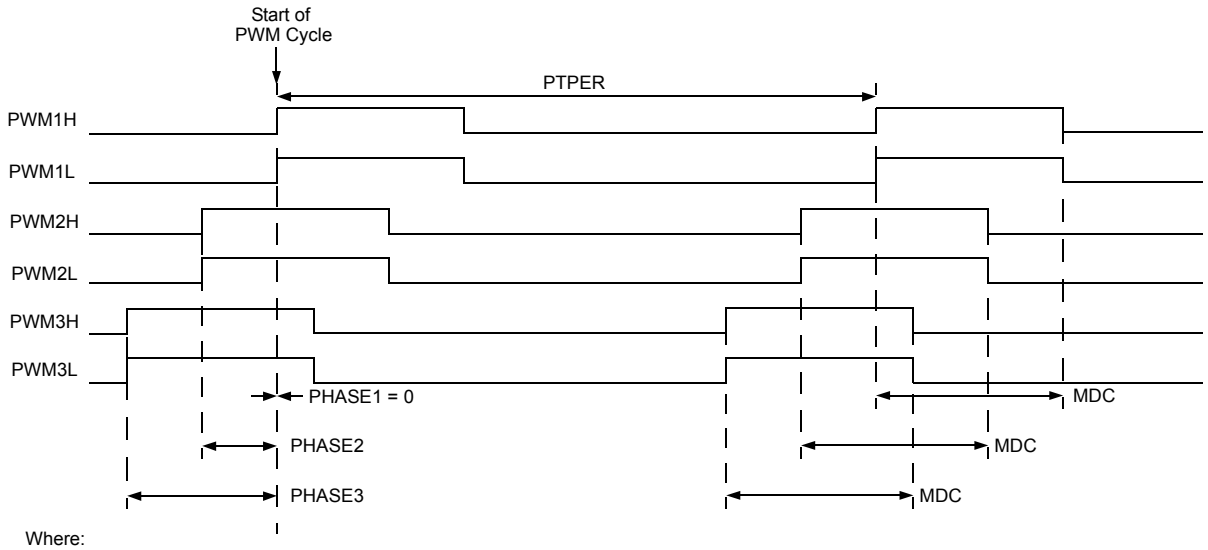
Figure 14-24: Redundant PWM Mode - Independent Duty Cycle and Phase, Fixed Secondary Period, Edge-Aligned



Example 14-23: Redundant PWM Mode - Independent Duty Cycle and Phase, Fixed Secondary Period, Edge-Aligned

```
/* Set PWM Period on Secondary Time Base */
STPER = 1000;

/* Set Phase Shift */
PHASE1 = 0;
PHASE2 = 100;
PHASE3 = 200;

/* Set Duty Cycles */
PDC1 = 150;
PDC2 = 200;
PDC3 = 400;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 0;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 0;

/* Set PWM Mode to Redundant */
IOCON1 = IOCON2 = IOCON3 = 0xC400;

/* Set Secondary Time Base, Edge-Aligned Mode and Independent Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0008;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

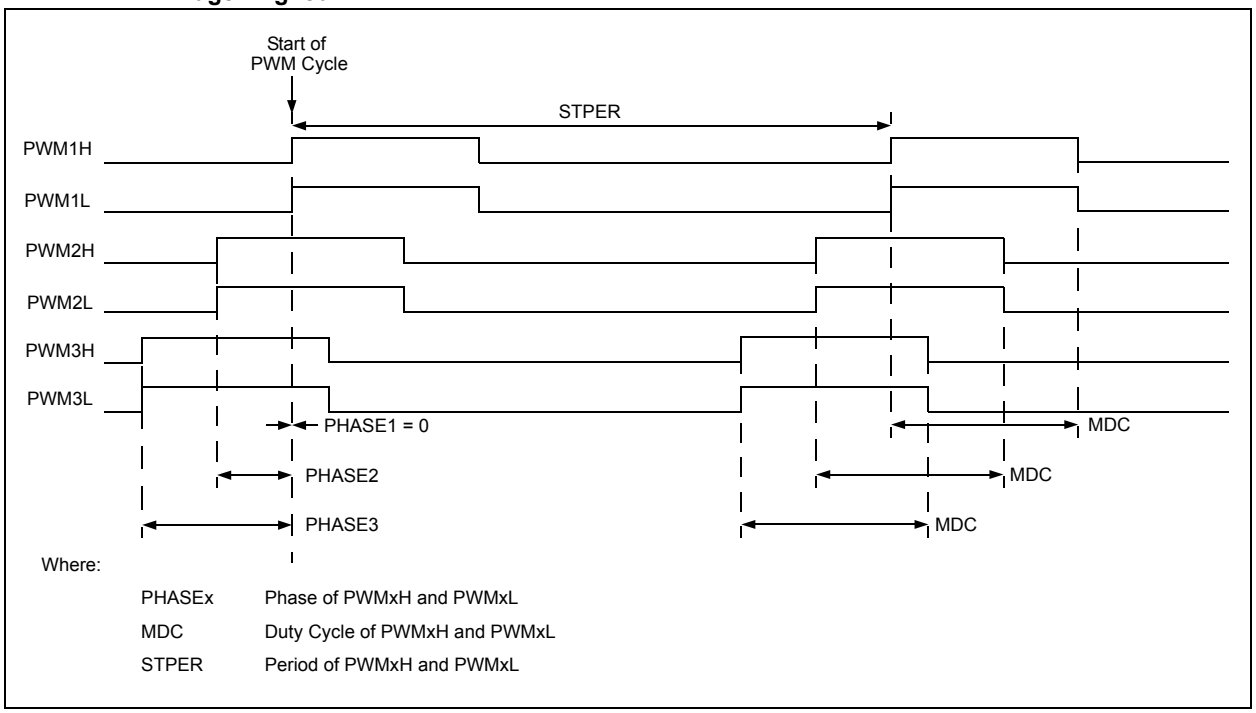
Figure 14-25: Redundant PWM Mode - Master Duty Cycle and Variable Phase, Fixed Primary Period, Edge-Aligned



Example 14-24: Redundant PWM Mode - Master Duty Cycle and Variable Phase, Fixed Primary Period, Edge-Aligned

```
/* Set PWM Period on Primary Time Base */
PTPER = 1000;

/* Set Phase Shift */
PHASE1 = 0;
PHASE2 = 100;
PHASE3 = 200;

/* Set Duty Cycles */
MDC = 400;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 0;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 0;

/* Set PWM Mode to Redundant */
IOCON1 = IOCON2 = IOCON3 = 0xC400;

/* Set Primary Time Base, Edge-Aligned Mode and Master Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0100;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

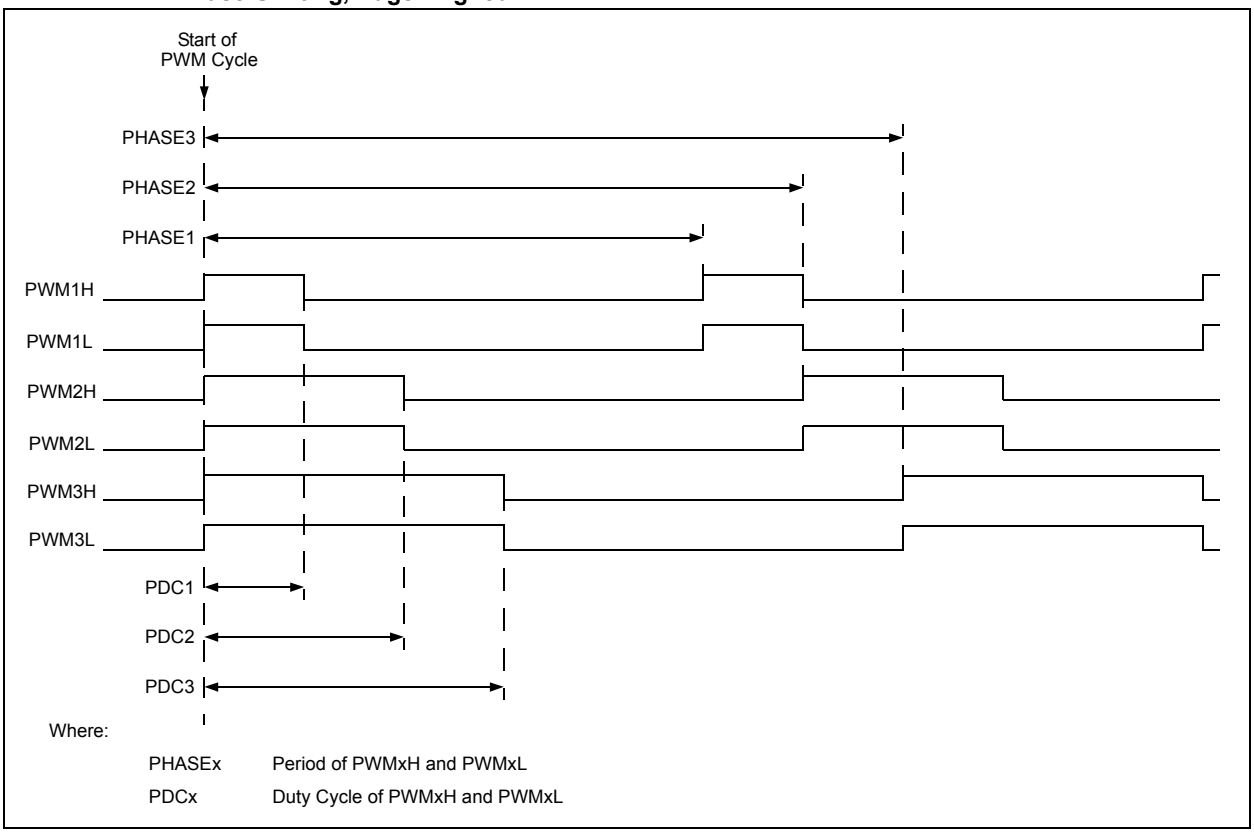
Figure 14-26: Redundant PWM Mode - Master Duty Cycle and Variable Phase, Fixed Secondary Period, Edge-Aligned



Example 14-25: Redundant PWM Mode - Master Duty Cycle and Variable Phase, Fixed Secondary Period, Edge-Aligned

```
/* Set PWM Period on Secondary Time Base */
STPER = 1000;

/* Set Phase Shift */
PHASE1 = 0;
PHASE2 = 100;
PHASE3 = 200;

/* Set Duty Cycles */
MDC = 400;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 0;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 0;

/* Set PWM Mode to Redundant */
IOCON1 = IOCON2 = IOCON3 = 0xC400;

/* Set Secondary Time Base, Edge-Aligned Mode and Master Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0108;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

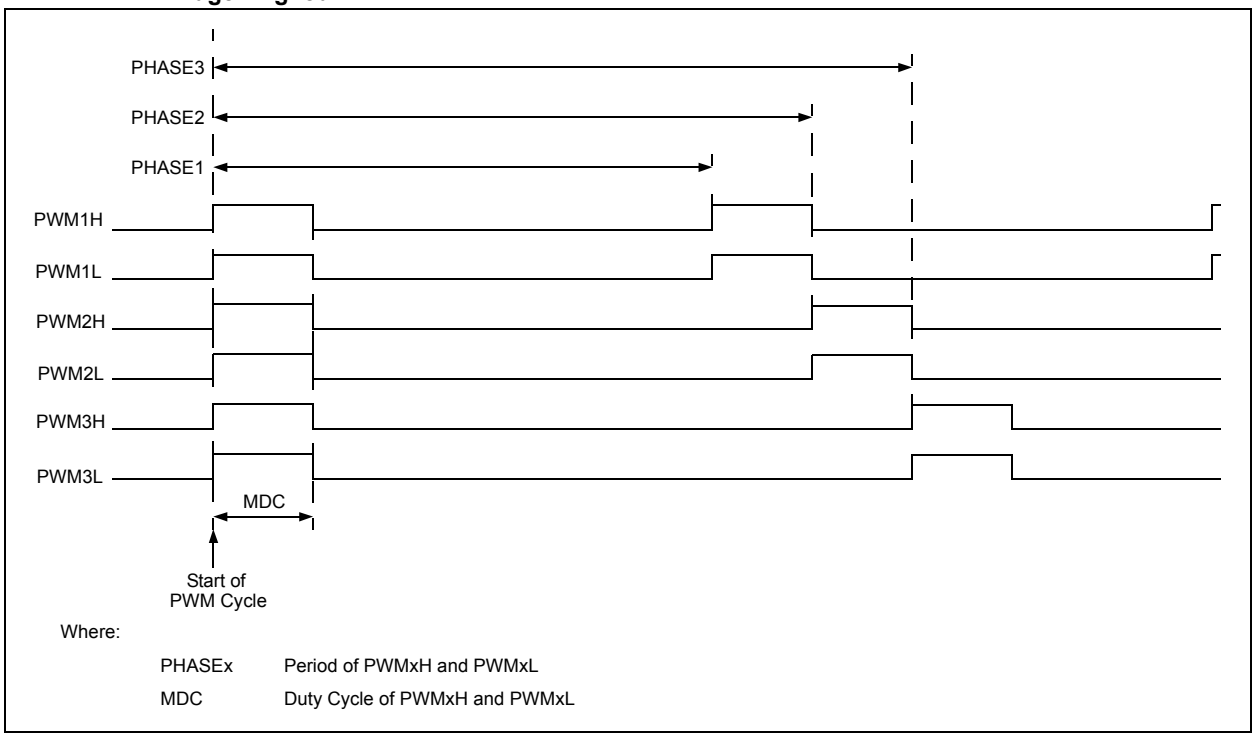
Figure 14-27: Redundant PWM Mode - Independent Duty Cycles and Independent Periods, No Phase-Shifting, Edge-Aligned



Example 14-26: Redundant PWM Mode - Independent Duty Cycles and Independent Periods, No Phase-Shifting, Edge-Aligned

```
/* Set PWM Periods on PHASEx Registers */
PHASE1 = 800;
PHASE2 = 900;
PHASE3 = 1000;

/* Set Duty Cycles */
PDC1 = 200;
PDC2 = 300;
PDC3 = 400;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 0;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 0;

/* Set PWM Mode to Redundant */
IOCON1 = IOCON2 = IOCON3 = 0xC400;

/* Set Independent Time Bases, Edge-Aligned Mode and Independent Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0200;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

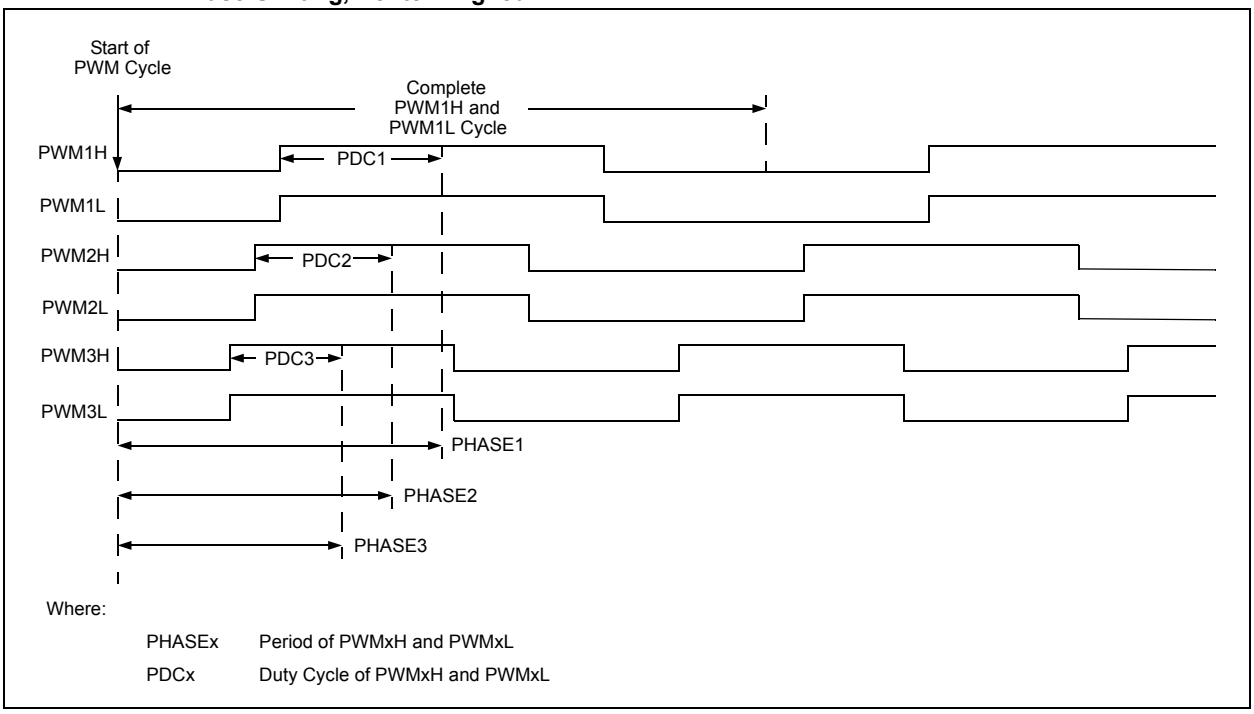
Figure 14-28: Redundant PWM Mode - Master Duty Cycles and Independent Periods, No Phase-Shifting, Edge-Aligned



Example 14-27: Redundant PWM Mode - Master Duty Cycles and Independent Periods, No Phase-Shifting, Edge-Aligned

```
/* Set PWM Periods on PHASEx Registers */
PHASE1 = 800;
PHASE2 = 900;
PHASE3 = 1000;

/* Set Duty Cycles */
MDC = 200;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 0;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 0;

/* Set PWM Mode to Redundant */
IOCON1 = IOCON2 = IOCON3 = 0xC400;

/* Set Independent Time Bases, Edge-Aligned Mode and Master Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0300;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

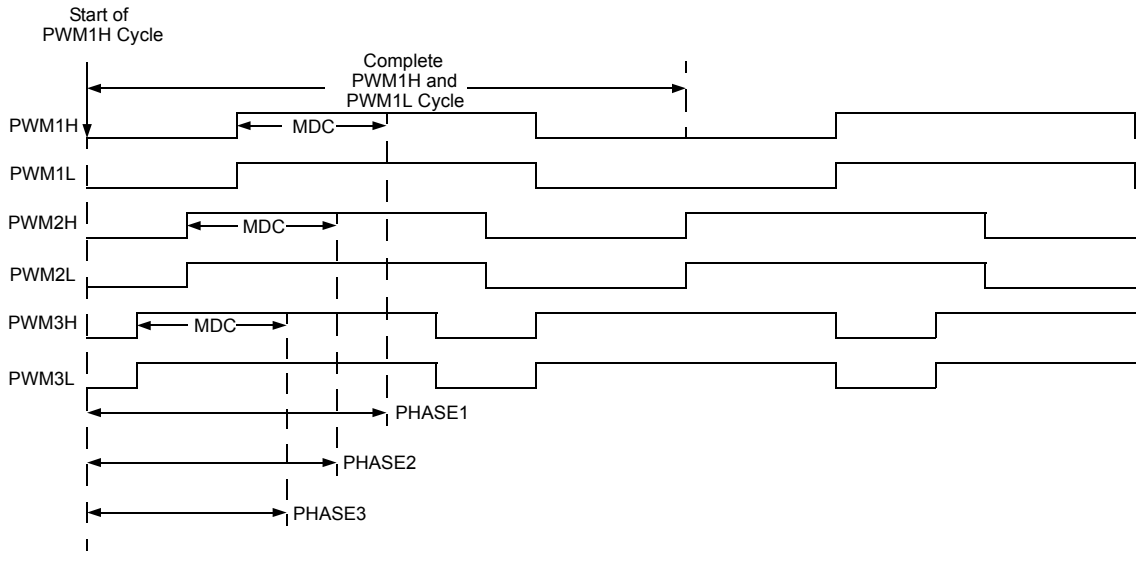
Figure 14-29: Redundant PWM Mode - Independent Duty Cycles and Independent Periods, No Phase-Shifting, Center-Aligned



Example 14-28: Redundant PWM Mode - Independent Duty Cycles and Independent Periods, No Phase-Shifting, Center-Aligned

```
/* Set PWM Periods on PHASEx Registers */
PHASE1 = 1000;
PHASE2 = 900;
PHASE3 = 800;

/* Set Duty Cycles */
PDC1 = 400;
PDC2 = 300;
PDC3 = 200;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 0;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 0;

/* Set PWM Mode to Redundant */
IOCON1 = IOCON2 = IOCON3 = 0xC400;

/* Set Independent Time Bases, Center-Aligned Mode and Independent Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0204;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

Figure 14-30: Redundant PWM Mode - Master Duty Cycles and Independent Periods, No Phase-Shifting, Center-Aligned



Example 14-29: Redundant PWM Mode, Master Duty Cycles and Independent Periods, No Phase-shifting, Center-Aligned

```
/* Set PWM Periods on PHASEx Registers */
PHASE1 = 1000;
PHASE2 = 900;
PHASE3 = 800;

/* Set Duty Cycles */
MDC = 200;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 0;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 0;

/* Set PWM Mode to Redundant */
IOCON1 = IOCON2 = IOCON3 = 0xC400;

/* Set Independent Time Bases, Center-Aligned Mode and Master Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0304;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

Table 14-2 provides PWM register functionality for the PWM modes.

Table 14-2: Complementary, Push-Pull and Redundant Mode Register Functionality

Function	Configuration in PWMCONx			Pins	
	MDCS	ITB	MTBS	PWMxH	PWMxL
PWM Duty Cycle	0	x	x	PDCx	PDCx
	1	x	x	MDC	MDC
PWM Phase Shift	x	0	x	PHASEx	PHASEx
PWM Period	x	x	0	PTPER	PTPER
	x	x	1	STPER	STPER
	x	1	x	PHASEx	PHASEx

Legend: x = don't care

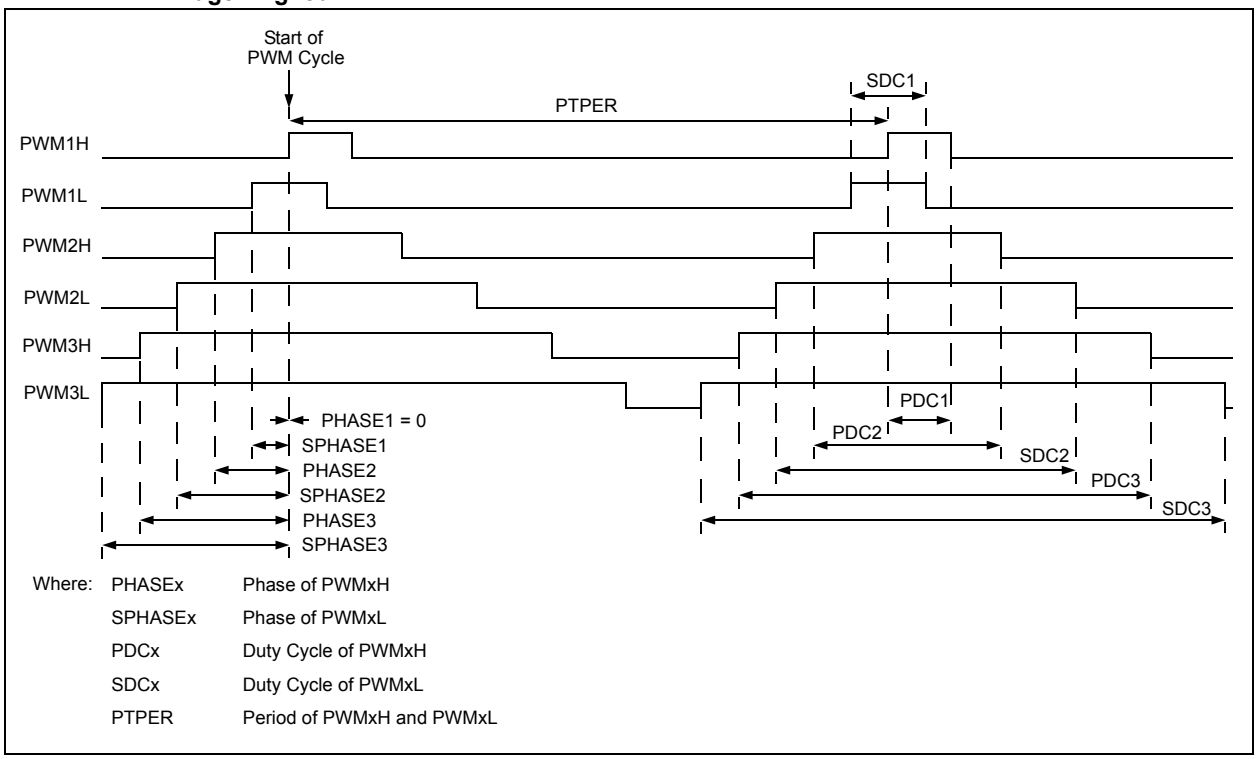
14.6.4 True Independent PWM Output Mode

In True Independent PWM Output mode (PMOD = 11), the PWM outputs (PWMxH and PWMxL) can have different duty cycles. The PDCx register specifies the duty cycle for the PWMxH output, whereas the SDCx register specifies the duty cycle for the PWMxL output. In addition, the PWMxH and PWMxL outputs can either have different periods or they can be phase-shifted relative to each other.

- When ITB = 1, the PHASEx register specifies the PWM period for the PWMxH output and the SPHASEx register specifies the PWM period for the PWMxL output
- When ITB = 0, the PHASEx register specifies the phase shift for the PWMxH output and the SPHASEx register specifies the phase shift for the PWMxL output

Note: In Independent Time Base mode (ITB = 1), there may not be a deterministic phase relationship between the PWMxH and PWMxL outputs.

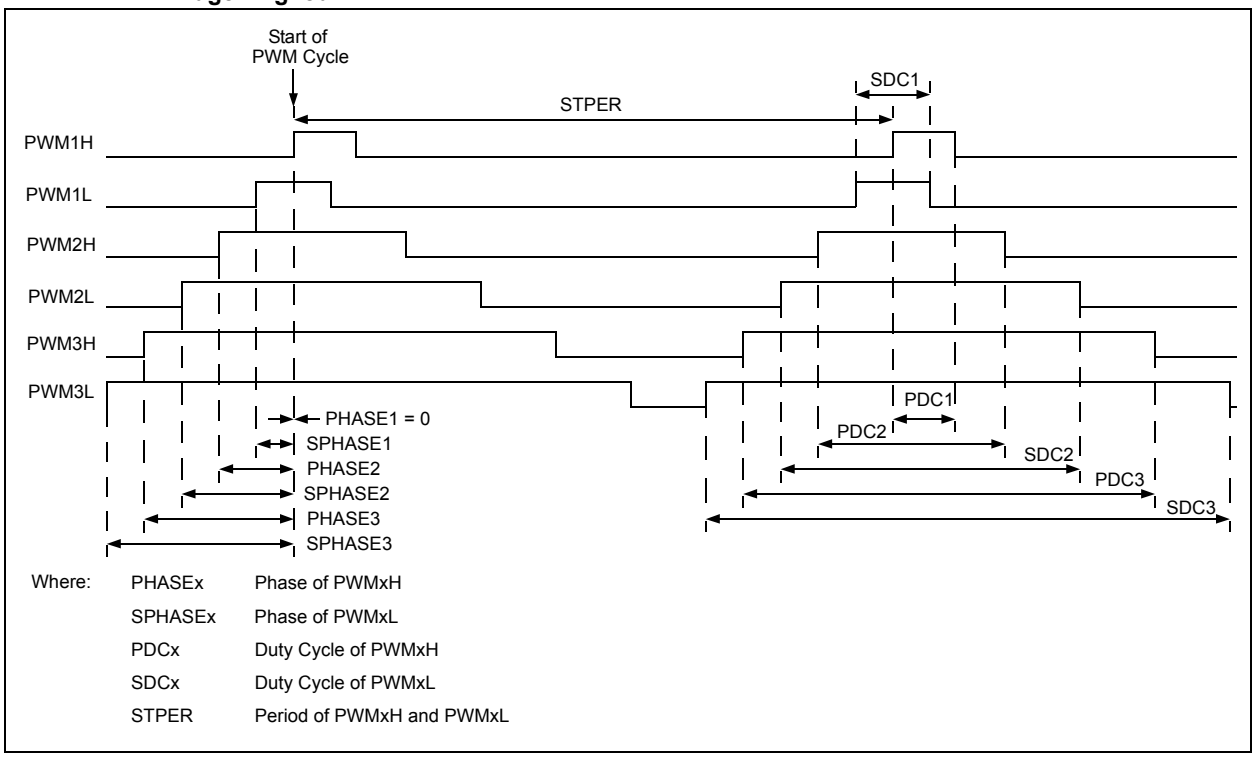
Figure 14-31: Independent PWM Mode - Independent Duty Cycle and Phase, Fixed Primary Period, Edge-Aligned



Example 14-30: Independent PWM Mode - Independent Duty Cycle and Phase, Fixed Primary Period, Edge-Aligned

```
/* Set PWM Period on Primary Time Base */
PTPER = 1000;

/* Set Phase Shift */
PHASE1 = 0;
SPHASE1 = 100;
PHASE2 = 200;
SPHASE2 = 300;
PHASE3 = 400;
SPHASE3 = 500;

/* Set Duty Cycles */
PDC1 = 100;
SDC1 = 200;
PDC2 = 300;
SDC2 = 400;
PDC3 = 500;
SDC3 = 600;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 0;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 0;

/* Set PWM Mode to Independent */
IOCON1 = IOCON2 = IOCON3 = 0xCC00;

/* Set Primary Time Base, Edge-Aligned Mode and Independent Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0000;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

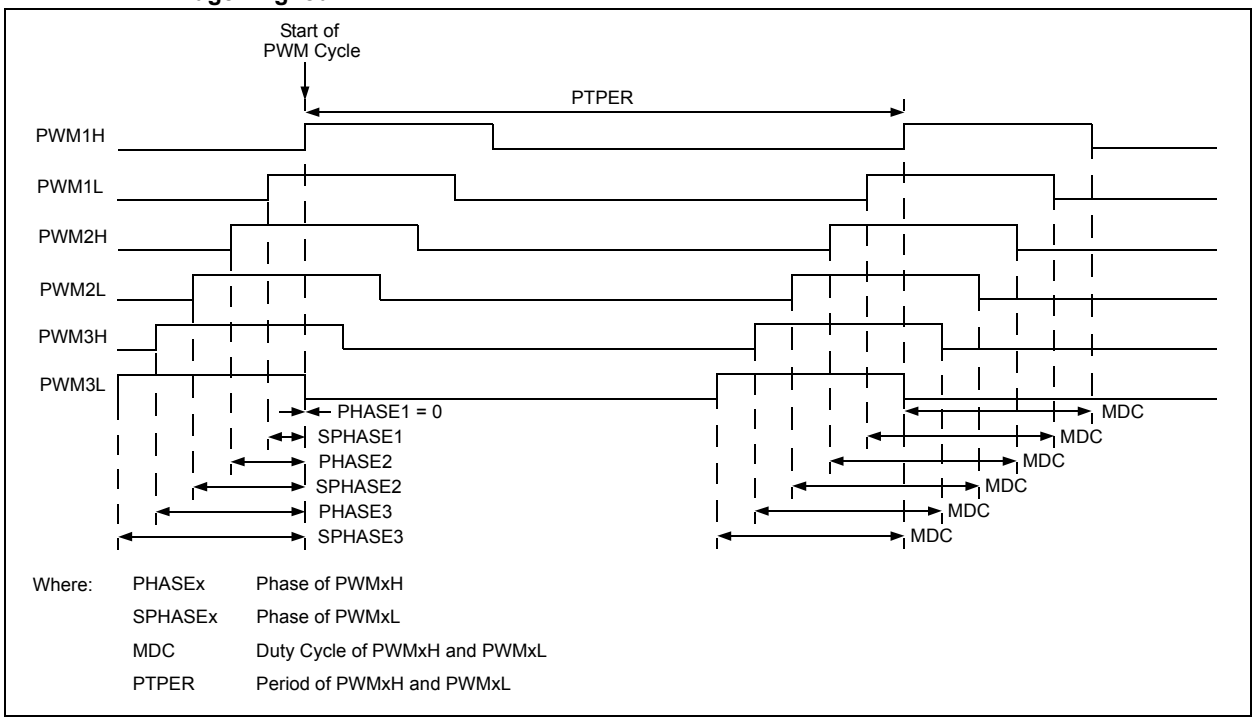
Figure 14-32: Independent PWM Mode - Independent Duty Cycle and Phase, Fixed Secondary Period, Edge-Aligned



Example 14-31: Independent PWM Mode - Independent Duty Cycle and Phase, Fixed Secondary Period, Edge-Aligned

```
/* Set PWM Period on Secondary Time Base */
STPER = 1000;

/* Set Phase Shift */
PHASE1 = 0;
SPHASE1 = 100;
PHASE2 = 200;
SPHASE2 = 300;
PHASE3 = 400;
SPHASE3 = 500;

/* Set Duty Cycles */
PDC1 = 100;
SDC1 = 200;
PDC2 = 300;
SDC2 = 400;
PDC3 = 500;
SDC3 = 600;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 0;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 0;

/* Set PWM Mode to Independent */
IOCON1 = IOCON2 = IOCON3 = 0xCC00;

/* Set Secondary Time Base, Edge-Aligned Mode and Independent Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0008;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

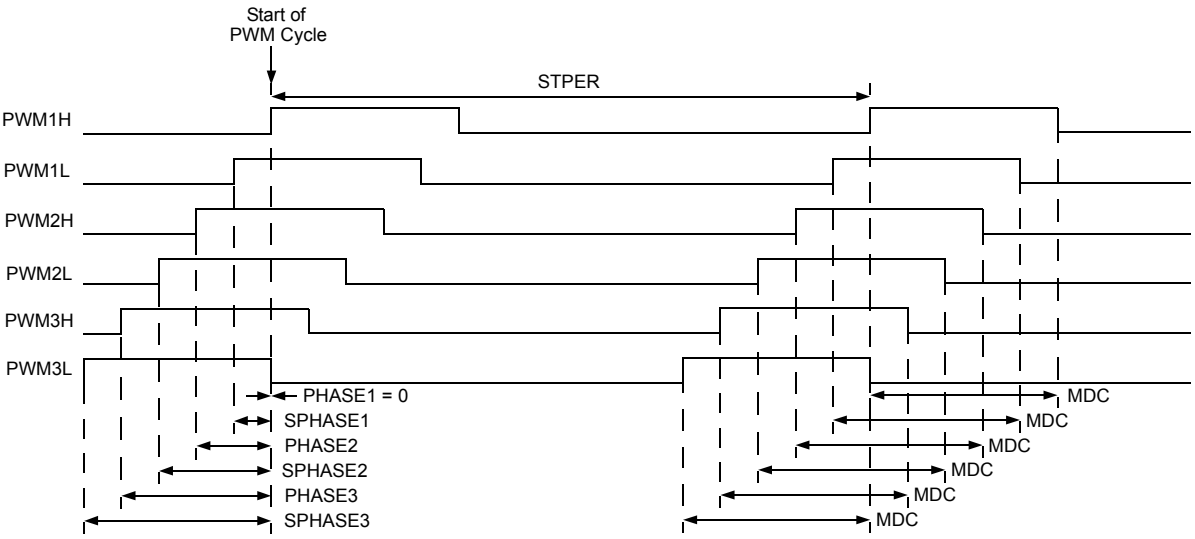
Figure 14-33: Independent PWM Mode - Master Duty Cycle, Variable Phase, Fixed Primary Period, Edge-Aligned



Example 14-32: Independent PWM Mode - Master Duty Cycle, Variable Phase, Fixed Primary Period, Edge-Aligned

```
/* Set PWM Period on Primary Time Base */
PTPER = 1000;

/* Set Phase Shift */
PHASE1 = 0;
SPHASE1 = 100;
PHASE2 = 200;
SPHASE2 = 300;
PHASE3 = 400;
SPHASE3 = 500;

/* Set Duty Cycles */
MDC = 300;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 0;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 0;

/* Set PWM Mode to Independent */
IOCON1 = IOCON2 = IOCON3 = 0xCC00;

/* Set Primary Time Base, Edge-Aligned Mode and Master Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0100;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

Figure 14-34: Independent PWM Mode - Master Duty Cycle, Variable Phase, Fixed Secondary Period, Edge-Aligned



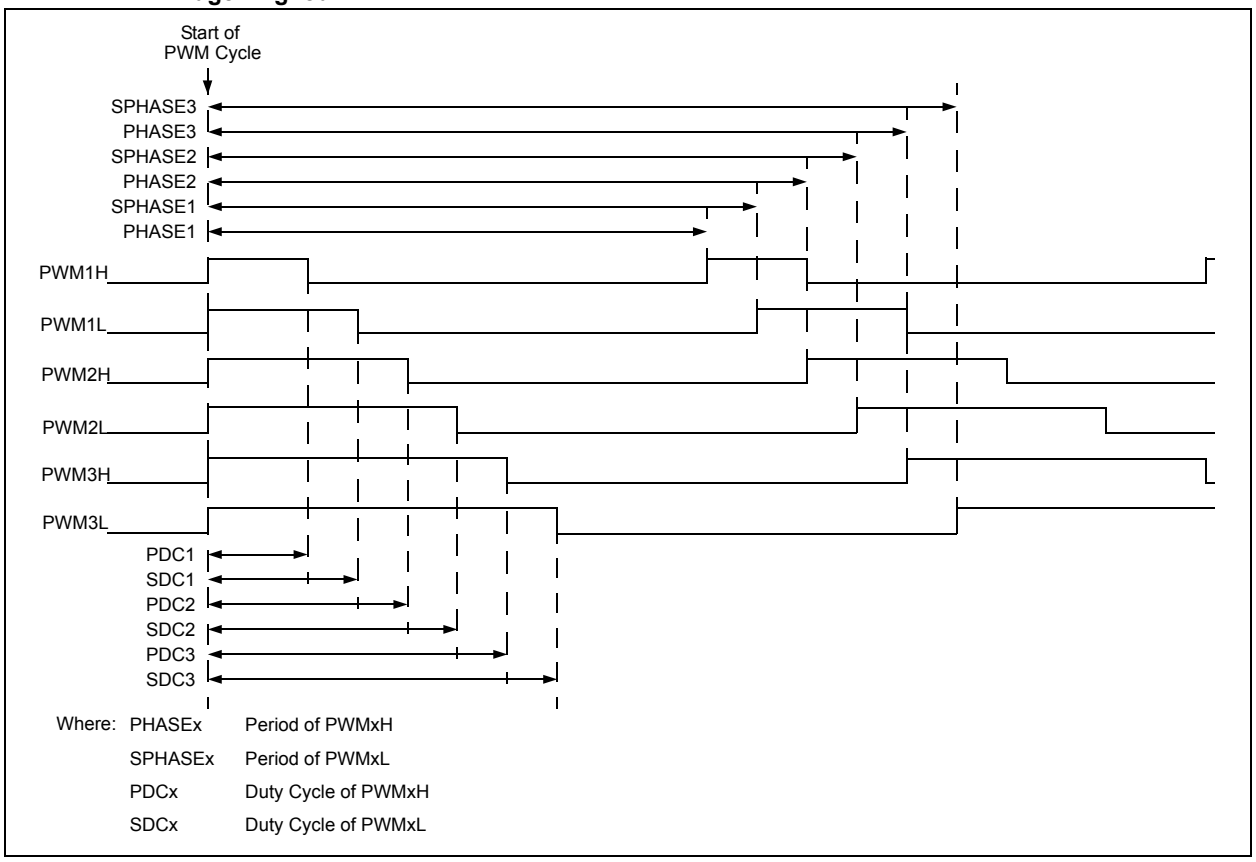
Where:

	PHASEx	Phase of PWMxH
	SPHASEx	Phase of PWMxL
	MDC	Duty Cycle of PWMxH and PWMxL
	STPER	Period of PWMxH and PWMxL

Example 14-33: Independent PWM Mode - Master Duty Cycle, Variable Phase, Fixed Secondary Period, Edge -Aligned

```
/* Set PWM Period on Secondary Time Base */
STPER = 1000;

/* Set Phase Shift */
PHASE1 = 0;
SPHASE1 = 100;
PHASE2 = 200;
SPHASE2 = 300;
PHASE3 = 400;
SPHASE3 = 500;

/* Set Duty Cycles */
MDC = 300;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 0;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 0;

/* Set PWM Mode to Independent */
IOCON1 = IOCON2 = IOCON3 = 0xCC00;

/* Set Secondary Time Base, Edge-Aligned Mode and Master Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0108;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

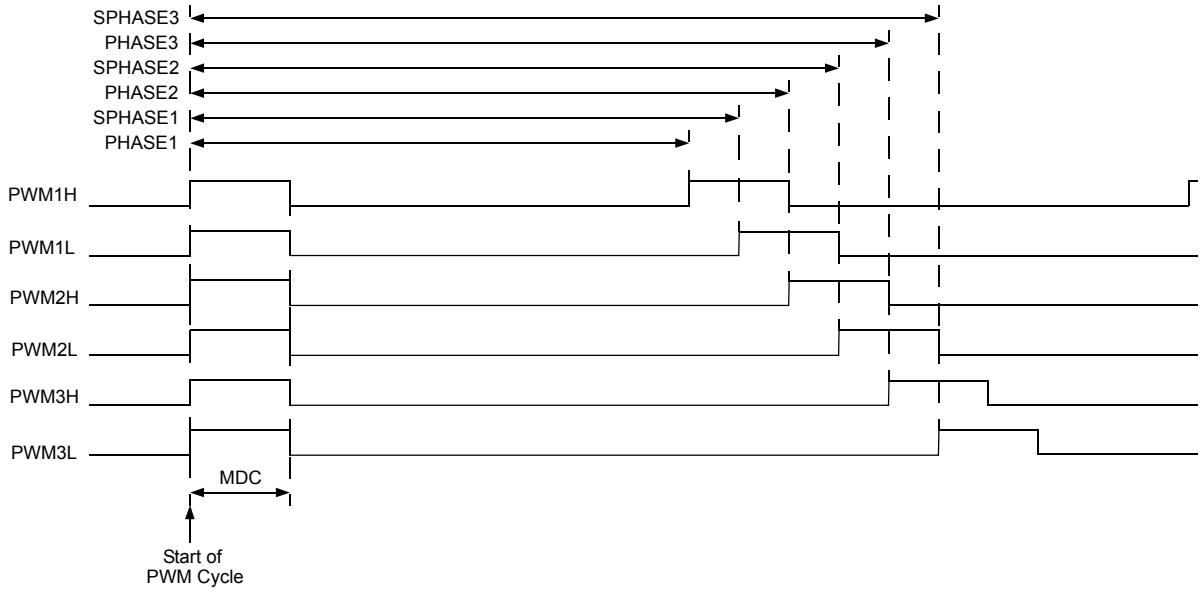
Figure 14-35: Independent PWM Mode - Independent Duty Cycles and Periods, No Phase-Shifting, Edge-Aligned



Example 14-34: Independent PWM Mode - Independent Duty Cycles and Periods, No Phase-Shifting, Edge-Aligned

```
/* Set PWM Periods on PHASEx Registers */
PHASE1 = 800;
SPHASE1 = 900;
PHASE2 = 1000;
SPHASE2 = 1100;
PHASE3 = 1200;
SPHASE3 = 1300;

/* Set Duty Cycles */
PDC1 = 200;
SDC1 = 300;
PDC2 = 400;
SDC2 = 500;
PDC3 = 600;
SDC3 = 700;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 0;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 0;

/* Set PWM Mode to Independent */
IOCON1 = IOCON2 = IOCON3 = 0xCC00;

/* Set Independent Time Bases, Edge-Aligned Mode and Independent Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0200;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

Figure 14-36: Independent PWM Mode - Master Duty Cycles, Independent Periods, No Phase-Shifting, Edge-Aligned

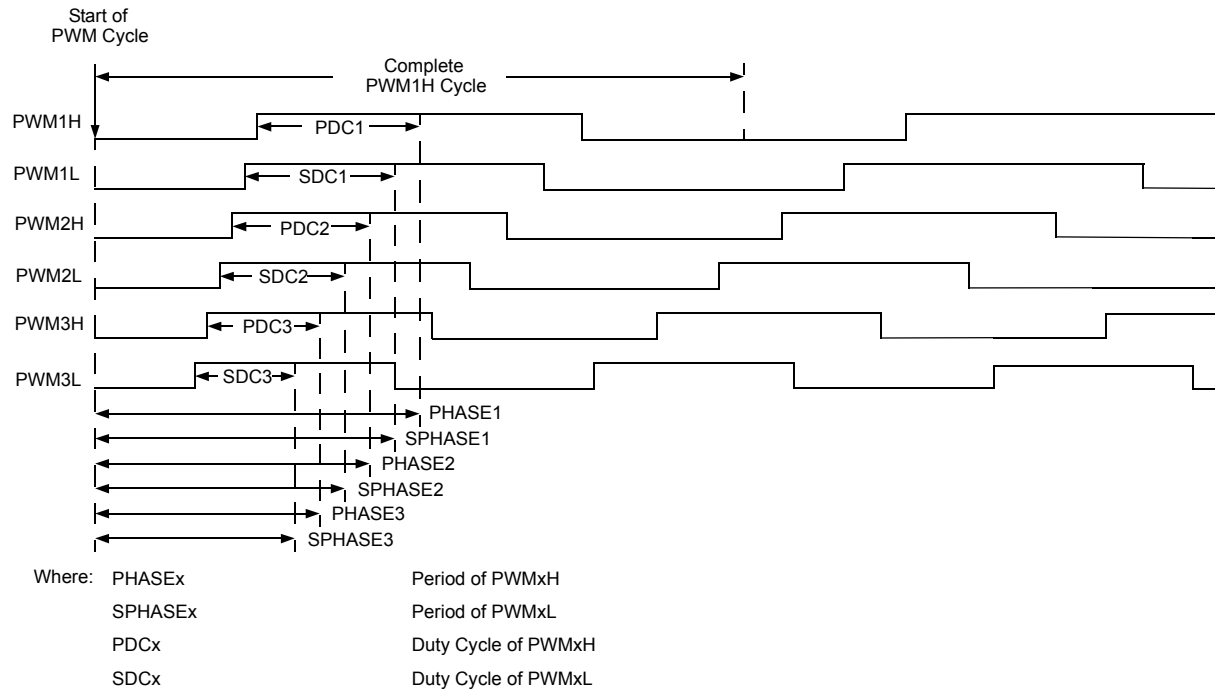


Where: PHASEx Period of PWMxH
SPHASEx Period of PWMxL
MDC Duty Cycle of PWMxH and PWMxL

Example 14-35: Independent PWM Mode - Master Duty Cycles, Independent Periods, No Phase-Shifting, Edge-Aligned

```
/* Set PWM Periods on PHASEx Registers */
PHASE1 = 800;
SPHASE1 = 900;
PHASE2 = 1000;
SPHASE2 = 1100;
PHASE3 = 1200;
SPHASE3 = 1300;

/* Set Duty Cycles */
MDC = 200;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 0;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 0;

/* Set PWM Mode to Independent */
IOCON1 = IOCON2 = IOCON3 = 0xCC00;

/* Set Independent Time Bases, Edge-Aligned Mode and Master Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0300;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

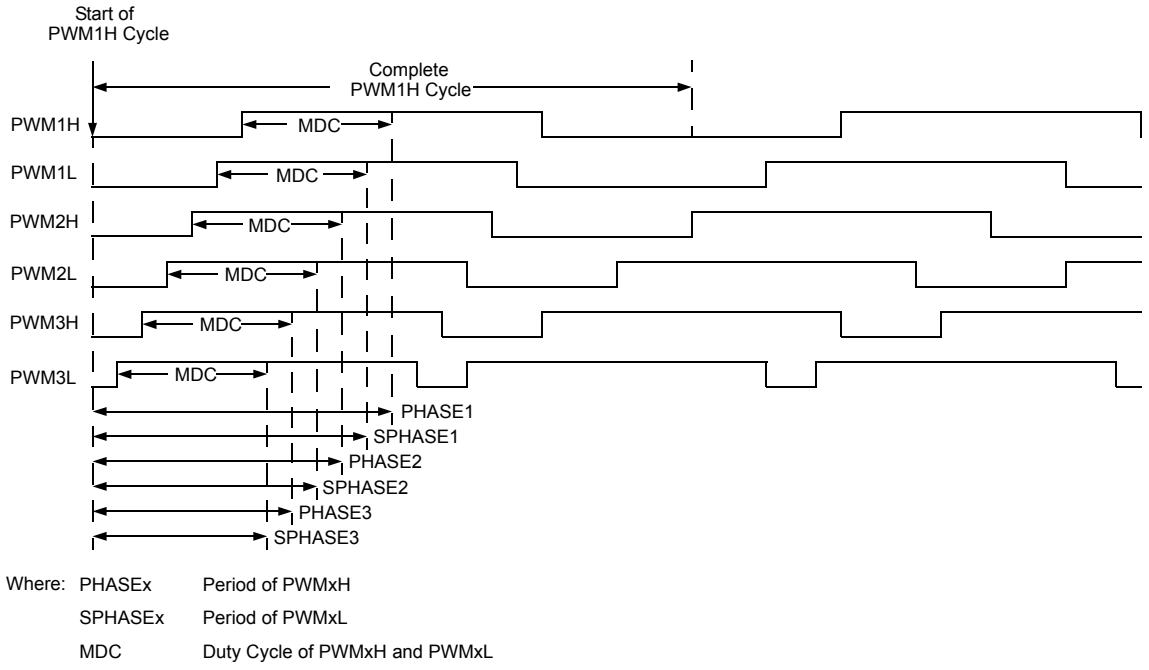
Figure 14-37: Independent PWM Mode - Independent Duty Cycles and Periods, No Phase-Shifting, Center-Aligned



Example 14-36: Independent PWM Mode - Independent Duty Cycles and Periods, No Phase-Shifting, Center-Aligned

```
/* Set PWM Periods on PHASEx Registers */
PHASE1 = 1300;
SPHASE1 = 1200;
PHASE2 = 1100;
SPHASE2 = 1000;
PHASE3 = 900;
SPHASE3 = 800;

/* Set Duty Cycles */
PDC1 = 700;
SDC1 = 600;
PDC2 = 500;
SDC2 = 400;
PDC3 = 300;
SDC3 = 200;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 0;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 0;

/* Set PWM Mode to Independent */
IOCON1 = IOCON2 = IOCON3 = 0xCC00;

/* Set Independent Time Bases, Center-Aligned Mode and Independent Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0204;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

Figure 14-38: Independent PWM Mode - Master Duty Cycles, Independent Periods, No Phase-Shifting, Center-Aligned



Example 14-37: Independent PWM Mode - Master Duty Cycles, Independent Periods, No Phase-Shifting, Center-Aligned

```
/* Set PWM Periods on PHASEx Registers */
PHASE1 = 1300;
SPHASE1 = 1200;
PHASE2 = 1100;
SPHASE2 = 1000;
PHASE3 = 900;
SPHASE3 = 800;

/* Set Duty Cycles */
MDC = 700;

/* Set Dead Time Values */
DTR1 = DTR2 = DTR3 = 0;
ALTDTR1 = ALTDTR2 = ALTDTR3 = 0;

/* Set PWM Mode to Independent */
IOCON1 = IOCON2 = IOCON3 = 0xCC00;

/* Set Independent Time Bases, Center-Aligned Mode and Master Duty Cycles */
PWMCON1 = PWMCON2 = PWMCON3 = 0x0304;

/* Configure Faults */
FCLCON1 = FCLCON2 = FCLCON3 = 0x0003;

/* 1:1 Prescaler */
PTCON2 = 0x0000;

/* Enable PWM Module */
PTCON = 0x8000;
```

Table 14-3 lists the PWM register functionality for Independent Output mode.

Table 14-3: Independent Output Mode Register Functionality

Function	Configuration in PWMCONx			Pins	
	MDCS	ITB	MTBS	PWMxH	PWMxL
PWM Duty Cycle	0	x	x	PDCx	SDCx
	1	x	x	MDC	MDC
PWM Phase Shift	x	0	x	PHASEx	SPHASEx
PWM Period	x	x	0	PTPER	PTPER
	x	x	1	STPER	STPER
	x	1	x	PHASEx	SPHASEx

Note: 'x' is a don't care

14.7 PWM GENERATOR

This section describes the functionality of the PWM generator.

14.7.1 PWM Period

The PWM period value defines the switching frequency of the PWM pulses. The PWM period value can be controlled by the Primary Master Time Base Period register (PTPER), or by the Independent Time Period PHASEx and SPHASEx registers for the respective primary and secondary PWM outputs.

The PWM period value can be controlled in two ways when the High-Speed PWM module operates in Independent Time Base mode:

- In some modes, the PHASEx register controls the PWM period value of the PWM output signals (PWMxH and PWMxL)
- In True Independent Output mode, the PHASEx register controls the PWM period value of the PWMxH output signal, and the SPHASEx register controls the PWM period value of the PWMxL output signal

Refer to **14.6 "PWM Operating Modes"**, for detailed information about various PWM modes and their features.

When the High-Speed PWM module operates in Master Time Base mode, the PTPER register holds the 16-bit value, which specifies the counting period for the PMTMR timer. When the High-Speed PWM module operates in Independent Time Base mode, the PHASEx and SPHASEx registers hold the 16-bit value that specifies the counting period for the PTMRx and STMRx timer, respectively. The timer period can be updated at any time by the user-assigned application. The PWM time period (PTPER) in Edge-Aligned PWM mode (CAM bit (PWMCONx<2>) is set to '0') can be determined by using the Equation 14-1.

Equation 14-1: PERIOD, PHASEx and SPHASEx Register Value Calculation for Edge-Aligned Mode

$$PTPER,\ PHASEx,\ SPHASEx = \frac{FOSC}{FPWM * PWM\ Input\ Clock\ Prescaler}$$

Where:
$FPWM$ = Desired PWM frequency
$FOSC$ = Oscillator output (120 MHz for 60 MIPS)
PWM Input Clock Prescaler = Value defined in the PCLKSEL<2:0> bits (PTCON2<2:0)

Based on Equation 14-1, while operating in the master time base (PTPER register) or the independent time base (PHASEx and SPHASEx registers), the register value to be loaded is shown in Example 14-38.

Example 14-38: PWM Time Period Calculation for Edge-Aligned Mode

$$PTPER = \frac{120MHz}{20kHz \times 1} = 6000$$

Where:
Desired PWM Switching Frequency = 20 kHz
PWM Input Clock Prescaler = 1:1
System Oscillator Frequency (FOSC) = 120 MHz

The PWM time period (PHASEx or SPHASEx) in Center-Aligned mode (CAM = 1 and ITB = 1), can be determined using Equation 14-2. The PTPER register does not represent the period in Center-Aligned mode, since this mode requires independent time bases, which are enabled by setting ITB = 1.

Equation 14-2: PHASEx or SPHASEx Register Value Calculation in Center-Aligned Mode

$$PHASEx, SPHASEx = \frac{FOSC}{FPWM \cdot PWM\ Input\ Clock\ Prescaler \cdot 2}$$

Based on Equation 14-2, when operating in independent time bases (PHASEx and SPHASEx registers), the register value to be loaded is shown in Example 14-39.

Example 14-39: PWM Time Period Calculation Example in Center-Aligned Mode

$$PHASEx, SPHASEx = \frac{120\ MHz}{20\ kHz \cdot 1 \cdot 2} = 3000$$

Where:
PWM Frequency (FPWM) = 20 kHz
PWM Input Clock Prescaler = 1:1
System Oscillator Frequency (FOSC) = 120 MHz

The maximum available PWM period resolution is TOSC for Edge-Aligned mode and TOSC * 2 in Center-Aligned mode. The PCLKSEL<2:0> bits (PTCON2<2:0>), determine the type of PWM clock. The timer/counter is enabled or disabled by setting or clearing the PTEN bit (PTCON<15>). The PMTMR timer can also be cleared using the PTEN bit.

If the EIPU bit (PTCON<10>) is set, the active master period register (an internal shadow register) is updated immediately instead of waiting for the PWM cycle to end. The EIPU bit affects the PMTMR master time base.

Example 14-40: Clock Prescaler Selection

```
/* Select PWM time base input clock prescaler */
/* Choose divide ratio of 1:2 */

PTCON2bits.PCLKSEL = 1;
```

Example 14-41: PWM Time Period Selection

```
/* Select Time Base Period Control */
/* Choose one of these options */

PWMCON1bits.ITB = 0;   /* PTPER provides the PWM time period value */
PWMCON1bits.ITB = 1;   /* PHASEx/SPHASEx provides the PWM time period value */
```

Example 14-42: PWM Time Period Initialization

```
/* Choose PWM time period based on FRC input clock */
/* PWM frequency is 100 kHz */
/* Choose one of the following options */

PTPER = 4808;

PHASEx = 4808;

SPHASEx = 4808;
```

14.7.2 PWM Duty Cycle Control

The duty cycle determines the period of time the PWM output should remain in the active state. Each duty cycle register allows a 16-bit duty cycle value to be specified. The duty cycle values can be updated at any time by setting the IUE bit (PWMCONx<0>). If the IUE bit is '0', the active register updates at the start of the next PWM cycle.

The Master Duty Cycle register (MDC) enables multiple PWM generators to share a common duty cycle register.The MDC register has an important role in Master Time Base mode.

In addition, each PWM generator has a Primary Duty Cycle register (PDCx) and a Secondary Duty Cycle register (SDCx) that provides separate duty cycles to each PWM.

14.7.2.1 MASTER DUTY CYCLE (MDC)

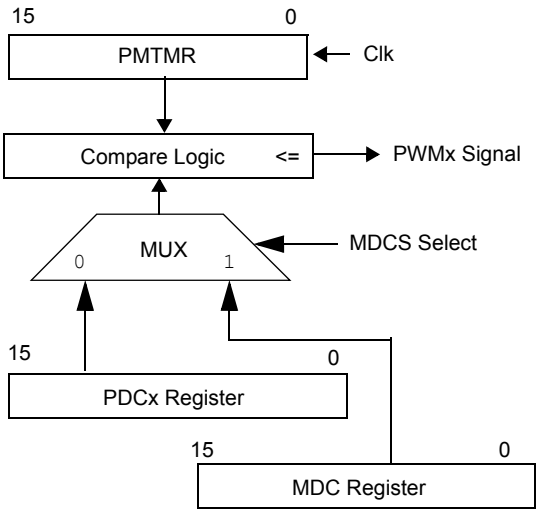
The master time base generator controls the master duty cycle. The MDCS bit (PWMCONx<8>), determines whether the duty cycle of each of the PWMxH and PWMxL outputs are controlled by the PWM Master Duty Cycle register (MDC) or the PWM Primary Duty Cycle (PDCx) and PWM Secondary Duty Cycle (SDCx) registers.

The MDC register enables sharing of the common duty cycle register among multiple PWM generators and saves the CPU overhead required in updating multiple duty cycle registers.

14.7.2.2 PRIMARY DUTY CYCLE (PDCx)

The independent time base controls the primary duty cycle when the ITB bit (PWMCONx<9>) is set to '1'. The PDCx register is an input register that provides the duty cycle value for the primary PWM output signal (PWMxH).

Figure 14-39: Primary Duty Cycle Comparison

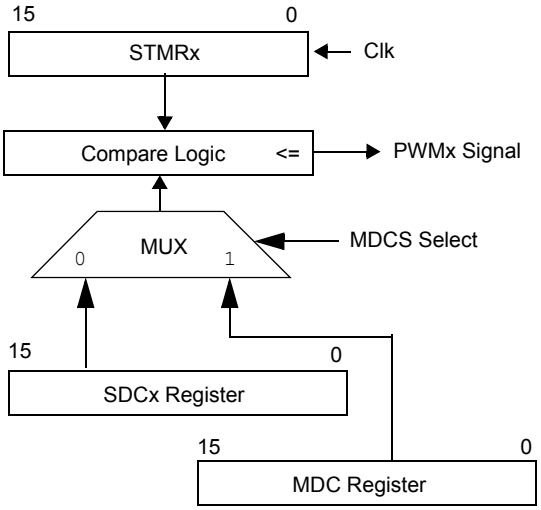


Note: In Independent Output mode, PDCx affects PWMxH only.

14.7.2.3 SECONDARY DUTY CYCLE (SDCx)

The independent time base in the PWMCONx register controls the Secondary Duty Cycle register (SDCx) when the ITB bit is set to '1'. The SDCx register is an input register that provides the duty cycle value for the secondary PWM output signal (PWMxL).

Figure 14-40: Secondary Duty Cycle Comparison



Note: In Independent Output mode, SDCx affects PWMxL only.

The duty cycle in Edge-Aligned mode can be determined by using Equation 14-3.

Equation 14-3: MDC, PDCx and SDCx Calculation for Edge-Aligned Mode

$$MDC,\ PDCx,\ and\ SDCx\ =\ \frac{F_{OSC}}{F_{PWM}\cdot\ PWM\ Input\ Clock\ Prescaler}\cdot\ Desired\ Duty\ Cycle$$

Where:
F_{PWM} = PWM Frequency
F_{OSC} = System Oscillator Output (120 MHz for 60 MIPS)
PWM Input Clock Prescaler = Value defined in the PCLKSEL<2:0> bits (PTCON<2:0>)
Desired Duty Cycle = Value between 0 and 1 for desired duty cycle

The duty cycle for Center-Aligned mode can be determined by using Equation 14-4.

Equation 14-4: MDC, PDCx and SDCx Calculation for Center-Aligned Mode

$$MDC,\ PDCx,\ and\ SDCx\ =\ \frac{F_{OSC}}{F_{PWM}\cdot\ PWM\ Input\ Clock\ Prescaler\cdot\ 2}\cdot\ Desired\ Duty\ Cycle$$

Where:
F_{PWM} = PWM Frequency
F_{OSC} = System Oscillator Output (120 MHz for 60 MIPS)
PWM Input Clock Prescaler = Value defined in the PCLKSEL<2:0> bits (PTCON<2:0>)
Desired Duty Cycle = Value between 0 and 1 for desired duty cycle

Note: If a duty cycle value is greater than or equal to the period value, a signal will have a duty cycle of 100 percent.

Based on Equation 14-3, when the master, independent primary, or independent secondary duty cycle is used, the register value is loaded in the MDC, PDCx, or SDCx register, respectively.

14.7.2.4 DUTY CYCLE RESOLUTION

The PWM duty cycle and period resolution are 8.3 ns per Least Significant Byte (LSB) with the PWM clock configured for the highest prescaler setting. The PWM duty cycle bit resolution for Edge-Aligned mode can be determined using Equation 14-5.

Equation 14-5: Bit Resolution Calculation for Edge-Aligned Mode

$$Bit\ Resolution = \log 2\left[\frac{FOSC}{FPWM \cdot PWM\ Input\ Clock\ Prescaler}\right]$$

The PWM duty cycle bit resolution for Center-Aligned mode can be determined using Equation.

Equation 14-6: Bit Resolution Calculation for Center-Aligned Mode

$$Bit\ Resolution = \log 2\left[\frac{FOSC}{FPWM \cdot PWM\ Input\ Clock\ Prescaler \cdot 2}\right]$$

Table 14-4 shows the duty cycle bit resolution versus PWM frequencies at the highest PWM clock frequency.

Table 14-4: PWM Duty Cycle Resolution and Maximum Frequency

PWM Mode	PWM Duty Cycle Resolution	Maximum PWM Frequency with Fosc = 120 MHz and 1:1 Clock Input
Edge-Aligned	16 bits	1.8 kHz
	15 bits	3.7 kHz
	14 bits	7.3 kHz
	13 bits	14.6 kHz
	12 bits	29.3 kHz
	11 bits	58.6 kHz
	10 bits	117.2 kHz
Center-Aligned	16 bits	0.9 kHz
	15 bits	1.8 kHz
	14 bits	3.7 kHz
	13 bits	7.3 kHz
	12 bits	14.6 kHz
	11 bits	29.3 kHz
	10 bits	58.6 kHz

At the highest clock frequency, the clock period is 8.33 ns when Tosc = 120 MHz. The PWM resolution becomes coarser by configuring other PWM clock prescaler settings.

Example 14-43: PWM Duty Cycle Selection

```
PWMCON1bits.MDCS = 0;          /* PDCx/SDCx provides duty cycle value */
PWMCON1bits.MDCS = 1;          /* MDC provides duty cycle value */
```

Example 14-44: PWM Duty Cycle Initialization

```
/* Initialize PWM Duty Cycle Value */

PDC1 = 2404;    /* Independent Primary Duty Cycle is 50% of the period */

SDC1 = 2404;    /* Independent Secondary Duty Cycle is 50% of the period */

MDC = 2404;     /* Master Duty Cycle is 50% of the period */
```

14.7.3 Dead Time Generation

The dead time refers to a programmable period of time (specified by the PWM Dead Time register (DTRx) or the PWM Alternate Dead Time register (ALTDTRx)), which prevents a PWM output from being asserted until its complementary PWM signal has been deasserted for the specified time.

The High-Speed PWM module has four dead time control units. Each dead time control unit has its own dead time value.

Dead time generation can be provided when any of the PWM I/O pin pairs are operating in Complementary Output mode. Many power converter circuits require dead time because 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 or the turn-on event of the other transistor.

The High-Speed PWM module provides positive as well as negative dead time. The positive dead time prevents overlapping of PWM outputs. Positive dead time generation is available for all output modes. Positive dead time circuitry works by blanking (gating) the leading edge of the PWM signal. Negative dead time is the forced overlap of the PWMxH and PWMxL signals. Negative dead time works when the extended time period of the currently active PWM output overlaps the PWM output that is just asserted. Certain converter techniques require a limited amount of current shoot-through.

Negative dead time is specified only for complementary PWM signals. Negative dead time does not apply to user or current-limit, or fault overrides. This mode can be implemented by using phase shift values in the PHASEx registers that shifts the PWM outputs so that the outputs overlap another PWM signal from a different PWM output channel. Negative dead time is only implemented for Edge-Aligned mode (CAM = 0).

The dead time logic acts as a gate and allows an asserted PWM signal or an override value to propagate to the output. The dead time logic never asserts a PWM output on its own initiative.

Figure 14-41: Dual Dead Time Waveforms for Edge-Aligned Mode

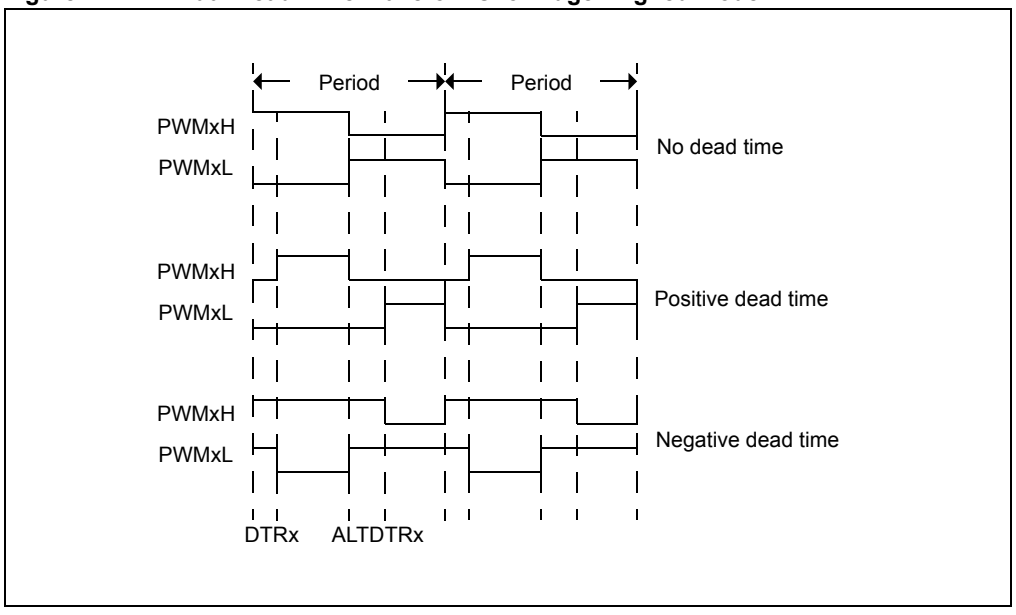
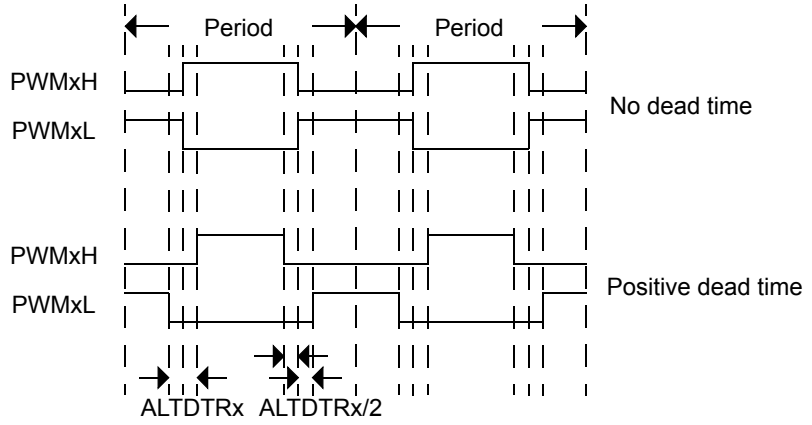


Figure 14-42: Dual Dead Time Waveforms for Center-Aligned Mode



Note: Dual dead times are not allowed in Center-Aligned mode.
Negative dead time is not supported in Center-Aligned mode.

The dead time feature can be disabled for each PWM generator. The dead time functionality is controlled by the DTC<1:0> bits (PWMCONx<7:6>).

14.7.4 Dead Time Generators

Each complementary output pair for the High-Speed PWM module has a 12-bit down counter 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 dead time timer generates the specific delay period.

The dead time logic monitors the rising and falling edges of the PWM signals. The dead time counters reset when the associated PWM signal is inactive, and start counting when the PWM signal is active. Any selected signal source that provides the PWM output signal is processed by the dead time logic.

The dead time can be determined using the formula shown in Equation 14-7.

Equation 14-7: Dead Time Calculation

$$ALTDTRx,\ DTRx = F_{OSC} * \frac{Desired\ Dead\ Time}{PWM\ Input\ Clock\ Prescaler}$$

Three Dead Time Control modes are available, which are described in the following sections.

14.7.4.1 POSITIVE DEAD TIME

The Positive Dead Time mode describes a period of time when both PWMxH and PWMxL outputs are not asserted. This mode is useful when the application designer needs to allocate time to disable some power transistors prior to enabling other transistors. This is similar to a “Break before Make” switch. When Positive Dead Time mode is specified in Edge-Aligned PWM mode (CAM = 0), the DTRx register specifies the dead time for the PWMxH output, and the ALTDTRx register specifies the dead time for the PWMxL output. When Center-Aligned mode is enabled (CAM = 1), the ALTDTR register specifies the dead time, while the DTRx register is ignored. These two modes are shown in Figure 14-43 and Figure 14-44.

Figure 14-43: Positive Dead Time in Edge-Aligned Mode and Complementary Mode

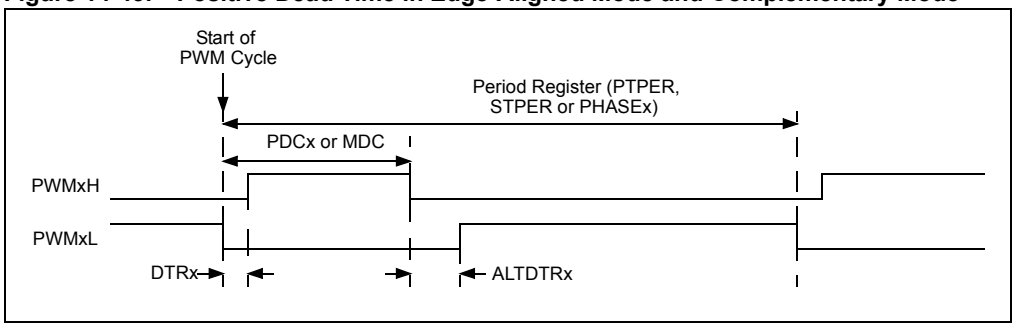
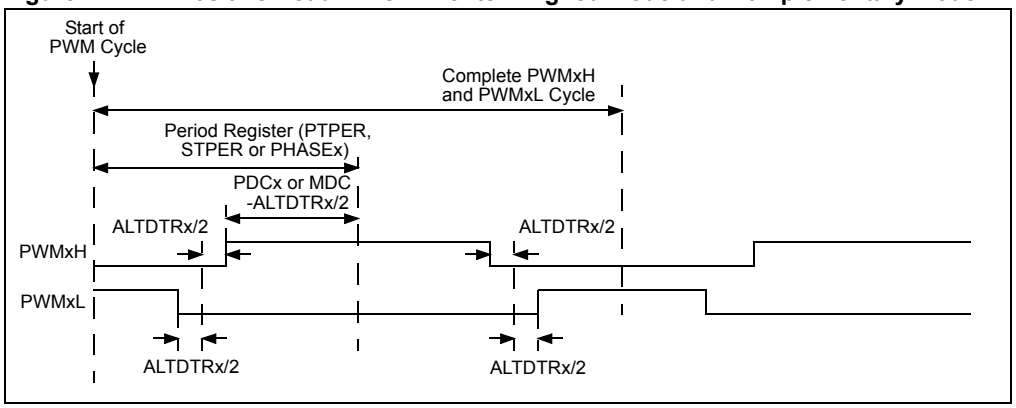


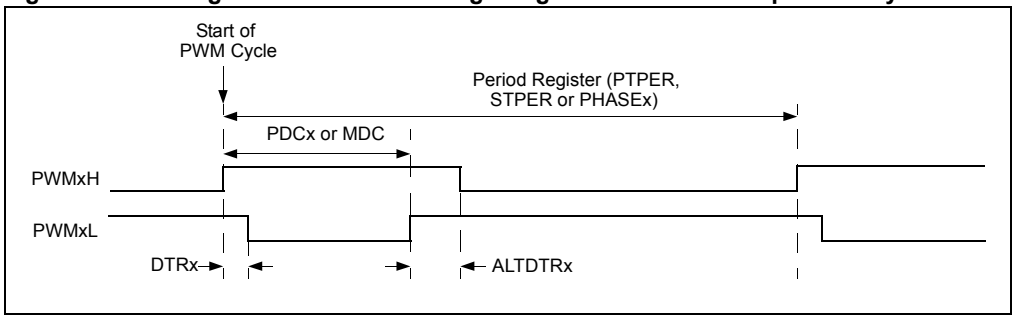
Figure 14-44: Positive Dead Time in Center-Aligned Mode and Complementary Mode



14.7.4.2 NEGATIVE DEAD TIME

The Negative Dead Time mode describes a period of time when both PWMxH and PWMxL outputs are asserted. This mode is useful in current fed topologies that need to provide a path for current to flow when the power transistors are switching. This is similar to a “Make before Break” switch. When Negative Dead Time mode is specified in Edge-Aligned mode, the DTRx register specifies the negative dead time for the PWMxL output, and the ALTDTRx register specifies the negative dead time for the PWMxH output. When Center-Aligned mode is enabled (CAM = 1), negative dead time is not supported, and the ALTDTRx and DTRx registers are ignored. This mode is shown in Figure 14-45.

Figure 14-45: Negative Dead Time in Edge-Aligned Mode and Complementary Mode



14.7.4.3 DEAD TIME COMPENSATION

When dead time is applied to the PWM signals in AC motor control applications, the transistors are disabled. During dead time, motor current continues to flow through the free-wheeling diodes, but the applied voltage is zero. The zero applied voltage during dead time causes a distortion of the desired voltage waveform and subsequently a motor current distortion. This distortion causes torque variations that can affect the stability of the control system, and the performance of the motor.

Dead Time Compensation mode enables external signals (DTCMP1 to DTCMP7) to modify the duty cycle to overcome motor current distortion caused by the dead time.

When Dead Time Compensation mode is selected via the DTC<1:0> bits (PWMCONx<7:6>), an external input signal (DTCMPx) will cause the value in the DTRx register to be added to, or subtracted from, the duty cycle specified by the PDCx or MDC registers. The ALTDTRx register will specify the dead time for both the PWMxH and PWMxL output signals.

Dead time compensation is available only for Complementary PWM output mode with Positive Dead Time mode. Negative dead times or any other PWM output mode are not supported by Dead Time Compensation mode.

Figure 14-46: Dead Time Compensation in Edge-Aligned Mode (DTCMPx Pin = 0 and DTCP = 0 or DTCMPx Pin = 1 and DTCP = 1)

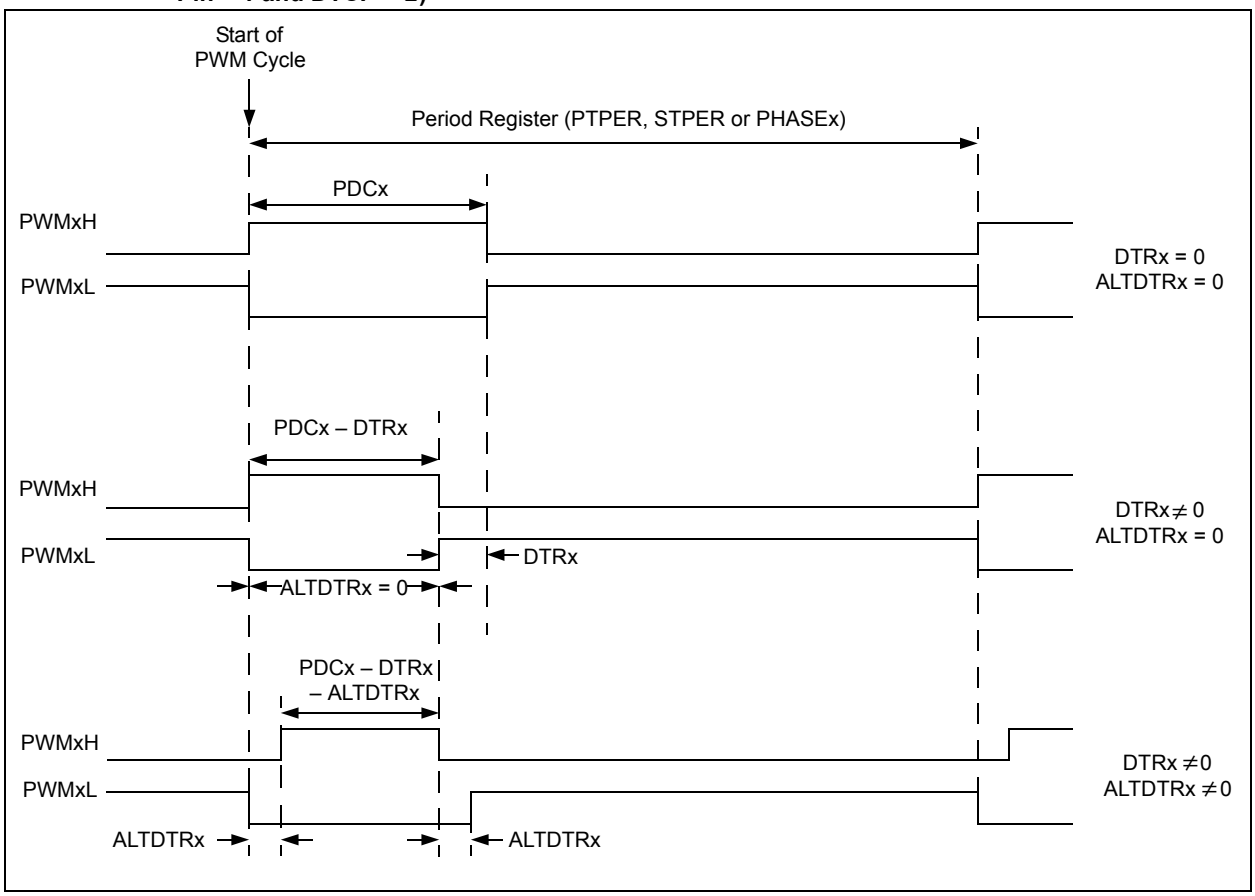


Figure 14-47: Dead Time Compensation in Edge-Aligned Mode (DTCMPx Pin = 1 and DTCP = 0 or DTCMPx Pin = 0 and DTCP = 1)

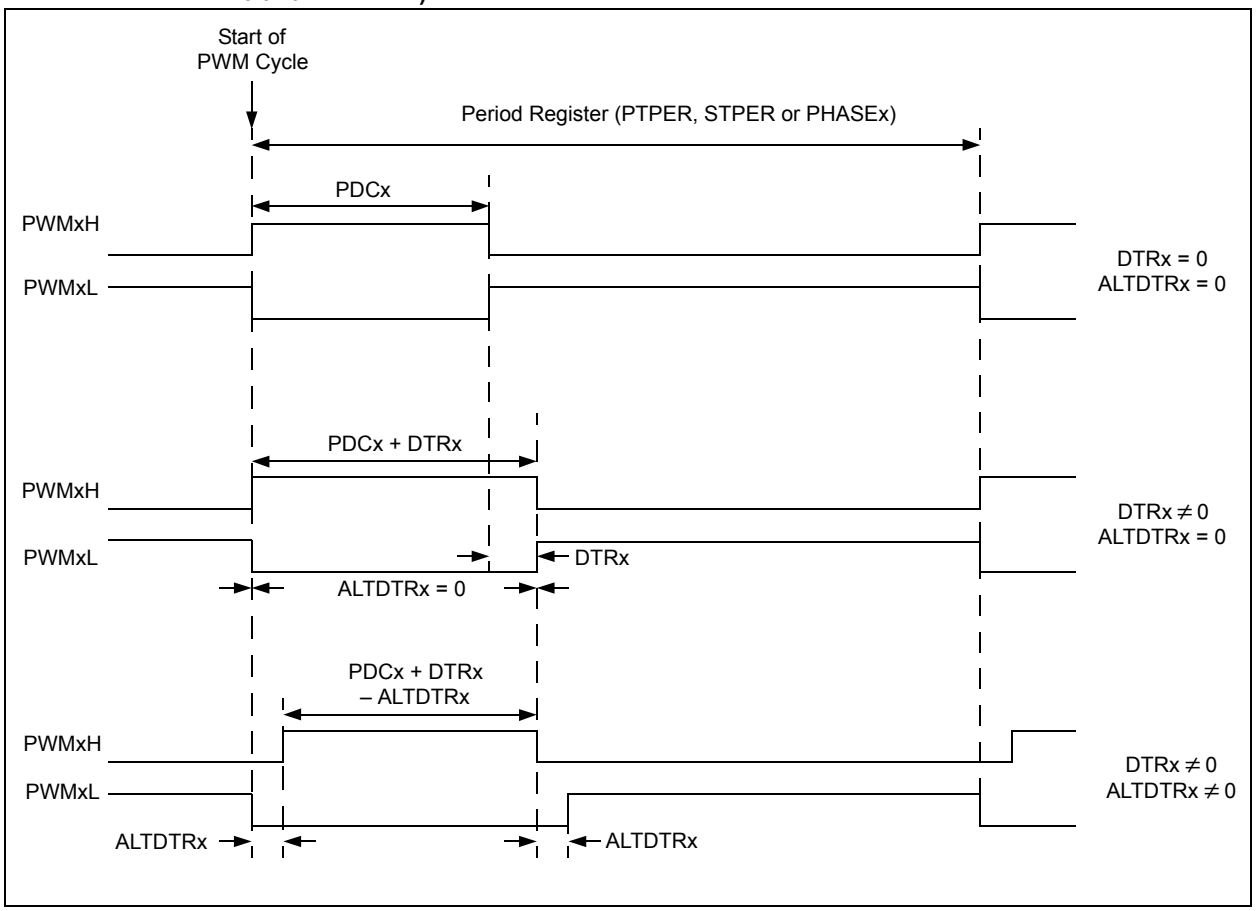


Figure 14-48: Dead Time Compensation in Center-Aligned Mode (DTCMPx Pin = 0 and DTCP = 0 or DTCMPx Pin = 1 and DTCP = 1)

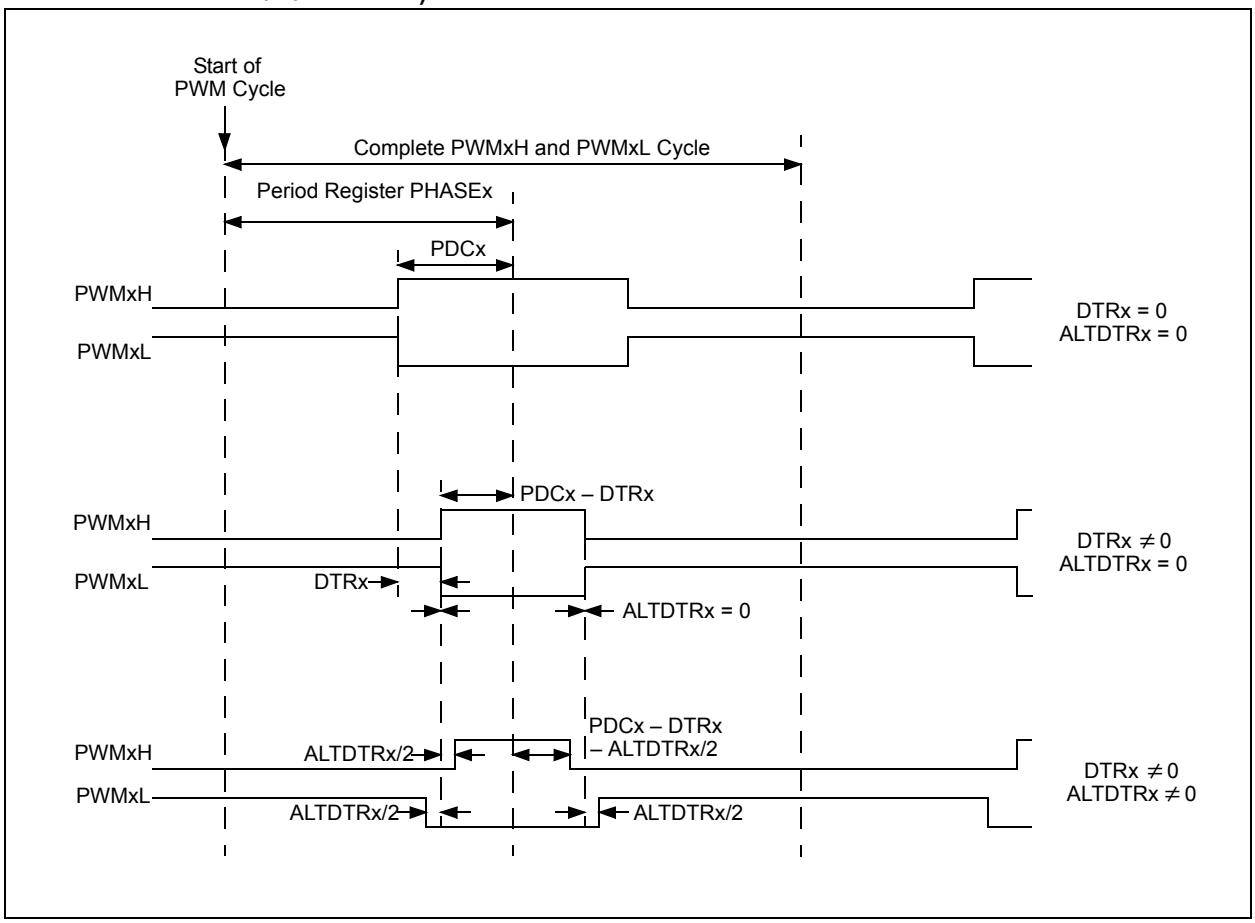
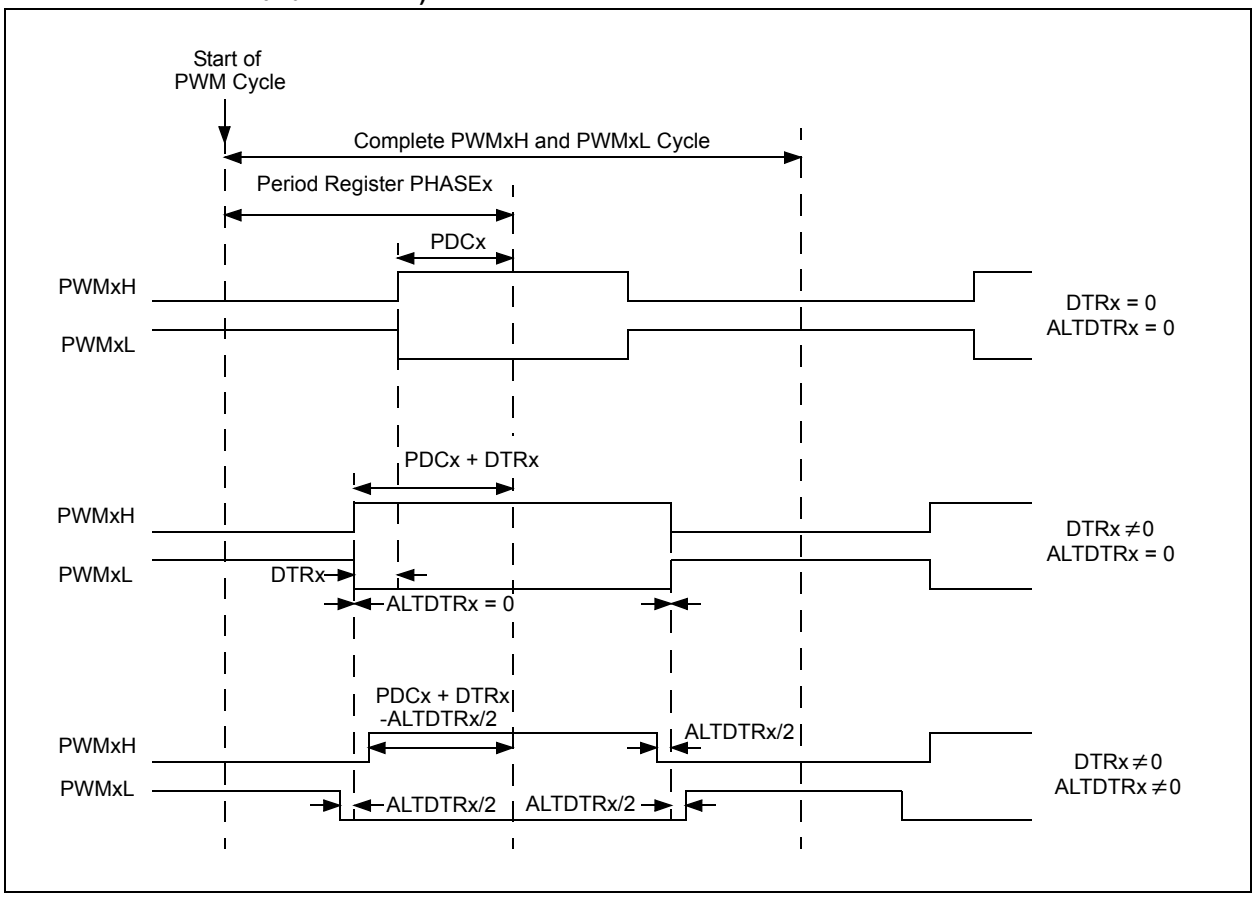


Figure 14-49: Dead Time Compensation in Center-Aligned Mode (DTCMPx Pin = 0 and DTCP = 0 or DTCMPx Pin = 1 and DTCP = 1)



The dead time compensation external input control signal, DTCMPx, is sampled at the beginning of each dead time period. This sampling process ensures that the control signal does not change midway through the dead time generation process. The DTCMPx signal determines whether the duty cycle will be increased or decreased by the amount specified in the DTRx register.

14.7.4.4 DEAD TIME DISABLED

The dead time logic can be disabled per PWM generator. The dead time functionality is controlled by the DTC<2:0> bits (PWMCONx<7:6>).

14.7.5 Dead Time Ranges

The dead time duration provided by each dead time unit is set by specifying an unsigned value in the DTRx and ALTDTRx registers. At maximum operating clock frequency with a TOSC duty cycle resolution, the dead time resolution is TOSC (8.33 ns at 120 MHz of FOSC). At the highest PWM resolution, the maximum dead time value is $(2^{14}-1)$TOSC = 136.5 µs when operating at 120 MHz FOSC.

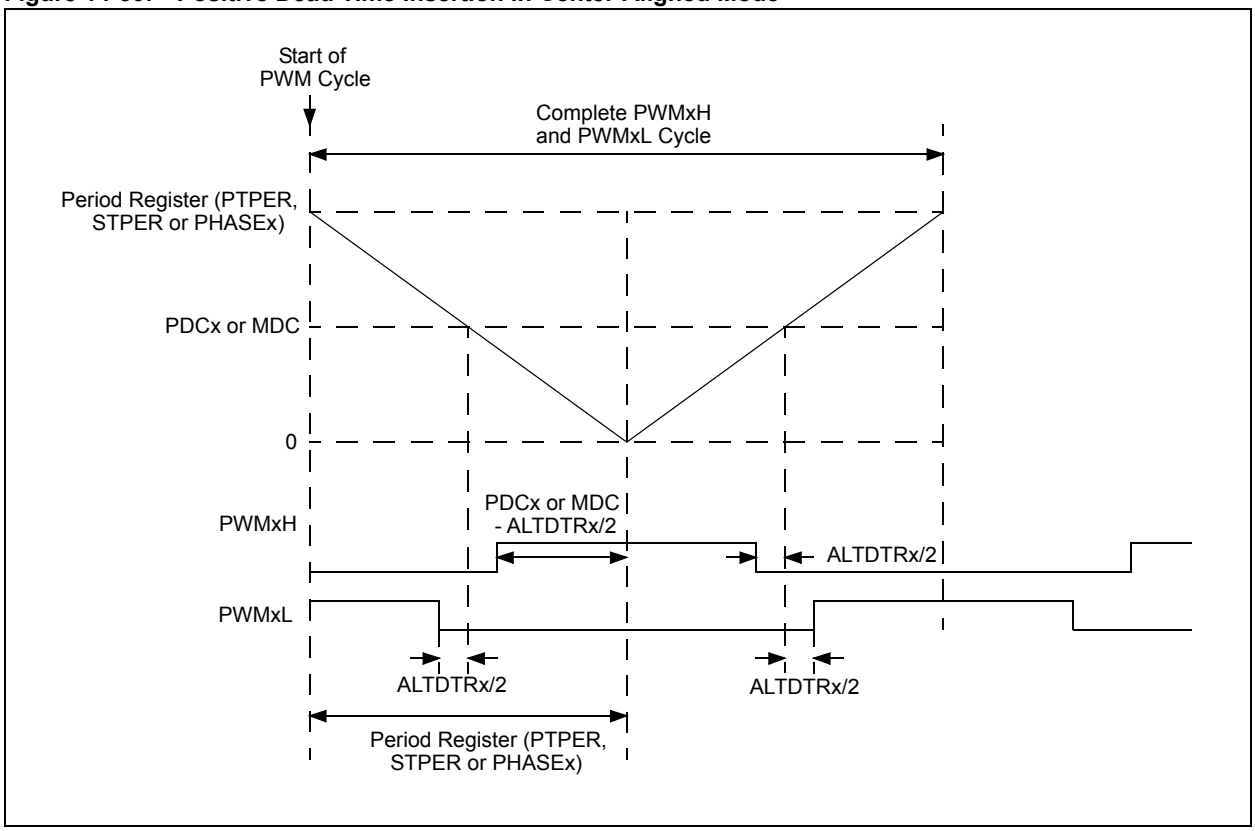
14.7.6 Dead Time Distortion

For duty cycle values near 0% or 100%, the PWM signal becomes nonlinear if dead time is active. For any duty cycle value less than the dead time, the PWM output is zero. For duty cycle values greater than “100% - dead time”, the PWM output is the same as if the duty cycle is “100% - dead time”.

14.7.7 Dead Time Insertion in Center-Aligned Mode

While using Center-Aligned mode and complementary PWM, only the ALTDTRx register should be used for dead time insertion. The dead time is inserted in the PWM waveform as shown in Figure 14-50.

Figure 14-50: Positive Dead Time Insertion in Center-Aligned Mode



14.7.8 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 SPHASEx register determines the relative phase shift between PWMxL and the master time base. The contents of the PHASEx register are used as an initialization value for the PTMRx register, and the SPHASEx register contents are used as an initialization value for the STMRx register.

Figure 14-51 and Figure 14-52 provide example waveforms for phase shifting in Complementary mode and Independent Output mode, respectively.

Figure 14-51: Phase Shifting (Complementary Mode)

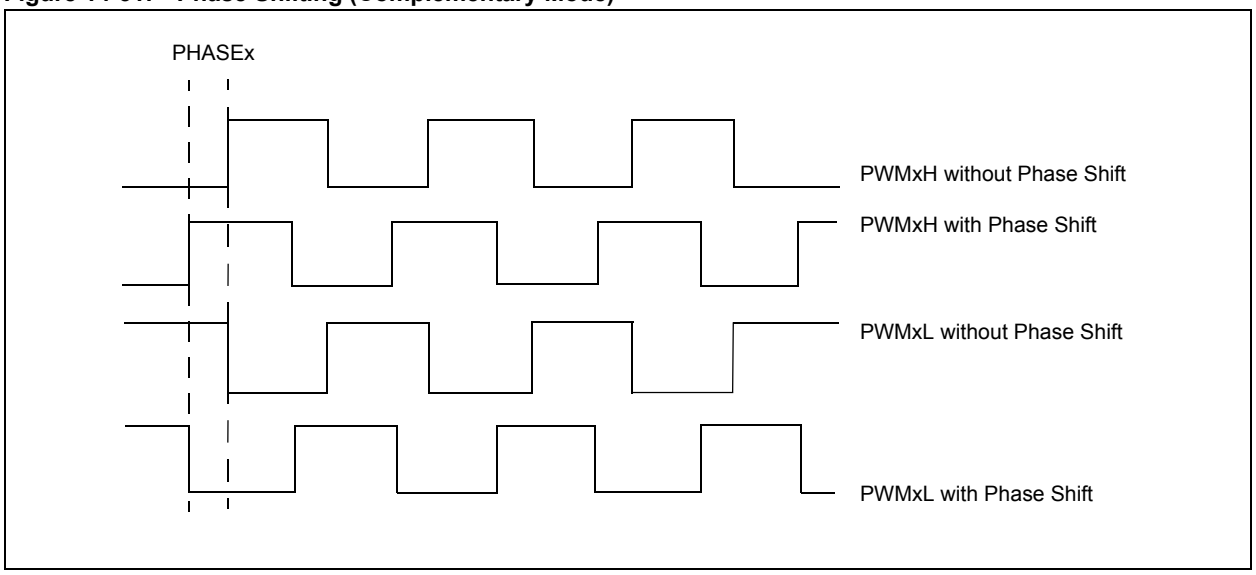
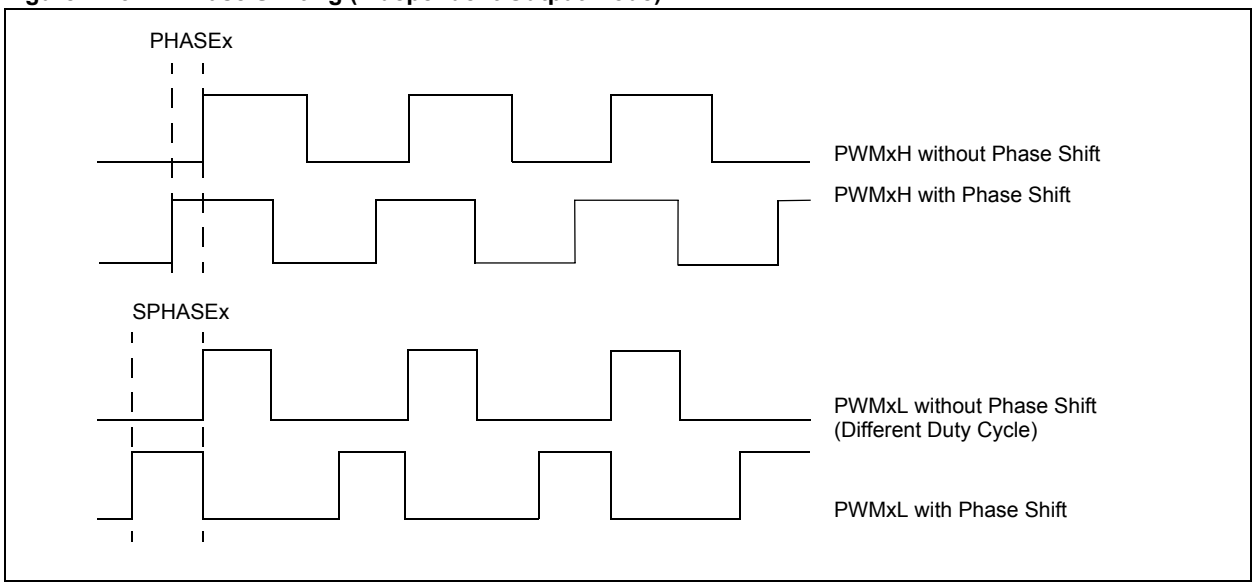


Figure 14-52: Phase Shifting (Independent Output Mode)

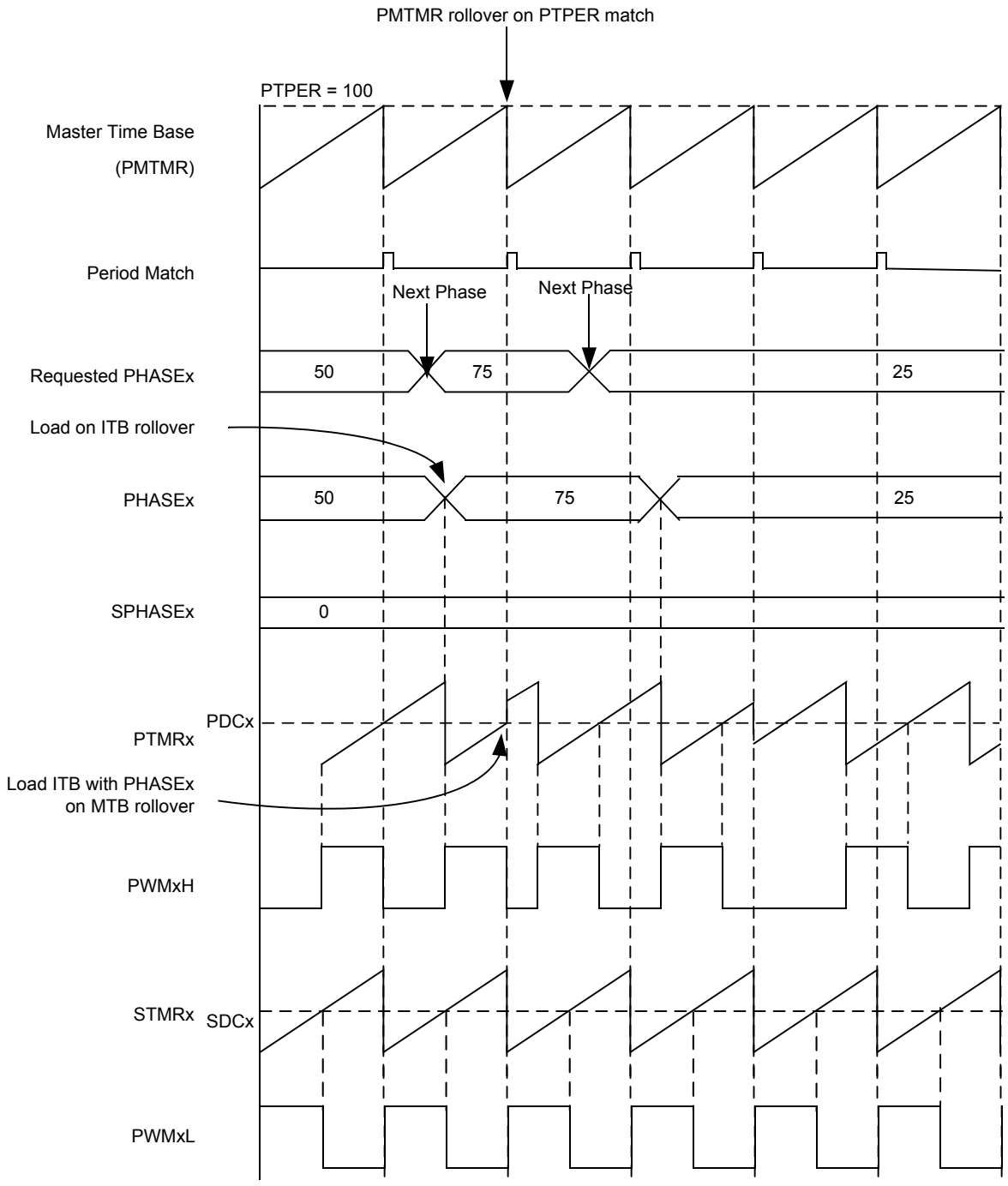


In addition, there are two shadow registers for the PHASEx and SPHASEx registers that are updated whenever new values are written by the user-assigned application. These values are transferred from the shadow registers to the PHASEx and SPHASEx registers on an independent time base Reset. The actual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

Figure 14-53 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 or SPHASEx 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 14-53: Phase Shift Waveforms



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

Example 14-45: PWM Phase Shift Initialization

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

PHASEx = 100;      /* Primary phase shift value */

SPHASEx = 100;     /* Secondary phase shift value */
```

The bit resolution of PWM duty cycle, phase, and dead time with respect to different input clock prescaler selections are shown in Table 14-5.

Table 14-5: Duty Cycle, Phase, Dead Time Bit Resolution vs. Prescaler Selection

PWM Clock Prescaler	Bit Resolution						
	64 * Tosc	32 * Tosc	16 * Tosc	8 * Tosc	4 * Tosc	2 * Tosc	Tosc
1:1	bit 6	bit 5	bit 6	bit 3	bit 2	bit 1	bit 0
1:2	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	—
1:4	bit 4	bit 3	bit 2	bit 1	bit 0	—	—
1:8	bit 3	bit 2	bit 1	bit 0	—	—	—
1:16	bit 2	bit 1	bit 0	—	—	—	—
1:32	bit 1	bit 0	—	—	—	—	—
1:64	bit 0	—	—	—	—	—	—

14.8 PWM TRIGGER

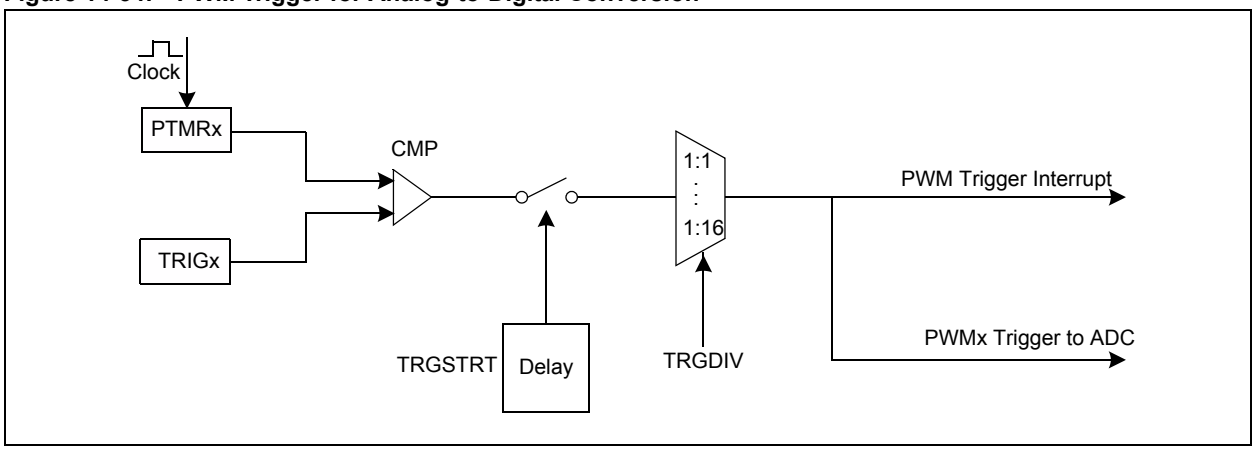
For the ADC module, the TRIGx register specifies the triggering point for the PWMxH and PWMxL outputs, respectively. An ADC trigger signal will be generated when the independent time base counter (PTMRx) register value matches with the specified TRIGx register value.

The TRGDIV<3:0> bits in the TRGCONx register act as a postscaler for the TRIGx register to generate ADC triggers. This allows the trigger signal to the ADC to be generated once for every 1, 2, 3.... and 16 trigger events. These bits specify how frequently the ADC trigger is generated.

Each PWM generator has TRGSTRT<5:0> bits (TRGCONx<5:0>) that specify how many PWM cycles to wait before generating the first ADC trigger.

Figure 14-54 shows the logic for ADC triggering by the High-Speed PWM module.

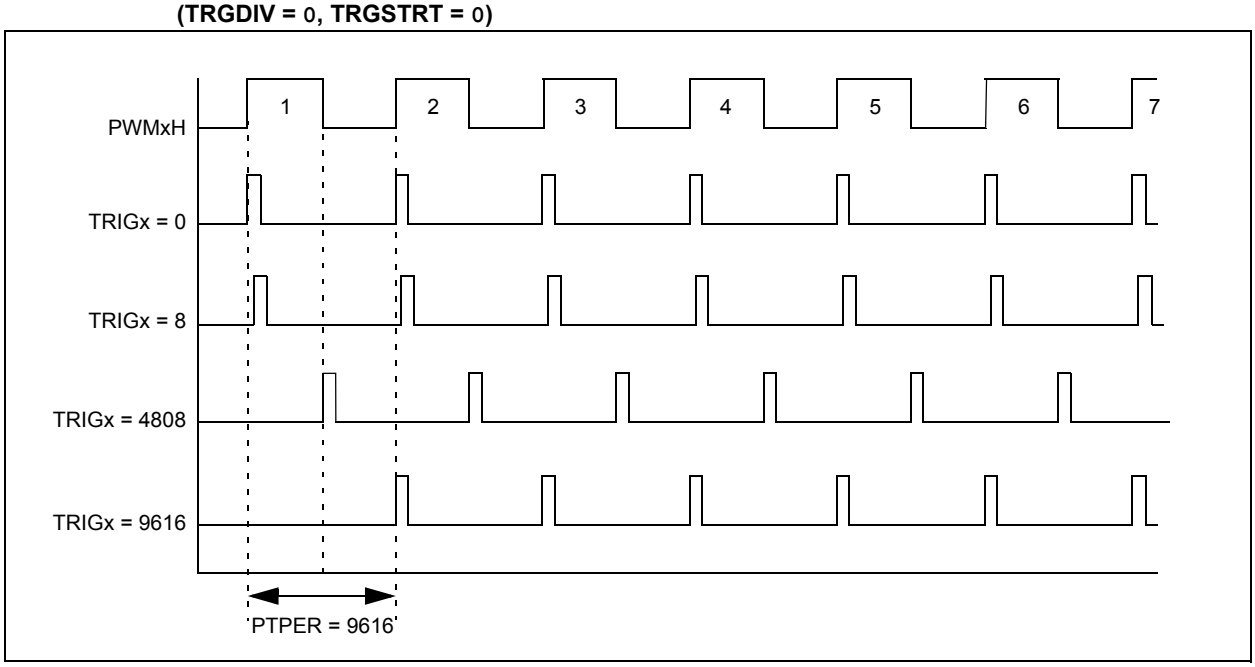
Figure 14-54: PWM Trigger for Analog-to-Digital Conversion



Depending on the settings of the TRGDIV<3:0> and TRGSTRT<5:0> bits, triggers are generated at different PWM intervals, as shown in Figure 14-55 through Figure 14-62.

Note: A trigger can only be generated on the first PWM interval when the TRGDIV<3:0> bits are set to '0'.

Figure 14-55: PWM Trigger Signal in Relation to the PWM Output in Edge-Aligned Mode (TRGDIV = 0, TRGSTRT = 0)



14 High-Speed PWM

Figure 14-56: PWM Trigger Signal in Relation to the PWM Output in Edge-Aligned Mode (TRGDIV = 0, TRGSTRT = 1)

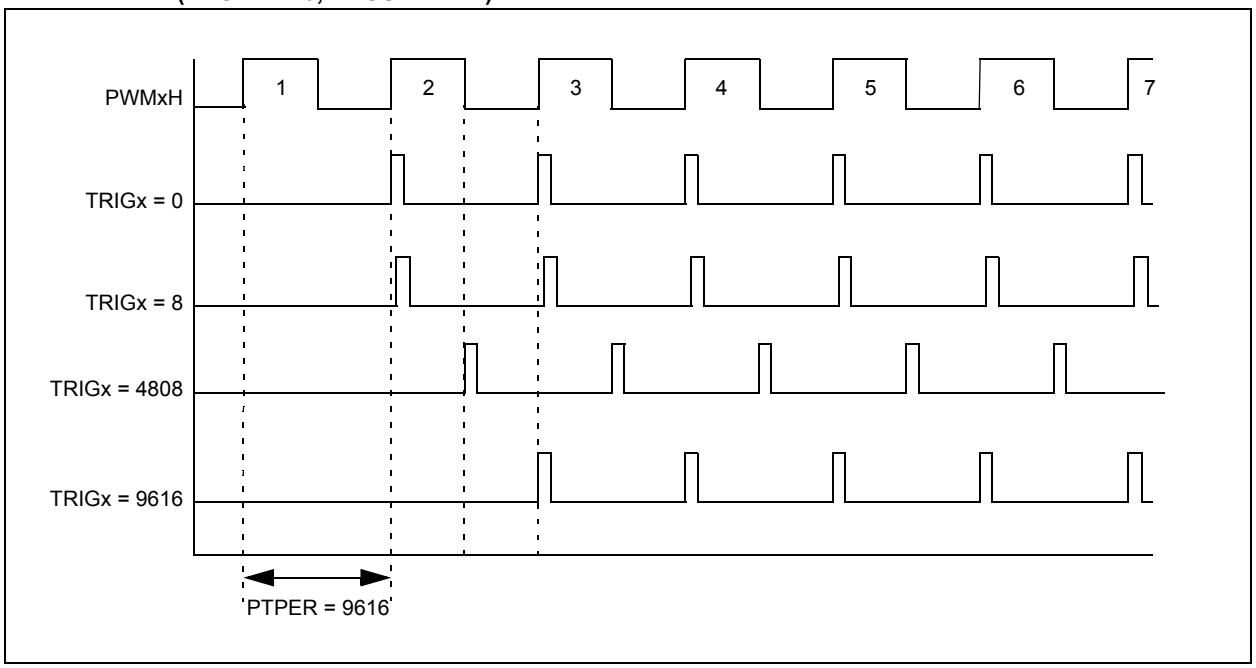


Figure 14-57: PWM Trigger Signal in Relation to the PWM Output in Edge-Aligned Mode (TRGDIV = 0, TRGSTRT = 2)

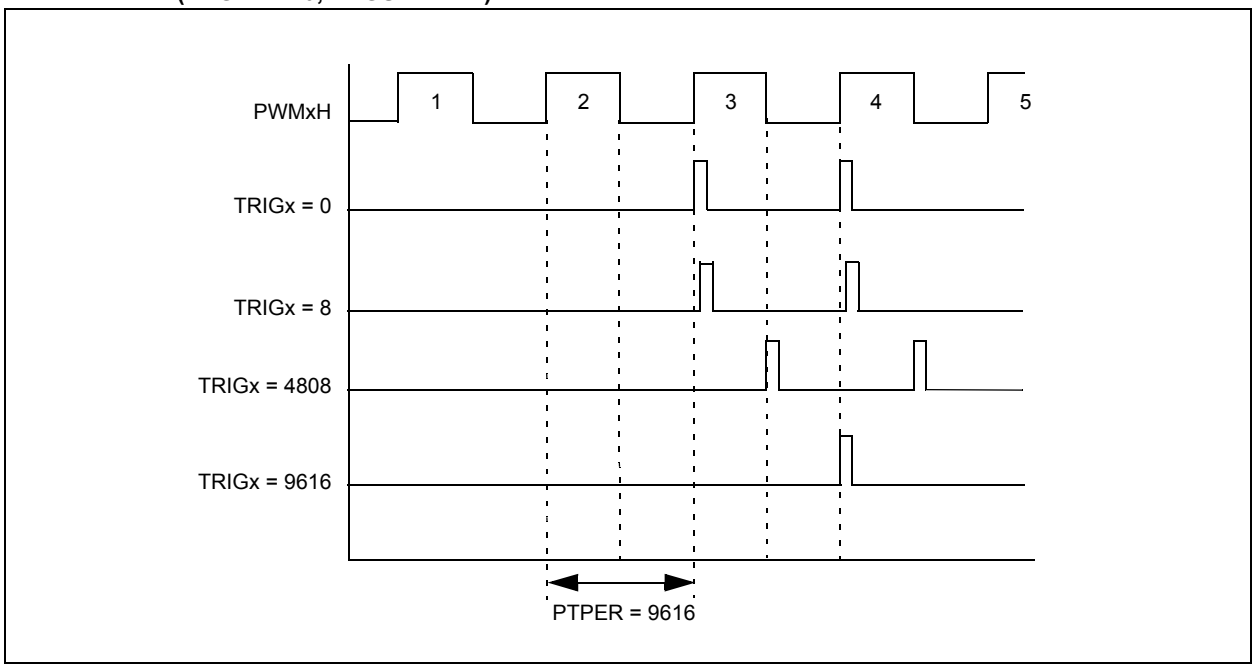


Figure 14-58: PWM Trigger Signal in Relation to the PWM Output in Edge-Aligned Mode (TRGDIV = 1, TRGSTRT = 0)

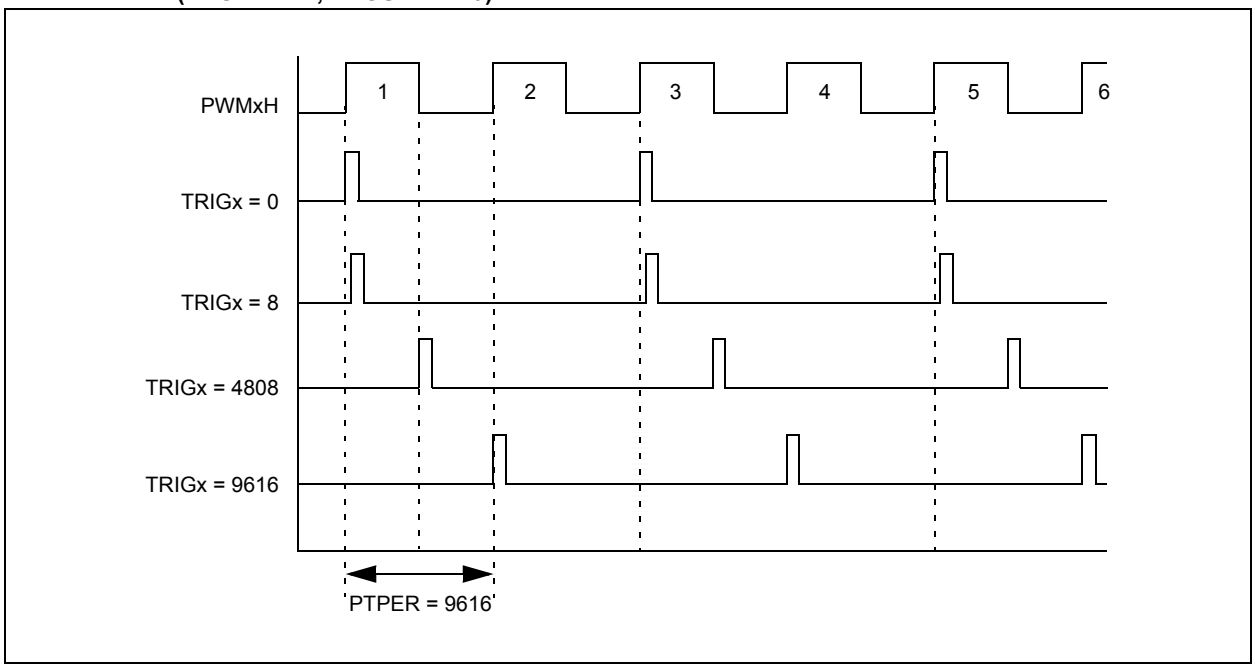


Figure 14-59: PWM Trigger Signal in Relation to the PWM Output in Edge-Aligned Mode (TRGDIV = 1, TRGSTRT = 1)

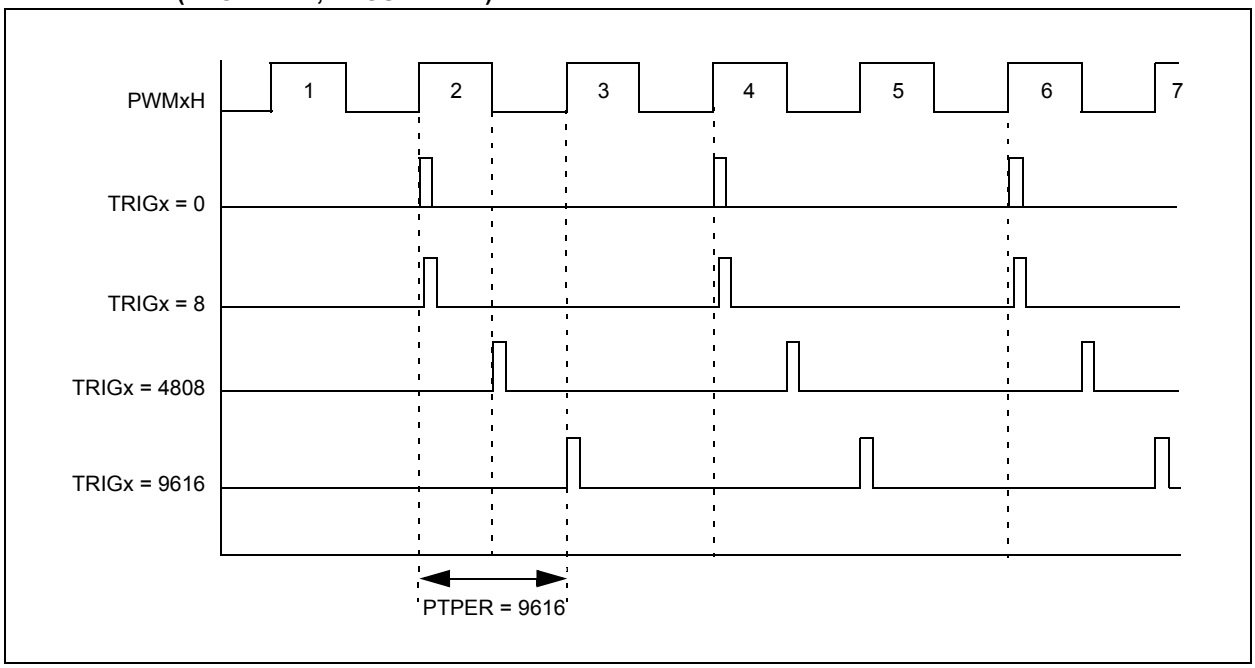


Figure 14-60: PWM Trigger Signal in Relation to the PWM Output in Edge-Aligned Mode (TRGDIV = 2, TRGSTRT = 0)

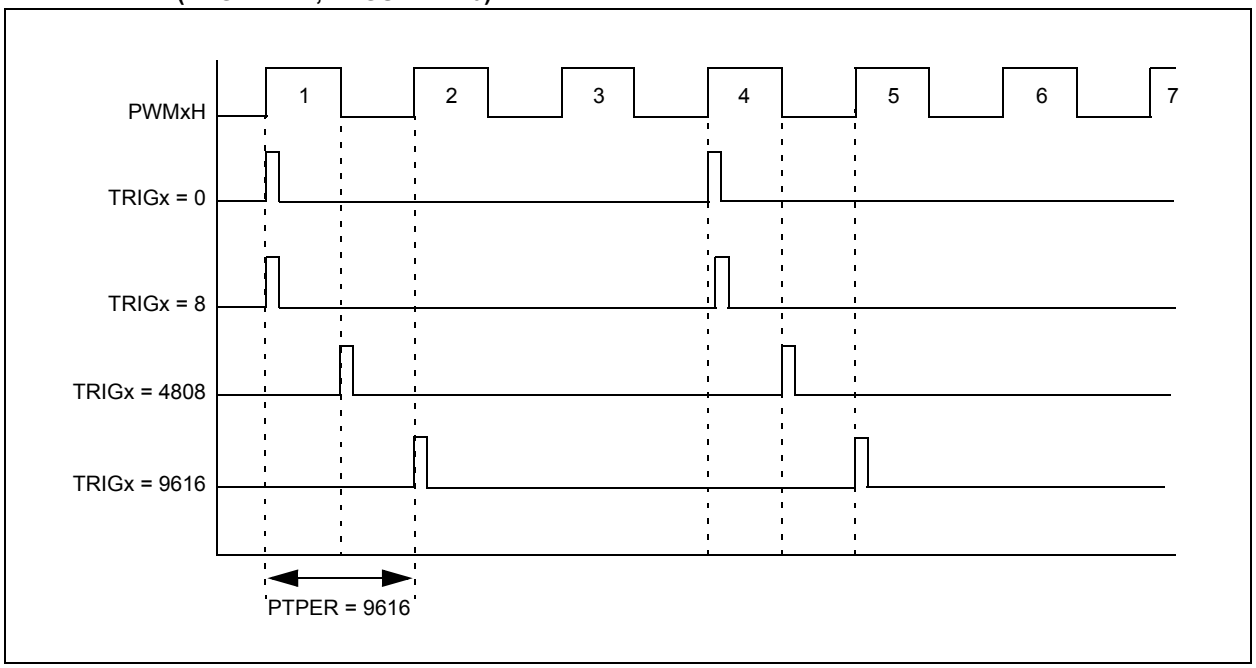


Figure 14-61: PWM Trigger Signal in Relation to the PWM Output in Edge-Aligned Mode (TRGDIV = 2, TRGSTRT = 2)

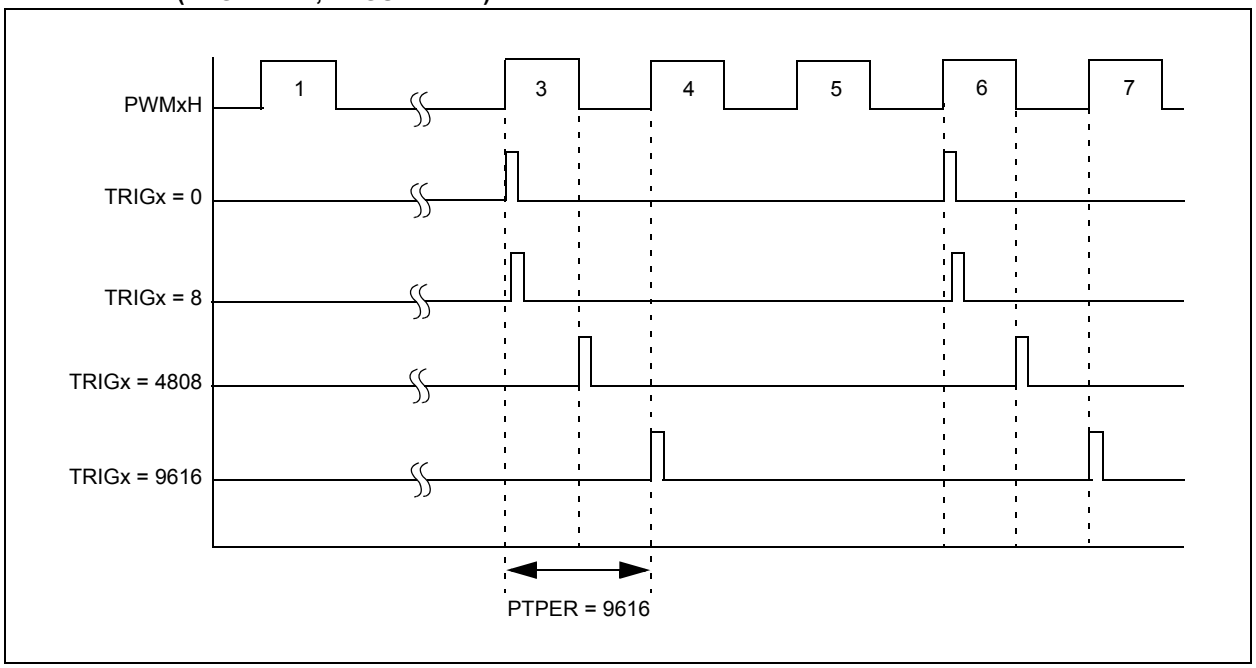


Figure 14-62: PWM Trigger Signal in Relation to the PWM Output in Edge-Aligned Mode (TRGDIV = 2, TRGSTRT = 3)

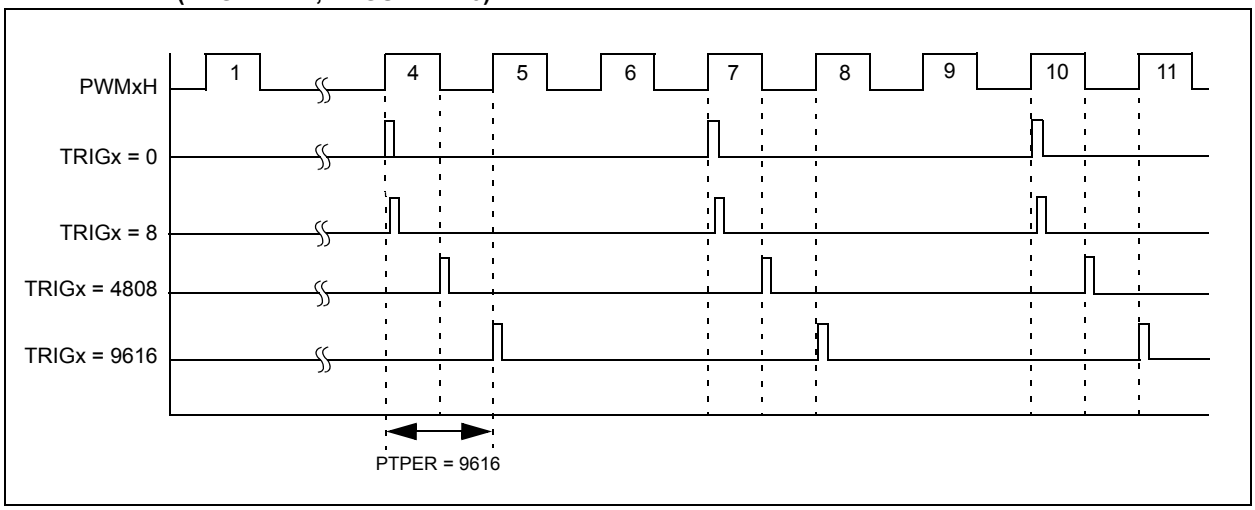


Figure 14-63: Trigger Signal in Relation to PWM Output in Center-Aligned Mode (TRGDIV = 0, TRGSTRT = 0)

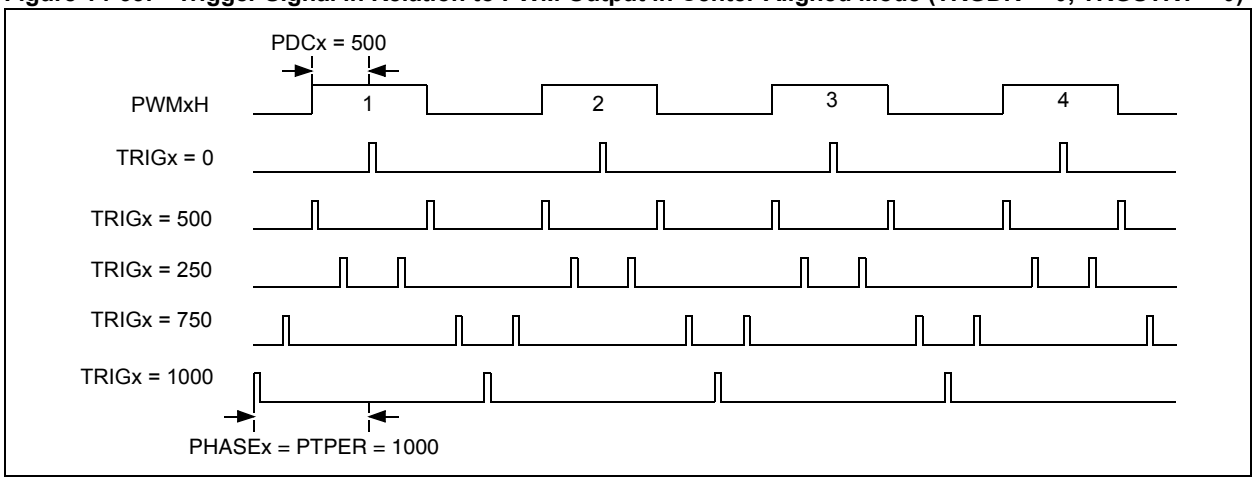


Figure 14-64: Trigger Signal in Relation to PWM Output in Center-Aligned Mode (TRGDIV = 0, TRGSTRT = 1)

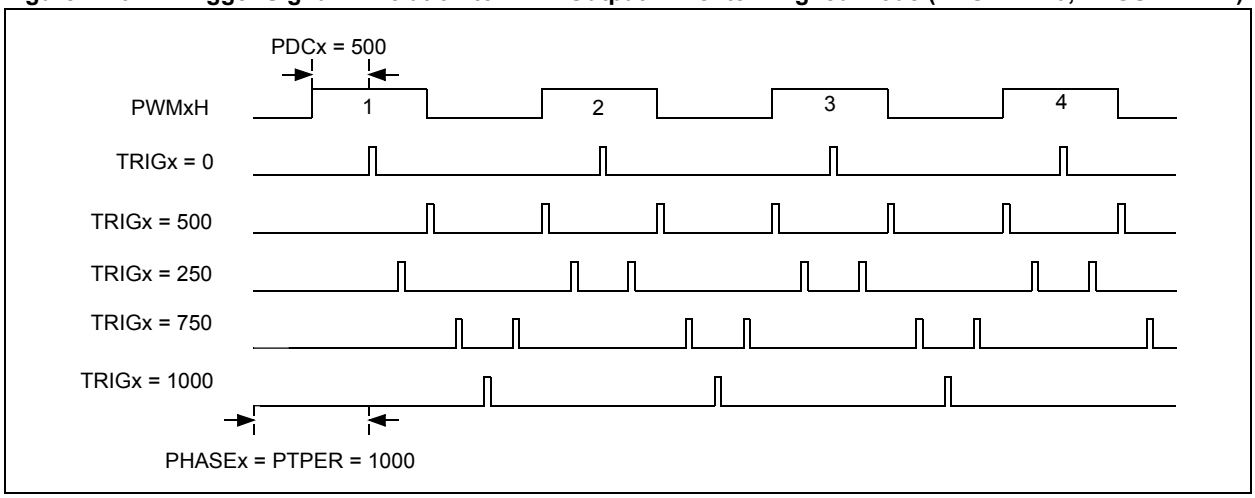


Figure 14-65: Trigger Signal in Relation to PWM Output in Center-Aligned Mode (TRGDIV = 0, TRGSTRT = 2)

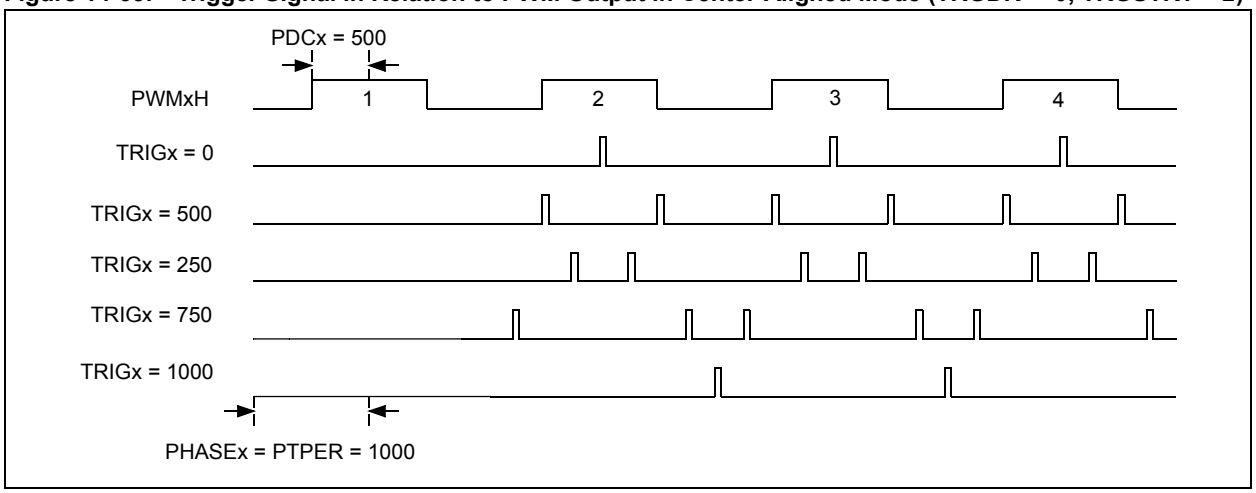


Figure 14-66: Trigger Signal in Relation to PWM Output in Center-Aligned Mode (TRGDIV = 1, TRGSTRT = 0)

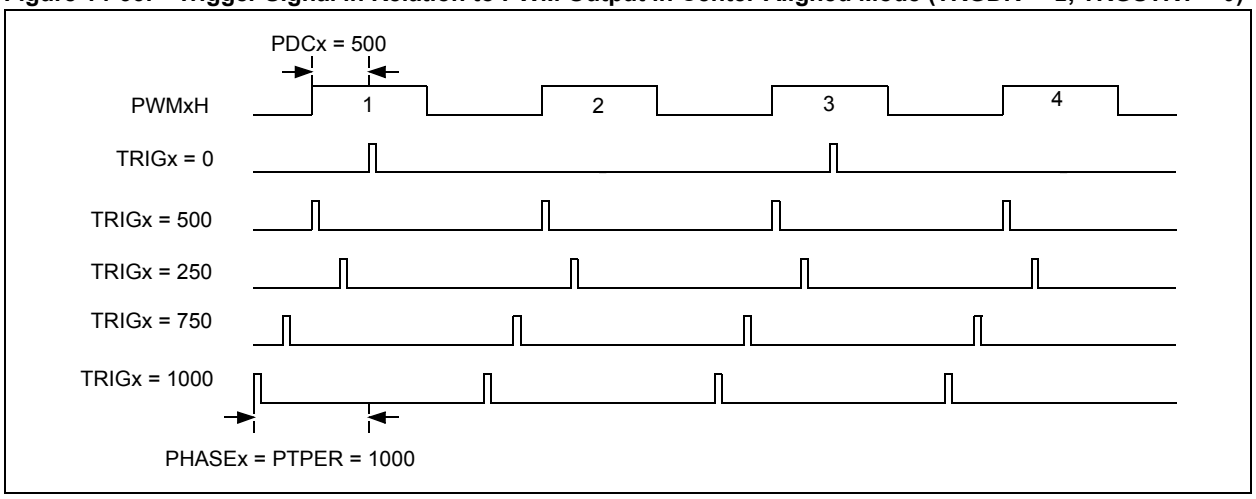
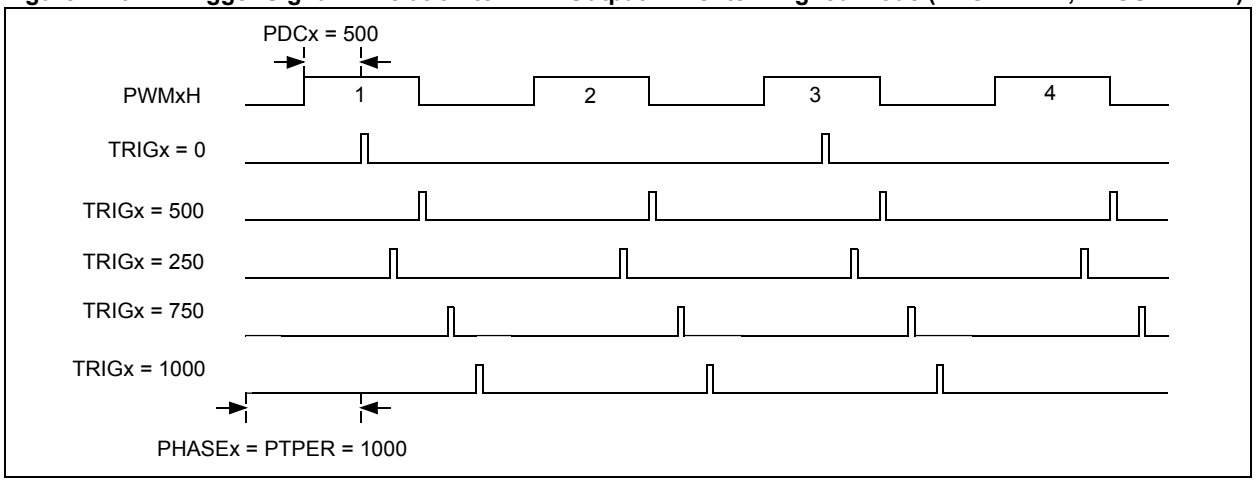


Figure 14-67: Trigger Signal in Relation to PWM Output in Center-Aligned Mode (TRGDIV = 1, TRGSTRT = 1)



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

If ADC triggers are generated at a rate faster than the rate that the ADC can process, the operation may result in loss of some samples. However, the user-assigned application can ensure that the time it provides is enough to complete an ADC operation within a single PWM cycle.

The trigger pulse is generated regardless of the state of the TRGIEN bit (PWMCONx<10>). If the TRGIEN bit (PWMCONx<10>) is set to '1', an interrupt request is generated.

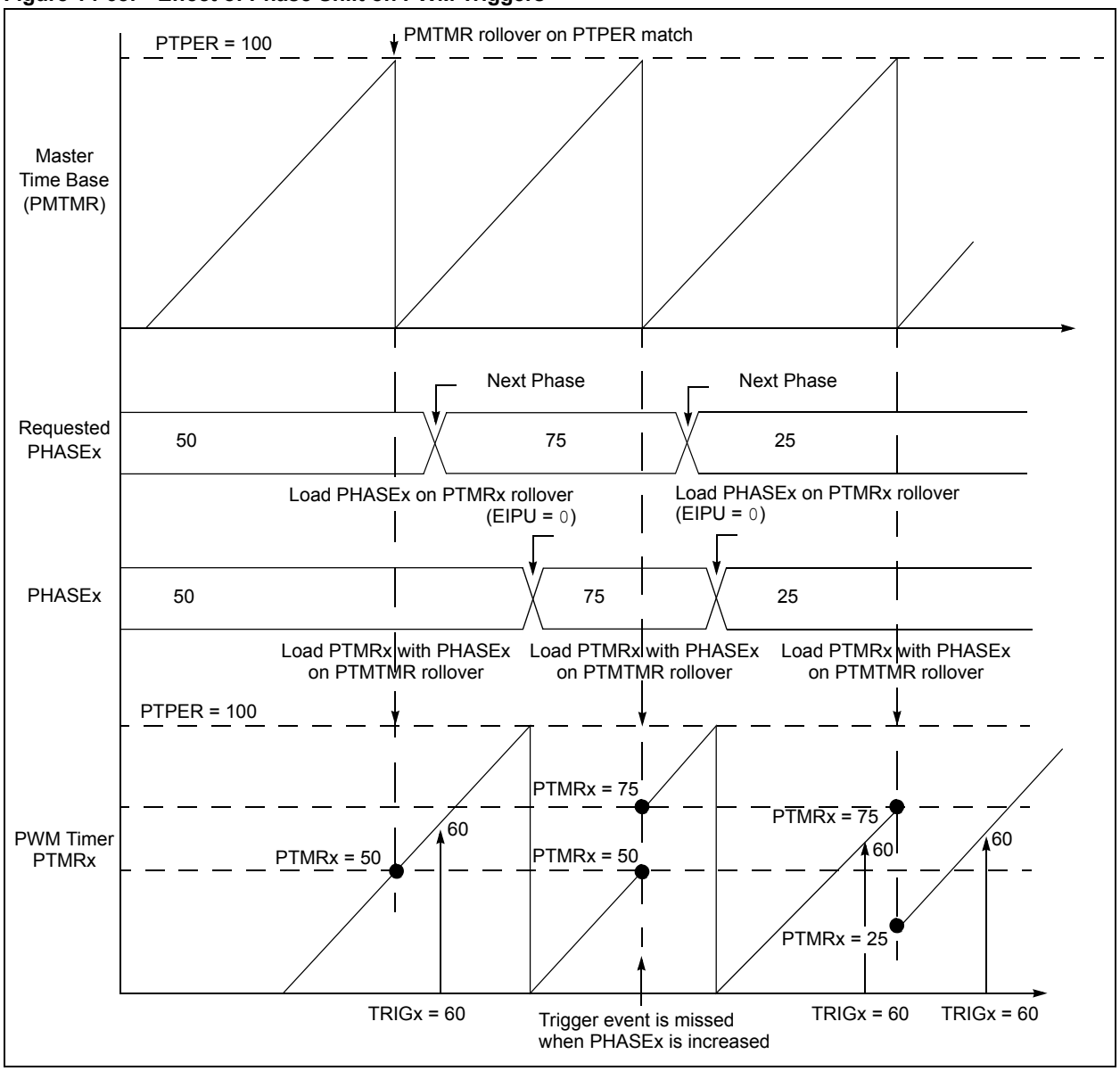
Example 14-46: Independent PWM ADC Triggering

```
/* Independent PWM ADC Triggering */

TRIG1 = 1248;              /* Point at which the ADC module is to be
                              triggered by primary PWM */
TRGCON1bits.TRGDIV = 0;    /* Trigger output divider set to trigger ADC on
                              every trigger match event */
TRGCON1bits.TRGSTRT = 4;   /* First ADC trigger event occurs after four
                              trigger match events */
PWMCON1bits.TRGIEN = 1;    /* Trigger event generates an interrupt request */
while (PWMCON1bits.TRGSTAT = = 1); /* Wait for ADC interrupt status change */
```

Note 1: The TRGSTAT bit is only cleared by clearing the TRGIEN bit. It is not cleared automatically.

2: Dynamic triggering can show some advantages where multiple PWM channels are used in applications, such as IPFC and multi-phase buck regulators. The TRIGx values can be changed based on the PWM period, duty, load current, etc. This is to ensure that the trigger points are separated from the rise and fall instances of the PWM channel.

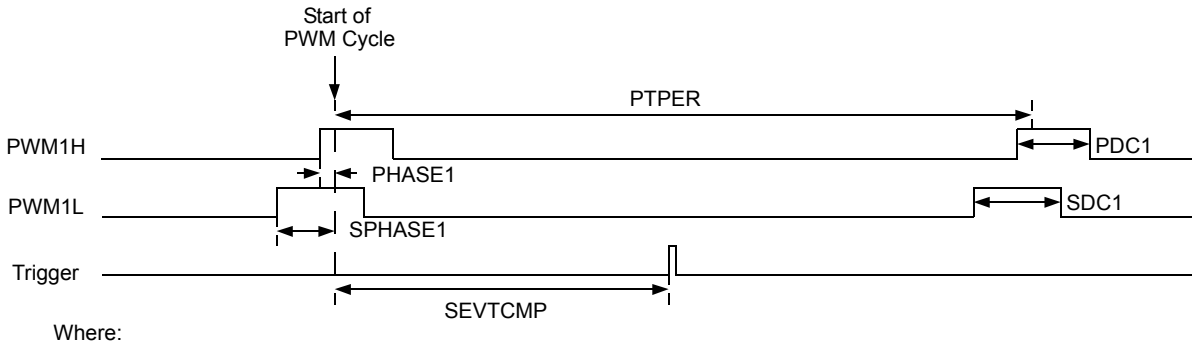
Figure 14-68: Effect of Phase Shift on PWM Triggers



When phase shifting the PWM signal, the PWM timer value is updated to reflect the new phase value. There is a possibility of missing trigger events when changing the phase from a smaller value to a larger value. The user-assigned application must ensure that this does not affect any control loop execution.

Timing diagrams that show triggers based on multiple PWM time base configurations are provided in Figure 14-63 through Figure 14-67.

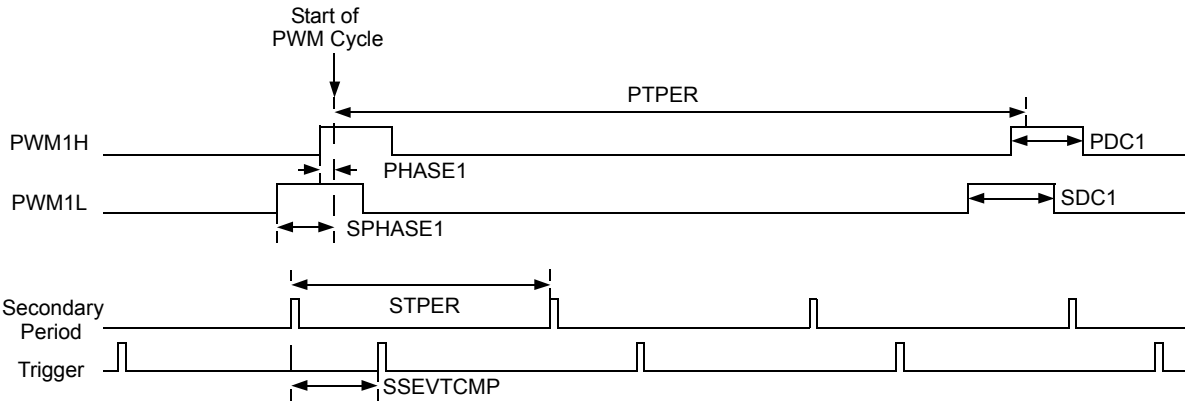
Figure 14-69: Independent PWM Mode. Independent Duty Cycle and Phase, Fixed Primary Period, Edge-Aligned, Triggering Primary Time Base



Where:

PHASE1	Phase of PWM1H
SPHASE1	Phase of PWM1L
PDC1	Duty Cycle of PWM1H
SDC1	Duty Cycle of PWM1L
PTPER	Period of PWM1H, PWM1L and Trigger
SEVTCMP	Special Event Compare

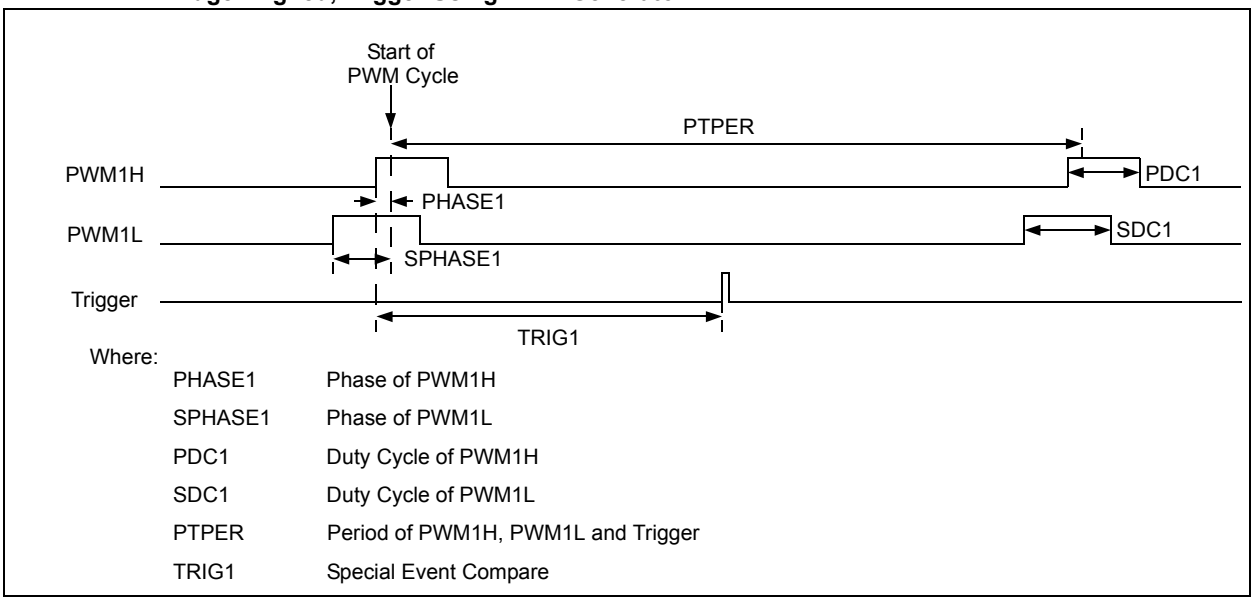
Figure 14-70: Independent PWM Mode. Independent Duty Cycle and Phase, Fixed Primary Period, Edge-Aligned, Trigger Using Secondary Time Base



Where:

PHASE1	Phase of PWM1H
SPHASE1	Phase of PWM1L
PDC1	Duty Cycle of PWM1H
SDC1	Duty Cycle of PWM1L
PTPER	Period of PWM1H and PWM1L
STPER	Period of Trigger
SSEVTCMP	Special Event Compare

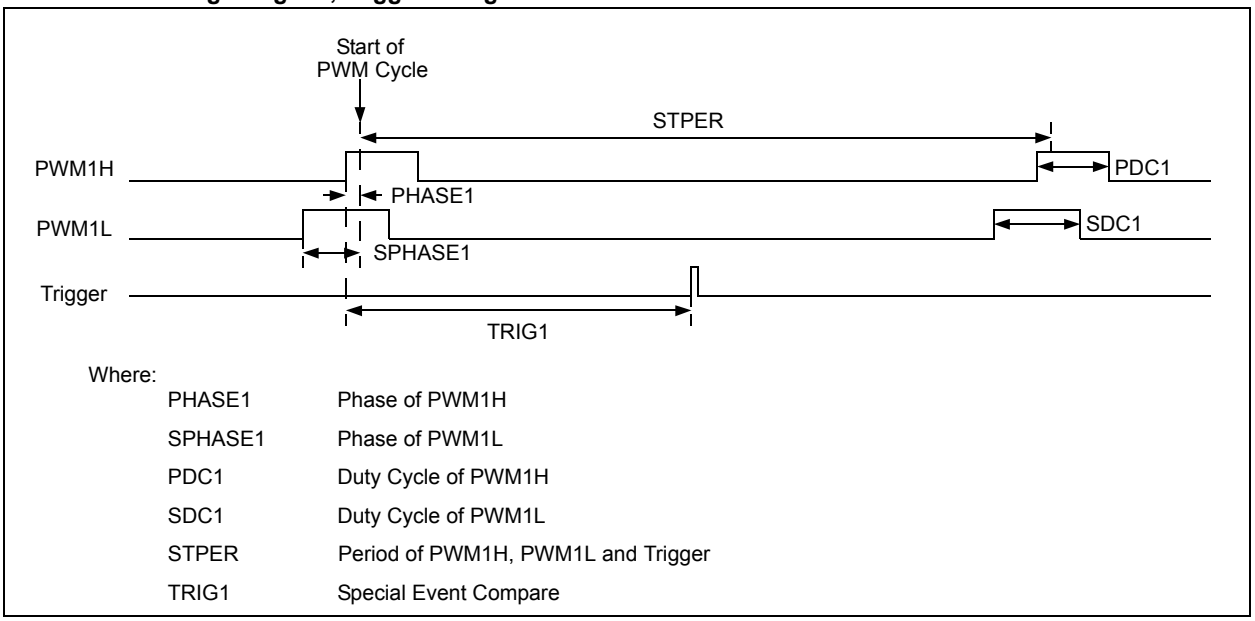
Figure 14-71: Independent PWM Mode - Independent Duty Cycle and Phase, Fixed Primary Period, Edge-Aligned, Trigger Using PWM Generator



Where:

PHASE1	Phase of PWM1H
SPHASE1	Phase of PWM1L
PDC1	Duty Cycle of PWM1H
SDC1	Duty Cycle of PWM1L
PTPER	Period of PWM1H, PWM1L and Trigger
TRIG1	Special Event Compare

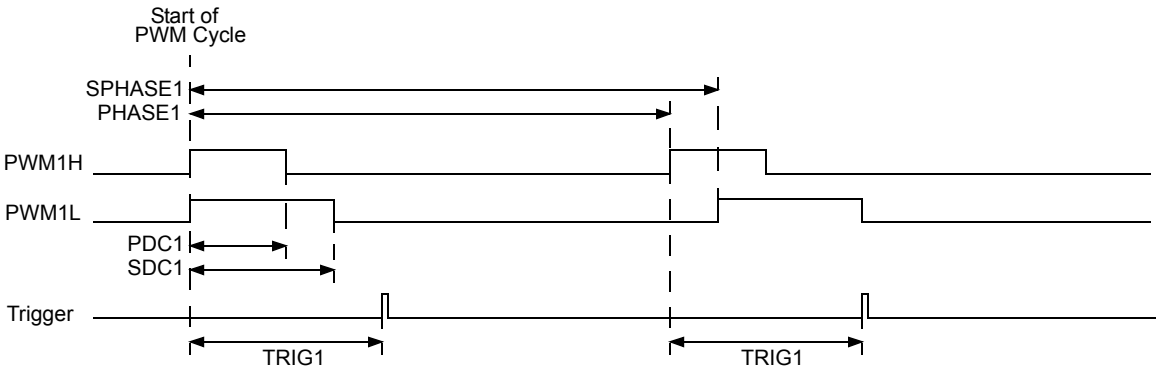
Figure 14-72: Independent PWM Mode - Independent Duty Cycle and Phase, Fixed Secondary Period, Edge-Aligned, Trigger Using PWM Generator



Where:

PHASE1	Phase of PWM1H
SPHASE1	Phase of PWM1L
PDC1	Duty Cycle of PWM1H
SDC1	Duty Cycle of PWM1L
STPER	Period of PWM1H, PWM1L and Trigger
TRIG1	Special Event Compare

Figure 14-73: Independent PWM Mode, Independent Duty Cycle and Period, No Phase Shifting, Edge-Aligned, Trigger Using PWM Generator



Where:

PHASE1	Period of PWM1H and Trigger
SPHASE1	Period of PWM1L
PDC1	Duty Cycle of PWM1H
SDC1	Duty Cycle of PWM1L
TRIG1	Special Event Compare

Table 14-6 provides a summary of the registers used as period and compare values for ADC triggers at different PWM configurations.

Table 14-6: ADC Triggers

ADxCON1(1)		PWMCONx		Function	
SSRCG	SSRC	ITB	MTBS	ADC Trigger Period	ADC Trigger Event
0	011	x	x	PTPER	SEVTCMP
0	101	x	x	STPER	SSEVTCMP
1	000	0	0	PTPER	TRIG1
1	000	0	1	STPER	TRIG1
1	000	1	x	PHASE1	TRIG1
1	001	0	0	PTPER	TRIG2
1	001	0	1	STPER	TRIG2
1	001	1	x	PHASE2	TRIG2
1	010	0	0	PTPER	TRIG3
1	010	0	1	STPER	TRIG3
1	010	1	x	PHASE3	TRIG3
1	011	0	0	PTPER	TRIG4
1	011	0	1	STPER	TRIG4
1	011	1	x	PHASE4	TRIG4
1	100	0	0	PTPER	TRIG5
1	100	0	1	STPER	TRIG5
1	100	1	x	PHASE5	TRIG5
1	101	0	0	PTPER	TRIG6
1	101	0	1	STPER	TRIG6
1	101	1	x	PHASE6	TRIG6
1	110	0	0	PTPER	TRIG7
1	110	0	1	STPER	TRIG7
1	110	1	x	PHASE7	TRIG7

Legend: x = don't care

Note 1: Refer to **Section 16. "Analog-to-Digital Converter (ADC)"** (DS70621) for more information about ADC Configuration registers.

14.9 PWM INTERRUPTS

The High-Speed PWM module can generate interrupts based on internal timing signals or 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 provides its own interrupt request signal to the interrupt controller. The interrupt for each PWM generator is a Boolean OR of the trigger event interrupt request, the current-limit input event, or the fault input event for that module.

Besides one interrupt request signal per PWM generator, the interrupt controller receives an interrupt request signal from the primary time base on special events.

The interrupt requests coming from each PWM generator are called Individual PWM Interrupts. The Interrupt Request (IRQ) for each one of these individual interrupts can come from the PWM individual trigger, PWM fault logic, or PWM current-limit logic. Each PWM generator has the PWM interrupt flag in an IFSx register. When an interrupt request is generated by any of the above sources, the PWM interrupt flag associated with the selected PWM generator is set.

If more than one IRQ source is enabled, the interrupt source is determined using the user-assigned application by checking the TRGSTAT, FLTSTAT and CLSTAT bits in the PWMCONx register.

14.9.1 PWM Time Base Interrupts

In each PWM generator, the High-Speed PWM module can generate interrupts based on the master time base and/or the individual time base. The PWM Special Event Compare register (SEVTCMP) specifies timer based interrupts for the primary time base, and the TRIGx registers specify timer based interrupts for the individual time bases.

The primary time base special event interrupt is enabled via the SEIEN bit (PTCON<11>). In each PWM generator, the individual time base interrupts generated by the trigger logic are controlled by the TRGIEN bit (PWMCON<10>).

Note:	When an appropriate match condition occurs, the Special Event Trigger signal and the individual PWM trigger pulses to the ADC are always generated regardless of the setting of their respective interrupt enable bits.

14.10 PWM FAULT PINS

The key functions of the PWM fault input pins are as follows:

- Each PWM generator can select its own fault input source from a selection of up to eight fault and current-limit pins
- Each PWM generator has FLTSRC<4:0> bits (FCLCONx<7:3>). These bits specify the source for its fault input signal.
- Each PWM generator has the FLTIEN bit (PWMCONx<12>). This bit enables the generation of fault interrupt requests.
- Each PWM generator has the FLTPOL bit (FCLCONx<2>). This bit selects the active state of the selected fault input.
- Upon occurrence of a Fault condition, the PWMxH and PWMxL outputs can be forced to one of the following states:
  - If the IFLTMOD bit is enabled, the FLTDAT<1:0> (High/Low) bits (IOCONx<5:4>) provides data values to be assigned to the PWMxH and PWMxL outputs
  - In Fault mode, the FLTDAT<1:0> (High/Low) bits provide the data values to be assigned to the PWMxH and PWMxL outputs

The following list describes major functions of the fault input pin:

- A fault can override the PWM outputs. The FLTDAT<1:0> bits (IOCONx<5:4>) can have a value of either '0' or '1'. If FLTDAT is set to '0', it is processed asynchronously to enable the immediate shutdown of the associated power transistors in the application circuit. If FLTDAT is set to '1', it is processed by the dead time logic and then applied to the PWM outputs.
- The fault signals can generate interrupts. The FLTIEN bit (PWMCONx<12>) controls the fault interrupt signal generation. The user-assigned application can specify interrupt signal generation even if the FLTMOD bits disable the fault override function. This allows the fault input signal to be used as a general purpose external interrupt request signal.
- The fault input signal that can be used as a trigger signal to the ADC, which initiates an ADC conversion process. The ADC trigger signals are always active regardless of the state of the High-Speed PWM module, the FLTMOD bits, or the FLTIEN bit.

The FLTx pins are normally active-high. The FLTPOL bit (FCLCONx<2>), when set to '1', inverts the selected fault input signal; therefore, these pins are set as active-low.

The fault pins are also readable through the port I/O logic when the High-Speed PWM module is enabled. This allows the user-assigned application to poll the state of the fault pins in software.

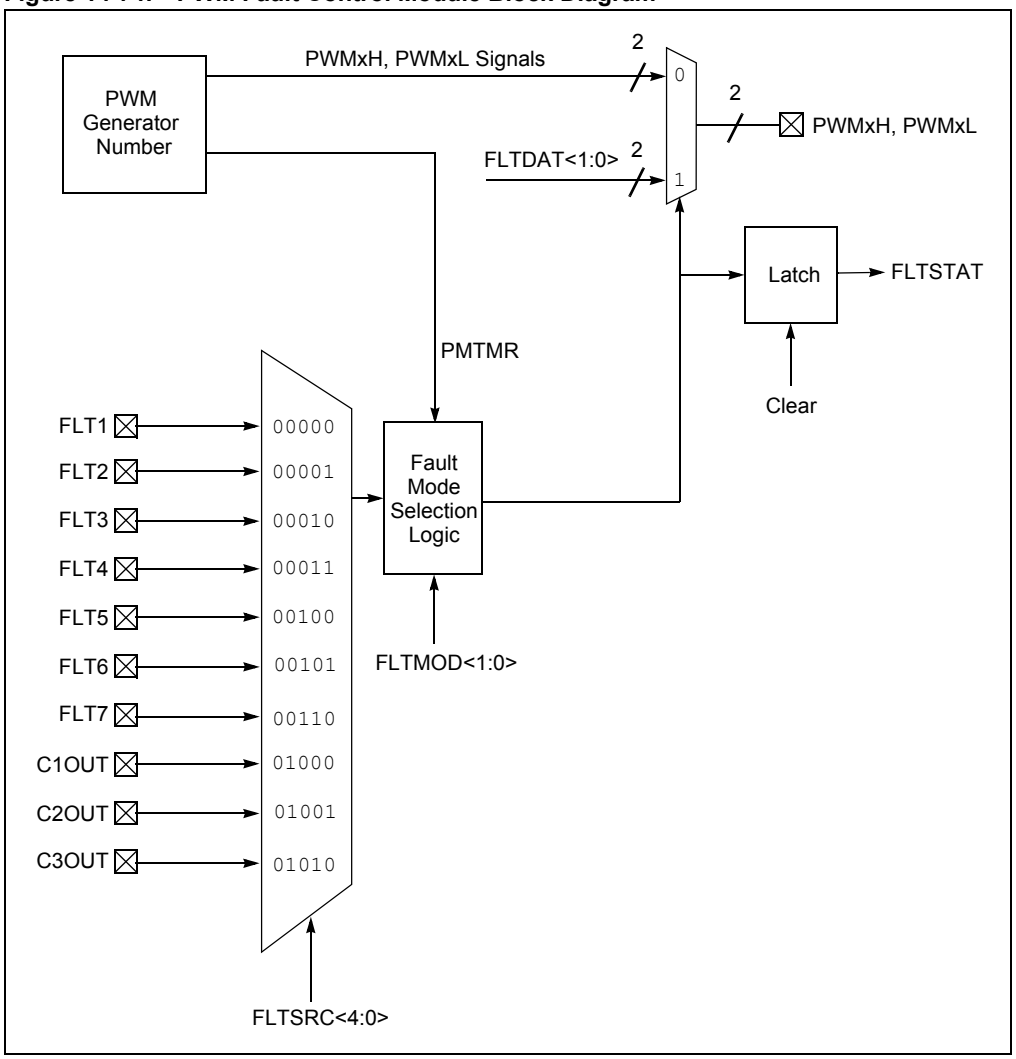
14.10.1 Class B Fault

Certain devices incorporate a fault that has been implemented with Class B safety features, which is known as the Class B Fault. This fault operates in a similar manner as other faults, with the exception that on any type of reset, the PWM module maintains ownership of that pin. See the specific device data sheet for the fault that incorporates this feature.

At reset, this fault is enabled in Latched mode to guarantee the fail-safe power-up of the application. The application software must clear this fault before the PWM module can be enabled. To clear the Fault condition, this fault pin must first be pulled high externally, or the internal pull-up resistor in the CNPUx register can be enabled. Once the Fault condition is cleared, the PWM module can be enabled, and if desired, the fault can be disabled.

Note:	If present on the device, the Class B Fault is enabled on any type of reset. The user application must clear this fault before the PWM module can be enabled.

Figure 14-74: PWM Fault Control Module Block Diagram



14.10.2 PWM Fault Generated by the Analog Comparator

The dsPIC33EPXXXMU806/810/814 and PIC24EPXXXGU810/814 devices support virtual (internal) connections to the output of the comparator modules C1OUT, C2OUT, and C3OUT (see Figure 26-1 in **Section 26.0 "Comparator Module"** (DS70357).

Virtual connections provide a simple way of inter-peripheral connection without utilizing a physical pin. For example, by setting the FLT1R<6:0> bits of the RPINR12 register to the value of 'b0000001, the output of the Analog Comparator, C1OUT, will be connected to the PWM Fault 1 input, which allows the Analog Comparator to trigger PWM faults without the use of an actual physical pin on the device.

Example 14-47 shows the configuration of the Analog Comparator 1 as one of the fault sources to the PWM that is connected to the fault input pin 1.

Example 14-47: Configuring Analog Comparator 1 as a Fault Source to the HS PWM Module

```
__builtin_write_OSCCONL(OSCCON & (~(1<<6))); // Unlock Register
RPINR12bits.FLT1R = 1;// Assign FLT1 to Pin C1OUT
__builtin_write_OSCCONL(OSCCON | (1<<6));// Lock Registers
```

14.10.3 Fault Interrupts

The FLTIENx bit (PWMCONx<12>) determine whether an interrupt will be generated when the FLTx input is asserted high. The FLTDAT<1:0> (High/Low) bits supply the data values to be assigned to the PWMxH and PWMxL pins in case of a fault.

The PWM Fault states are available on the FLTSTAT bit (PWMCONx<15>). The FLTSTAT bit displays the fault IRQ latch. 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 can be used as general purpose I/O or interrupt input pins.

In addition to its operation as the PWM logic, the fault pin logic can also operate as an external interrupt pin. If the faults are not allowed to affect the PWM generators in the FCLCONx register, the fault pin can be used as a general purpose interrupt pin.

14.10.3.1 FAULT INPUT PIN MODES

The fault input pin has two modes of operation:

- **Latched mode:** In Latched mode, the PWM outputs follow the states defined in the FLTDAT bits in the IOCONx registers when the fault pin is asserted. 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 (PWMCONx<15>) is cleared before the Fault condition ends, the High-Speed PWM module waits until the fault pin is no longer asserted. Software can clear the FLTSTAT bit by writing '0' to the FLTIEN bit in the PWMCONx register.
- **Cycle-by-Cycle mode:** In Cycle-by-Cycle mode, the PWM outputs remain in the deasserted PWM state as long as the fault input pin remains asserted. In 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 by using the FLTMOD bits (FCLCONx<1:0>).

14.10.4 Fault Entry

With respect to the device clock signals, the PWM pins always provide asynchronous response to the fault input pins. Therefore, if '0' is deasserted, the FLTDAT bit will immediately deassert the associated PWM output, and if the specified FLTDAT bits are asserted (set to '1'), the FLTDAT bits are processed by the dead time logic prior to being output as a PWM signal.

Refer to **14.12.4 "Fault/Current-Limit Override and Dead Time Logic"**, for more information on data sensitivity and behavior in response to current-limit or fault events.

14.10.5 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, if 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.
   b) Disable the PWM fault interrupt by clearing the FLTIEN bit (PWMCONx <12>).
   c) Enable the PWM fault interrupt by setting FLTMOD<1:0> (FCLCONx <1:0>) = 0b00.
3. Disable PWM faults by setting FLTMOD<1:0> (FCLCONx <1:0>) = 0b11.
4. Enable the latched PWM Fault mode by setting FLTMOD<1:0> (FCLCONx <1:0>) = 0b00.

14.10.6 Fault Exit with PMTMR 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 (the time base is disabled, so 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 FSTAT bit has been cleared in software.

14.10.7 Fault Pin Software Control

The fault pin can be controlled manually in software. Since the fault input is shared with a GPIO port pin, this pin can be configured as an output by clearing the corresponding TRIS bit. When the port bit for the pin is set, the fault input will be activated.

14.10.8 PWM Current-Limit Pins

The key functions of the PWM current-limit pins are as follows:

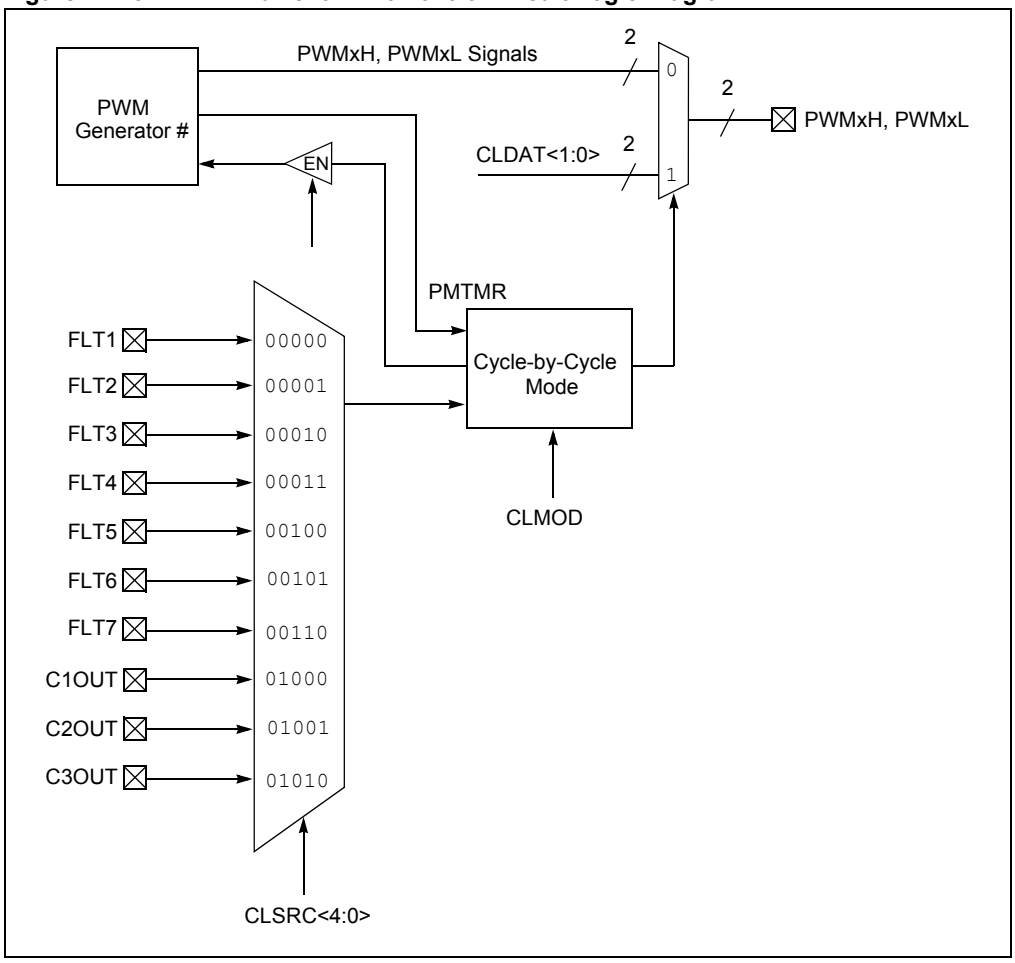
- Each PWM generator can select its own current-limit input source up to eight fault and current-limit pins. To configure the analog comparator as one of the current-limit sources, refer to **14.10.2 “PWM Fault Generated by the Analog Comparator”**.
- Each PWM generator has control bits, CLSRC<4:0> (FCLCONx<14:10>). These bits specify the source for its fault input signal.
- Each PWM generator has a corresponding CLIEN bit (PWMCONx<11>). This bit enables the generation of current-limit interrupt requests.
- Each PWM generator has an associated CLPOL bit (FCLCONx<9>).
- Upon occurrence of current-limit condition, outputs of the PWMxH and PWMxL generator change to one of the following states:
  - In Independent Fault mode of the IFLTMOD bit, the CLDAT<1:0> bits are not used for override functions
  - In the Current-Limit mode (CLMOD), the current-limit function is enabled. The CLDAT<1:0> (High/Low) bits supply the data values to be assigned to the PWMxH and PWMxL outputs.

The major functions of the current-limit pin are as follows:

- A current-limit can override the PWM outputs. The CLDAT<1:0> bits (IOCONx<3:2>) can have a value of either '0' or '1'. If CLDAT is set to '0', it is processed asynchronously to enable immediate shutdown of the associated power transistors in the application circuit. If CLDAT is set to '1', it is processed by the dead time logic and then applied to the PWM outputs.
- The current-limit signals can generate interrupts. The CLIEN bit (PWMCONx<11>) controls the current-limit interrupt signal generation. The user-assigned application can specify interrupt generation even if the CLMOD bit (FCLCONx<8>) disables the current-limit override function. This allows the current-limit input signal to be used as a general purpose external interrupt request signal.
- The current-limit input signal that can be used as a trigger signal to the ADC, which initiates an ADC conversion process. The ADC trigger signals are always active regardless of the state of the High-Speed PWM module, the FLTMOD bits, or the FLTIEN bit.
- A current-limit signal resets the time base for the affected PWM generator when the following occurs:
  - The CLMOD bit (FCLCONx<8>) for the PWM generator is '0'
  - The XPRES bit (PWMCONx<1>) is '1'
  - The PWM generator is in the Independent Time Base mode (ITB = 1)

    This behavior is called Current Reset mode, which is used in some Power Factor Correction (PFC) applications.

Figure 14-75: PWM Current-Limit Control Circuit Logic Diagram



14.10.9 Current-Limit Interrupts

The state of the PWM current-limit conditions is available on the CLSTAT bit (PWMCONx<14>). 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 can be used as general purpose I/O or interrupt input pins.

The current-limit pins are normally active-high. If set to '1', the CLPOL bit (FCLCONx<9>) invert the selected current-limit input signal and drives the signal into active-low state.

The interrupts generated by the selected current-limit signals are combined to create a single interrupt request signal. This signal is sent 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 High-Speed PWM module is enabled. This capability allows the user-assigned application to poll the state of the fault pins in software.

14.10.10 Simultaneous PWM Faults and Current-Limits

The current-limit override function, if enabled and active, forces the PWMxH and PWMxL pins to read the values specified by the CLDAT<1:0> bits (IOCONx<3:2>), unless the fault function is enabled and active. If the selected fault input is active, the PWMxH and PWMxL outputs read the values specified by the FLTDAT<1:0> bits (IOCONx<5:4>).

14.10.11 PWM Fault and Current-Limit Trigger Outputs to ADC

The current-limit and fault source selection bits, CLSRC<4:0>(FCLCONx<14:10>) and FLTSRC<4:0> (FCLCONx<7:3>), control the fault selection to each PWM generator module. The control multiplexers select the desired fault and current-limit signals for their respective modules.

Example 14-48: PWM Fault, Current-Limit and Leading-Edge Blanking Configuration

```
/* PWM Fault, Current-Limit and Leading-Edge Blanking Configuration */

FCLCON1bits.IFLTMOD = 0;    /* CLDAT bits control PWMxH and FLTDAT bits control PWMxL */
FCLCON1bits.CLSRC = 8;      /* Current-limit input source is Analog Comparator 1 */
FCLCON1bits.FLTSRC = 9;     /* Fault input source is Analog Comparator 2 */

FCLCON1bits.CLPOL = 1;      /* Current-limit source is active-low */
FCLCON1bits.FLTPOL = 1;     /* Fault Input source is active-low */

FCLCON1bits.CLMOD = 1;      /* Enable current-limit function */
FCLCON1bits.FLTMOD = 1;     /* Enable Cycle-by-Cycle Fault mode */

IOCON1bits.FLTDAT = 0;      /* PWMxH and PWMxL are driven inactive on occurrence of fault */

IOCON1bits.CLDAT = 0;       /* PWMxH and PWMxL are driven inactive on occurrence of current-limit */

LEBCON1bits.PHR = 1;        /* Rising edge of PWMxH will trigger LEB counter */
LEBCON1bits.PHF = 0;        /* Falling edge of PWMxH is ignored by LEB counter */
LEBCON1bits.PLR = 1;        /* Rising edge of PWMxL will trigger LEB counter */
LEBCON1bits.PLF = 0;        /* Falling edge of PWMxL is ignored by LEB counter */
LEBCON1bits.FLTLEBEN = 1;   /* Enable fault LEB for selected source */
LEBCON1bits.CLLEBEN = 1;    /* Enable current-limit LEB for selected source */

PWMCON1bits.XPRES = 0;      /* External pins do not affect PWM time base reset */
PWMCON1bits.FLTIEN = 1;     /* Enable fault interrupt */
PWMCON1bits.CLIEN = 1;      /* Enable current-limit interrupt */

while (PWMCON1bits.FLTSTAT == 1);     /* Wait when fault interrupt is pending */
while (PWMCON1bits.CLSTAT == 1);      /* Wait when current-limit interrupt is pending */
```

14.11 SPECIAL FEATURES

The following special features are available in the High-Speed PWM module:

- Leading-Edge Blanking (LEB)
- State Blanking
- Chop mode
- Individual time base capture
- PWM pin swapping
- PWM output pin control and override
- PWM immediate update

14.11.1 Leading-Edge Blanking (LEB)

Each PWM generator supports LEB of the current-limit and fault inputs through the LEB bits (LEBCONx<9:3>) and the PHR (LEBCONx<15>), PHF (LEBCONx<14>), PLR (LEBCONx<13>), PLF (LEBCONx<12>), FLTLEBEN (LEBCONx<11>), and CLLEBEN bits (LEBCONx<10>) in the Leading-Edge Blanking Control 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 are edge-sensitive. These bits support the blanking (ignoring) of the current-limit and fault inputs for a period of 0 ns to 1023 ns in 8.33 ns increments (when TOSC = 120 MHz) following any specified rising or falling edge of the PWMxH and PWMxL signals.

Equation 14-8: Leading-Edge Blanking Calculation

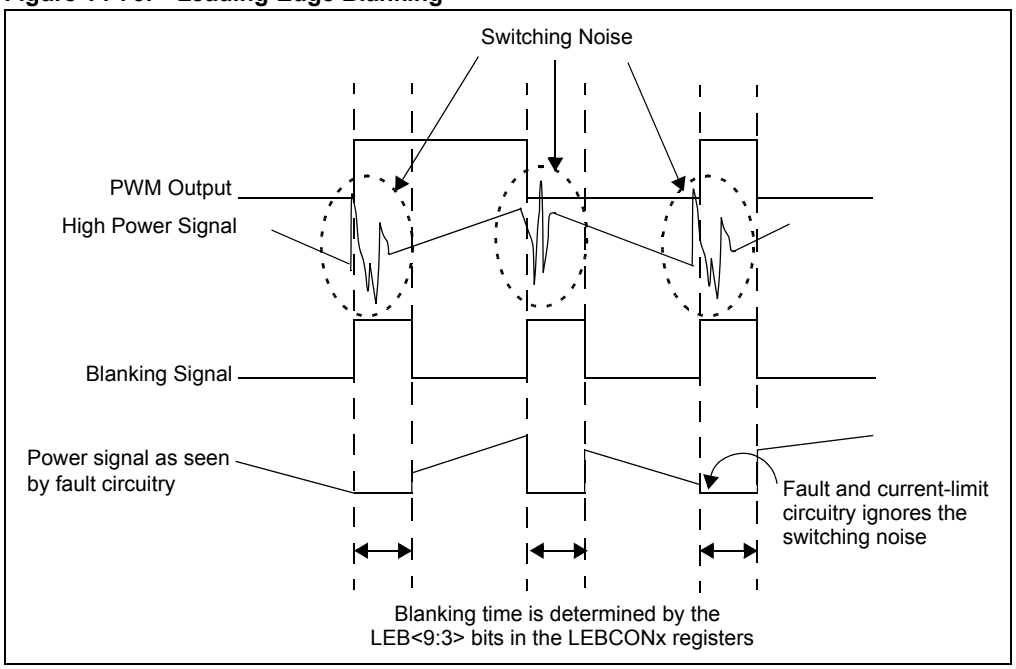
$$\textit{LEB Duration at Maximum Clock Rate} = (\textit{LEBDLYx<11:0>}) * 8.33\ \textit{ns}$$

In high-speed switching applications, switches (such as MOSFETs and IGBTs) typically generate very large transients. These transients can cause measurement errors. The LEB function enables the user-assigned application to ignore the expected transients caused by the MOSFETs/IGBTs switching that occurs near the edges of the PWM output signals.

The PHR, PHF, PLR and PLF bits select the edge type of the PWMxH and PWMxL signals, which starts 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 reinitializes and continues counting.

The FLTLEBEN and CLLEBEN bits enable the application of the blanking period to the selected fault and current-limit inputs. Figure 14-76 illustrates how an application ignores the fault signal in the specified blanking period.

Figure 14-76: Leading-Edge Blanking



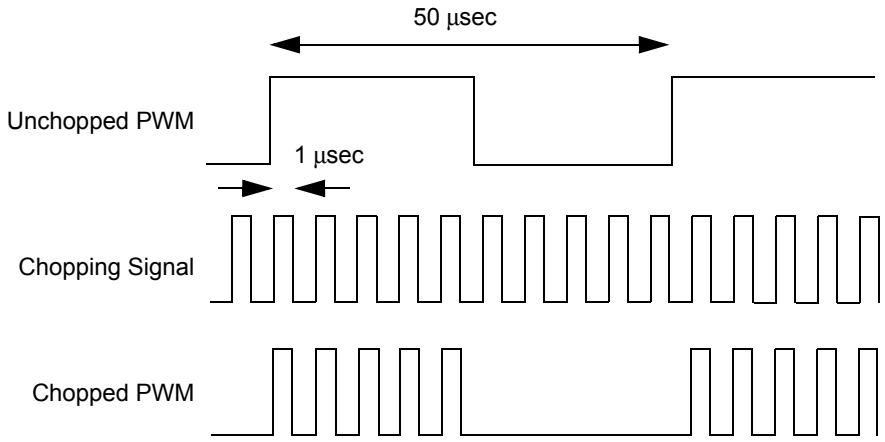
14.11.2 Chop Mode

Many power control applications use transistor configurations that require an isolated transistor gate drive. An example is a three-phase “H-bridge” configuration, where the upper transistors are at an elevated electrical potential.

One method to achieve an isolated gate drive circuit is to use pulse transformers to couple the PWM signals across a galvanic isolation barrier to the transistors. Unfortunately, in applications that use either long duty cycle ratios, or slow PWM frequencies, the transformer’s low-frequency response is poor. The pulse transformer cannot pass a long duration PWM signal to the isolated transistor(s). If the PWM signals are “chopped” or gated by a high-frequency clock signal, the high-frequency alternating signal easily passes through the pulse transformer. The chopping frequency is typically hundreds or thousands of times higher in frequency as compared to the PWM frequency. The higher the chopping (carrier) frequency relative to the PWM frequency, the more the PWM duty cycle resolution is preserved.

Figure 14-77 shows an example waveform of high-speed PWM chopping. In this example, a 20 kHz PWM signal is chopped with a 500 kHz carrier generated by the chop clock.

Figure 14-77: High-Frequency PWM Chopping



Note: Not drawn to scale.

The chopping function performs a logical AND operation of the PWM outputs. Because of the finite period of the chopping clock, the resultant PWM duty cycle resolution is limited to one half of the chop clock period.

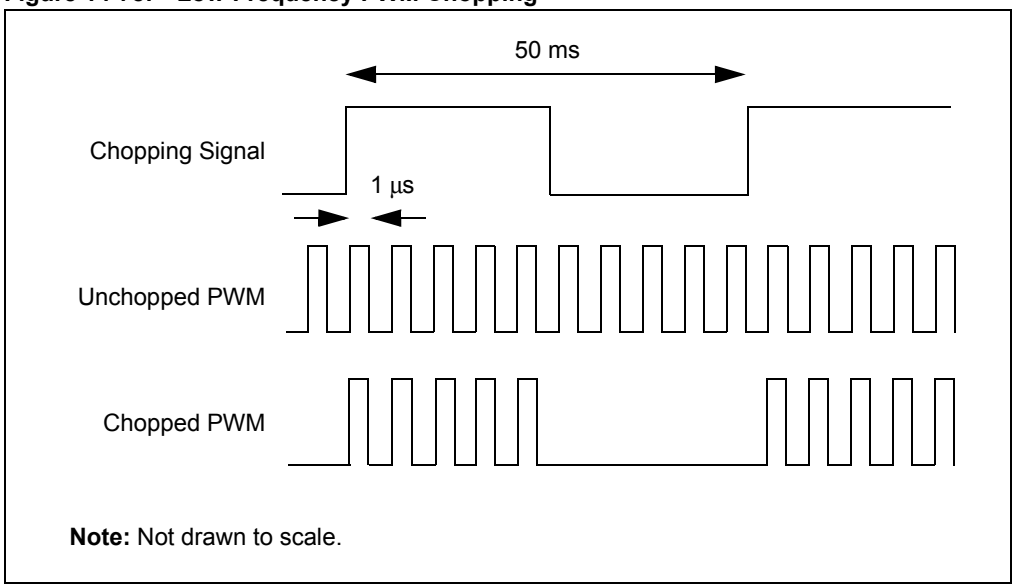
The CHOP register enables the user to specify a chopping clock frequency. The CHOP value specifies a PWM clock divide ratio. The chop clock divider operates at the PWM clock frequency specified by the PWM Clock Divider Select bits, PCLKDIV<2:0> (PTCON2<2:0>). The CHPCLKEN bit (CHOP<15>) enables the chop clock generator.

The CHOPHEN and CHOPLEN bits in the AUXCONx register enable the chop clock to be applied to the PWM outputs. The CHOPSEL<3:0> bits (AUXCONx<5:2>), select the desired source for the chop clock. The default selection is the chop clock generator controlled by the CHOP register. The CHOPSEL<3:0> bits enable the user to select other PWM generators as a chop clock source.

If the CHOPHEN or CHOPLEN bits are set in the AUXCONx register, the chopping function is applied to the PWM output signals after the current-limit and fault functions are applied to the PWM signal. The CHPCLK signal is available for output from the module for use as an output signal for the device.

Normally, the chopping clock frequency is higher than the PWM cycle frequency, but new applications can use chop clock frequencies that are much lower than the PWM cycle frequency. Figure 14-78 shows a low-frequency PWM chopping waveform. In this figure, another PWM generator operating at a lower frequency chops or “blanks” the PWM signal.

Figure 14-78: Low-Frequency PWM Chopping



Note: Not drawn to scale.

14.11.3 Individual Time Base Capture

Each PWM generator has a PWMCAPx register that automatically captures the independent time base counter value when the rising edge of the current-limit signal is detected. This feature is active only after the application of the LEB function. The user-assigned application should read the register before the next PWM cycle causes the capture register to be updated again.

The Capture register is used in current mode control applications that use the analog comparators or external circuitry to terminate the PWM duty cycle or period. By reading the independent time base value at the current threshold, the user-assigned application can calculate the slope of the current rise in the inductor. The secondary independent time base does not have an associated Capture register.

14.11.4 PWM Pin Swapping

The SWAP bit (IOCONx<1>), if set to '1', enables the user-assigned application to connect the PWMxH signal to the PWMxL pin and the PWMxL signal to the PWMxH pin. If the SWAP bit is set to '0', the PWM signals are connected to their respective pins.

To perform the swapping function on the PWM cycle boundaries, the OSYNC bit (IOCONx<0>) must be set. If the user-assigned application changes the state of the SWAP bit (IOCONx<1>) when the module is operating and the OSYNC bit is clear, the SWAP function will attempt to execute in the middle of a PWM cycle and the operation will yield unpredictable results.

The SWAP function should be executed prior to the application of dead time. Dead time processing is required since execution of switch function may enable the transistors in the user-assigned application that are previously in disable state, possibly causing current shoot-through.

The SWAP feature is useful for the applications that support multiple switching topologies with a single application circuit board. It also enables the user-assigned application to change the transistor modulation scheme in response to changing conditions.

The SWAP function can be implemented by using either of the following methods:

- **Dynamic Swapping:** In the dynamic swapping, the state of the SWAP bit can be changed dynamically based on the system response (e.g., Switch Mode Power Supply (SMPS) Power Control)
- **Static Swapping:** In static swapping, the SWAP bit is set during the start-up configuration and remains unchanged during the program execution or on-the-fly (e.g., Motor Control)

14.11.4.1 EXAMPLE 1: PIN SWAPPING WITH SMPS POWER CONTROL

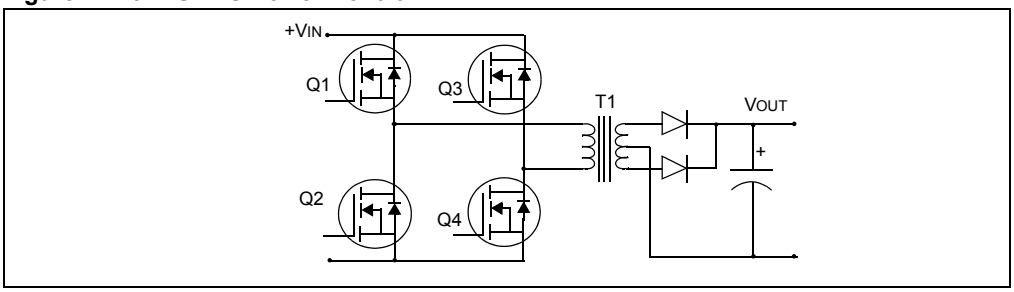
The SMPS Power Control example describes dynamic swapping. In power conversion/motor control application, the transistor modulation technique can be changed between the full-bridge Zero Voltage Transition (ZVT) and standard full-bridge “on-the-fly” transition to meet different load and efficiency requirements. As shown in Figure 14-79, the generic full-bridge converter can operate in Push-Pull mode. The transistors are configured as follows:

- Q1 = Q4
- Q2 = Q3

The generic full-bridge converter can also operate in a ZVT mode. The transistors are configured as follows:

- Q1 = PWM1H
- Q2 = PWM1L
- Q3 = PWM2H
- Q4 = PWM2L

Figure 14-79: SMPS Power Control



14.11.4.2 EXAMPLE 2: PIN SWAPPING WITH MOTOR CONTROL

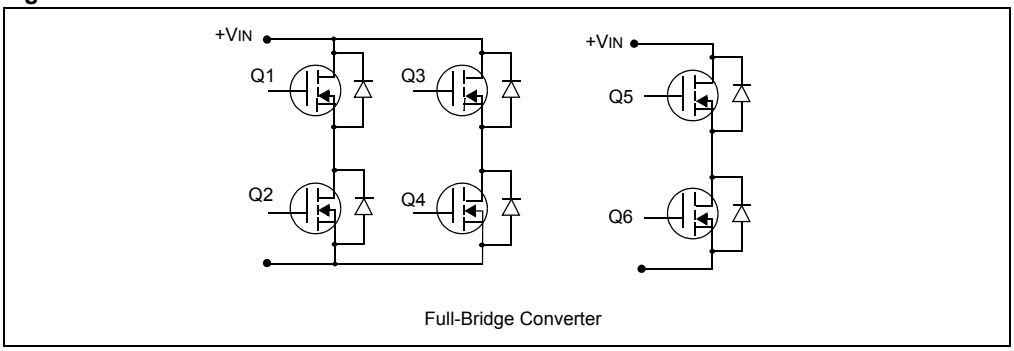
The Motor Control example describes static swapping. Consider a generic motor control system, which is capable of driving two different types of motors, such as DC motors and three-phase AC induction motors.

Brushed DC motors typically use a full-bridge transistor configuration, as shown in Figure 14-80. The Q1 and Q4 transistors are driven with similar waveforms, while the Q2 and Q3 transistors are driven with the complementary waveforms. This is also known as “driving the diagonals”. Note that the Q5 and Q6 transistors are not used in a brushed DC motor.

The transistors are configured as follows:

- Q1 = PWM1H
- Q2 = PWM1L
- Q3 = PWM2L
- Q4 = PWM2H

Figure 14-80: Motor Control



When compared to the DC motor, an AC induction motor uses all the transistors in the full-bridge configuration. However, the significant difference is that the transistors are now driven as three half-bridges where the upper transistors are driven by the PWMxH outputs and the lower transistors are driven by PWMxL outputs.

The transistors are configured as follows:

- Q1 = PWM1H
- Q2 = PWM1L
- Q3 = PWM2H (note the difference with DC motors)
- Q4 = PWM2L (note the difference with DC motors)
- Q5 = PWM3H
- Q6 = PWM3L

Example 14-49: PWM Pin Swapping

```
/* PWM Pin Swapping feature */

IOCON1bits.SWAP = 1;    /* PWMxH output signal is connected to the PWMxL pin
                           and vice versa */
```

14.12 PWM OUTPUT PIN CONTROL

If the High-Speed PWM module is enabled, the priority of PWMxH/PWMxL pin ownership from lowest to highest priority is as follows:

- PWM generator (lowest priority)
- Swap function
- PWM output override logic
- Current-limit override logic
- Fault override logic
- PTEN (GPIO/PWM) ownership (highest priority)

If the High-Speed PWM module is disabled, the GPIO module controls the PWMx pins.

Example 14-50: PWM Output Pin Assignment

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

Example 14-51: PWM Output Pins State Selection

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

Example 14-52: Enabling the High-Speed PWM Module

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

PTCONbits.PTEN = 1;
```

14.12.1 PWM Output Override Logic

The PWM output override feature is used to drive the individual PWM outputs to a desired state based on system requirements. The output can be driven to both the active state as well as the inactive state. The High-Speed PWM module override feature has the priority as assigned in the list above. All control bits associated with the PWM output override function are contained in the IOCONx register. If the bits, PENH (IOCONx<15>) and PENL (IOCONx<14>) are set, the High-Speed PWM module controls the PWMx output pins. The PWM Output Override bits allow the user-assigned 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 (IOCONx<7:6>), determine the state of the PWM I/O pins when a particular output is overridden by the bits, OVRENH (IOCONx<9>) and OVRENL (IOCONx<8>).

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.

14.12.2 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 bits (FLTDAT<1:0>) 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> bits (IOCONx<3:2>) 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.

14.12.3 Override Synchronization

If the OSYNC bit in the PWM I/O Control register (IOCONx<0>) 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.

14.12.4 Fault/Current-Limit Override and Dead Time Logic

In the event of a Fault and Current-Limit condition, the data in the FLTDAT<1:0> bits (IOCONx<5:4>) or CLDAT<1:0> bits (IOCONx<3:2>) determine the state of the PWM I/O pins.

If any of the FLTDAT<1:0> bits (IOCONx<5:4>) or CLDAT<1:0> bits (IOCONx<3:2>) are '0', the PWMxH and/or PWMxL outputs are driven inactive immediately, bypassing the dead time logic. This behavior turns off the PWM outputs immediately without any additional delays. This may aid many power conversion/motor control applications that require a fast response to fault shutdown signals to limit circuitry damage and control system accuracy.

If any of the FLTDAT<1:0> bits (IOCONx<5:4>) or CLDAT<1:0> bits (IOCONx<3:2>) are '1', the PWMxH and/or PWMxL outputs are driven active immediately passing through the dead time logic and, therefore, will be delayed by the specified dead time value. In this case, dead time will be inserted even if a Fault or Current-Limit condition occurs.

14.12.5 Asserting Outputs Via Current-Limit

In response to a current-limit event, the CLDAT bits (IOCONx<3:2>) can be used to assert the PWMxH and PWMxL outputs. 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.

Note:	In Complementary PWM Output mode, the dead time generator remains active under override condition. 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.

14.12.6 PENx (GPIO/PWM) Ownership

Most of the PWM output pins are normally multiplexed 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 RE0 and RE1, the configuration of GPIO pins will decide the PWM output status when halted by the Debugger.

Example 14-53: Code Example

```
/* PWM output will be pulled to low when the device is halted by the
    debugger */
TRISE = 0x0000; RE0 and RE1 configured for an output
LATE = 0x0000; RE0 and RE1 configured as Low output

/* PWM output will be pulled to high when the device is halted by the
    debugger */
TRISE = 0x0000; RE0 and RE1 configured for an output
LATE = 0x0003; RE0 and RE1 configured as High output

/* PWM output will be in tri-state when the device is halted by the
    debugger */
TRISE = 0x0003; RE0 and RE1 configured for an input
```

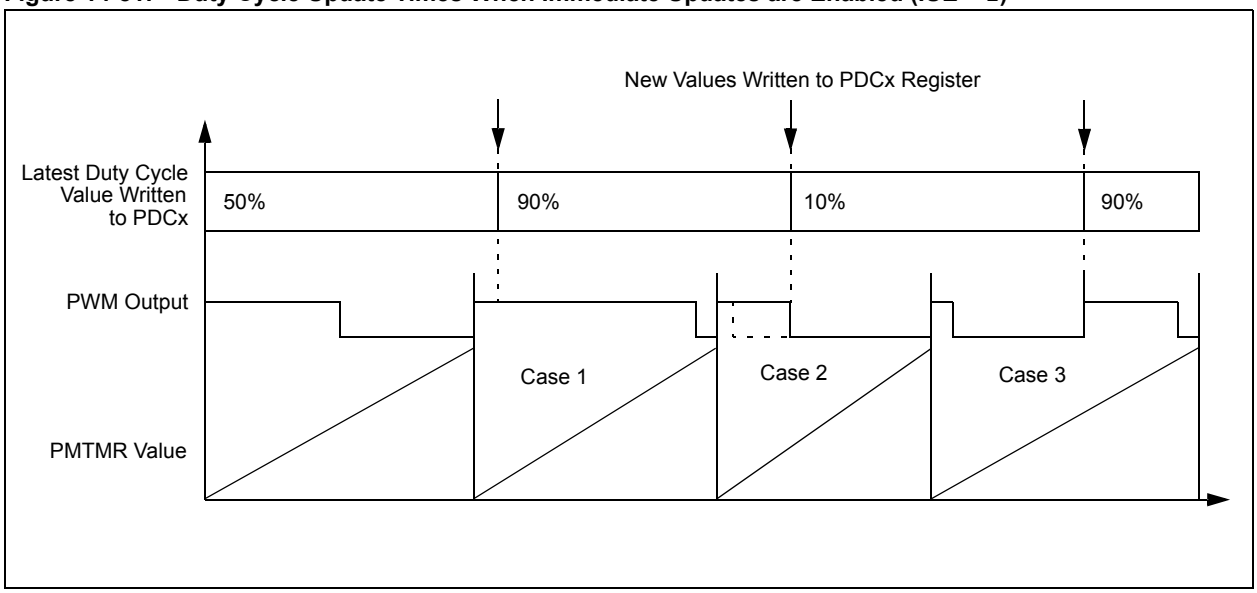
14.13 IMMEDIATE UPDATE OF PWM DUTY CYCLE

The high performance PWM control-loop application requires a maximum duty cycle update rate. Setting the Immediate Update Enable bit (IUE) in the PWM Control registers (PWMCONx<0>) enables this feature. In a closed-loop control application, any delay between the sensing of a system 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.

The IUE bit enables the user-assigned application to update the duty cycle values immediately after writing to the duty cycle registers, rather than waiting until the end of the time base period. If the IUE bit is set, an immediate update of the duty cycle is enabled. If the bit is cleared, immediate update of the duty cycle is disabled. The following three cases are possible when immediate update is enabled:

- **Case 1:** If the PWM output is active at the time the new duty cycle is written and the new duty cycle is greater than the current time base value, the PWM pulse-width is lengthened
- **Case 2:** If the PWM output is active at the time the new duty cycle value is written and the new duty cycle is less than the current time base value, the PWM pulse-width is shortened
- **Case 3:** If the PWM output is inactive when the new duty cycle value is written and the new duty cycle is greater than the current time base value, the PWM output becomes active immediately and remains active for the newly written duty cycle value

Figure 14-81: Duty Cycle Update Times When Immediate Updates are Enabled (IUE = 1)



Example 14-54: Immediate Update Selection

```
/* Enable Immediate Update of PWM */

PTCONbits.EIPU = 1;    /* Update active period register immediately
PWMCON1bits.IUE = 1;   /* Update active duty cycle, phase offset and
                          independent time period immediately */
```

14.14 POWER-SAVING MODES

This section discusses the operation of the High-Speed PWM module in Sleep mode and Idle mode.

14.14.1 High-Speed PWM Operation in Sleep Mode

When the device enters Sleep mode, the system clock is disabled. Since the clock for the PWM time base is derived from the system clock source (TCY), that clock will also be disabled and all enabled PWM output pins that were in effect prior to entering Sleep mode will be frozen in the output states. If the High-Speed PWM module is used to control a load in a power application, the High-Speed PWM module outputs must be placed into a safe state before executing the PWRSAV instruction. Depending on the application, the load may begin to consume excessive current when the PWM outputs are frozen in a particular output state. In such a case, the override functionality can be used to drive the PWM output pins into the inactive state.

If the fault inputs are configured for the High-Speed PWM module, the fault input pins continue to function normally when the device is in Sleep mode. If one of the fault pins is driven low while the device is in Sleep mode, the PWM outputs are driven to the programmed fault states. The fault input pins can also wake the CPU from Sleep mode. If the fault pin interrupt priority is greater than the current CPU priority, program execution starts at the fault pin interrupt vector location upon wake-up. Otherwise, execution continues from the next instruction following the PWRSAV instruction.

14.14.2 High-Speed PWM Operation in Idle Mode

The PWM module has a PTSIDL control bit in the PTCON register. This bit determines if the PWM module will continue to operate or stop when the device enters Idle mode. If PTSIDL = 0, the module will continue to operate as normal. If PTSIDL = 1, the module will shut down and stop its internal clocks. The system will not be able to access the special function registers in this mode. This will be the minimum power mode for the module. Stopped Idle mode functions like Sleep mode and fault pins will be asynchronously active. The control of the PWM pins will revert back to the GPIO bits associated with the PWM pins if the PWM module enters an Idle state.

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/motor control application, the action of putting the device into Idle mode will cause 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.

Note: For more details on power-saving modes, refer to **Section 9. "Watchdog Timer (WDT) and Power-Saving Modes"** (DS70615).

14.15 EXTERNAL CONTROL OF INDIVIDUAL TIME BASE(S)

External signals can reset the primary dedicated time bases, if the XPRES bit (PWMCONx<1>) is set. This mode of operation is called Current Reset PWM mode. If the user-assigned application sets the ITB bit, a PWM generator operates in Independent Time Base mode. If the user-assigned application sets the XPRES bit and operates the PWM generator in Master Time Base mode, the results may be unpredictable.

The current-limit source signal specified by the CLSRC<4:0> bits (FCLCONx<14:10>) causes the independent time base to reset. The active edge of the selected current-limit signal is specified by the CLPOL bit (FCLCONx<9>).

In Primary Independent Time Base mode, some Power Factor Correction (PFC) applications need to maintain the inductor current value above minimum desired current level. These applications use the external Reset feature. If the inductor current falls below the desired value, the PWM cycle is terminated early so that the PWM output can be asserted to increase the inductor current. The PWM period varies according to the application needs. This type of application is a variable frequency PWM mode.

14.16 APPLICATION INFORMATION

Typical applications that use different PWM operating modes and features are:

- Complementary Output Mode
- Push-Pull Output Mode
- Multi-Phase PWM
- Variable Phase PWM
- Current Reset PWM
- Constant Off-Time PWM
- Current-Limit PWM

Each application is described in the following sections.

14.16.1 Complementary Output Mode

The Complementary PWM mode shown in Figure 14-82 is generated in a manner that is similar to Standard Edge-Aligned mode. This mode provides a second PWM output signal on the PWMxL pin that is the complement of the primary PWM signal (PWMxH).

Figure 14-82: Complementary PWM Output Mode for SMPS

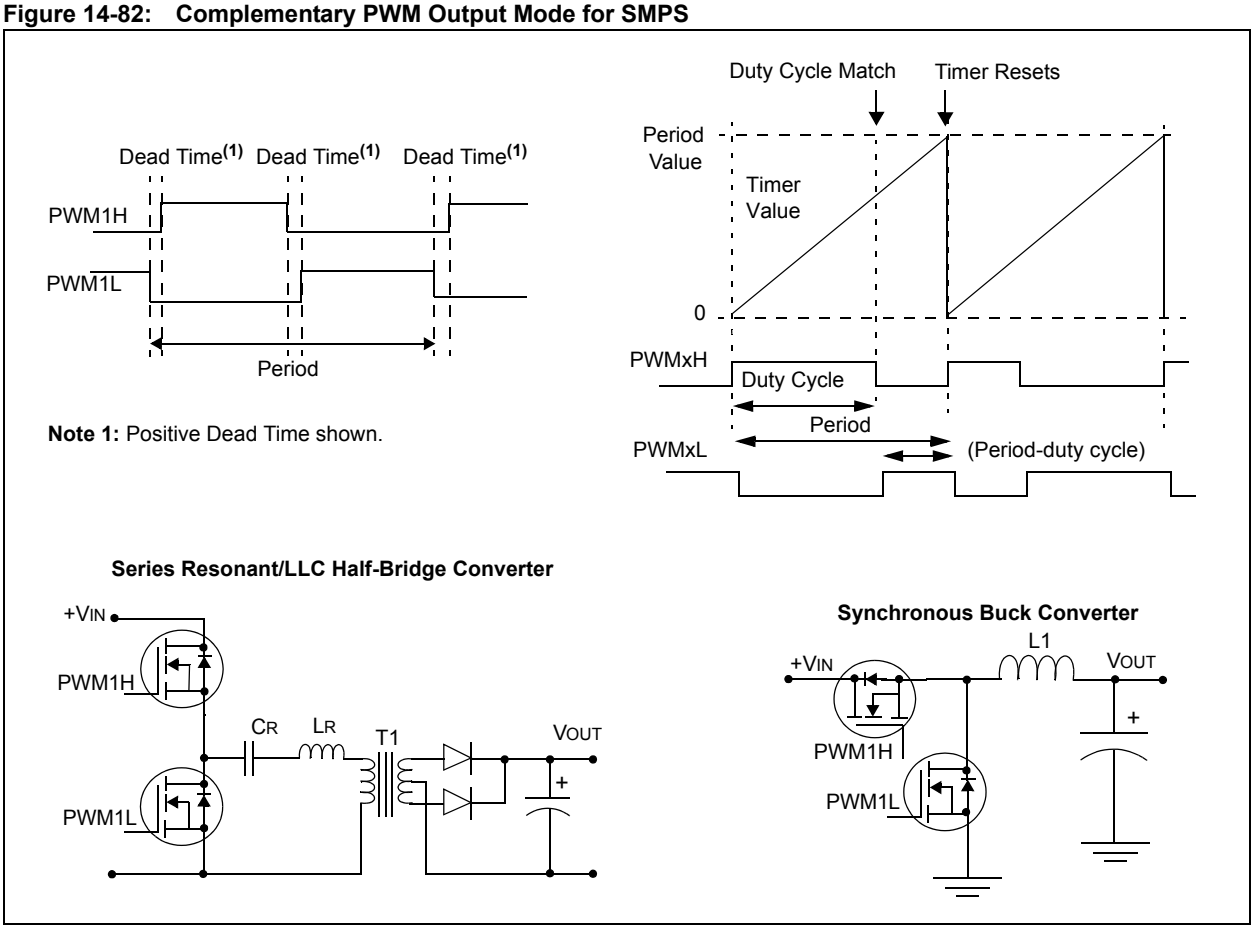
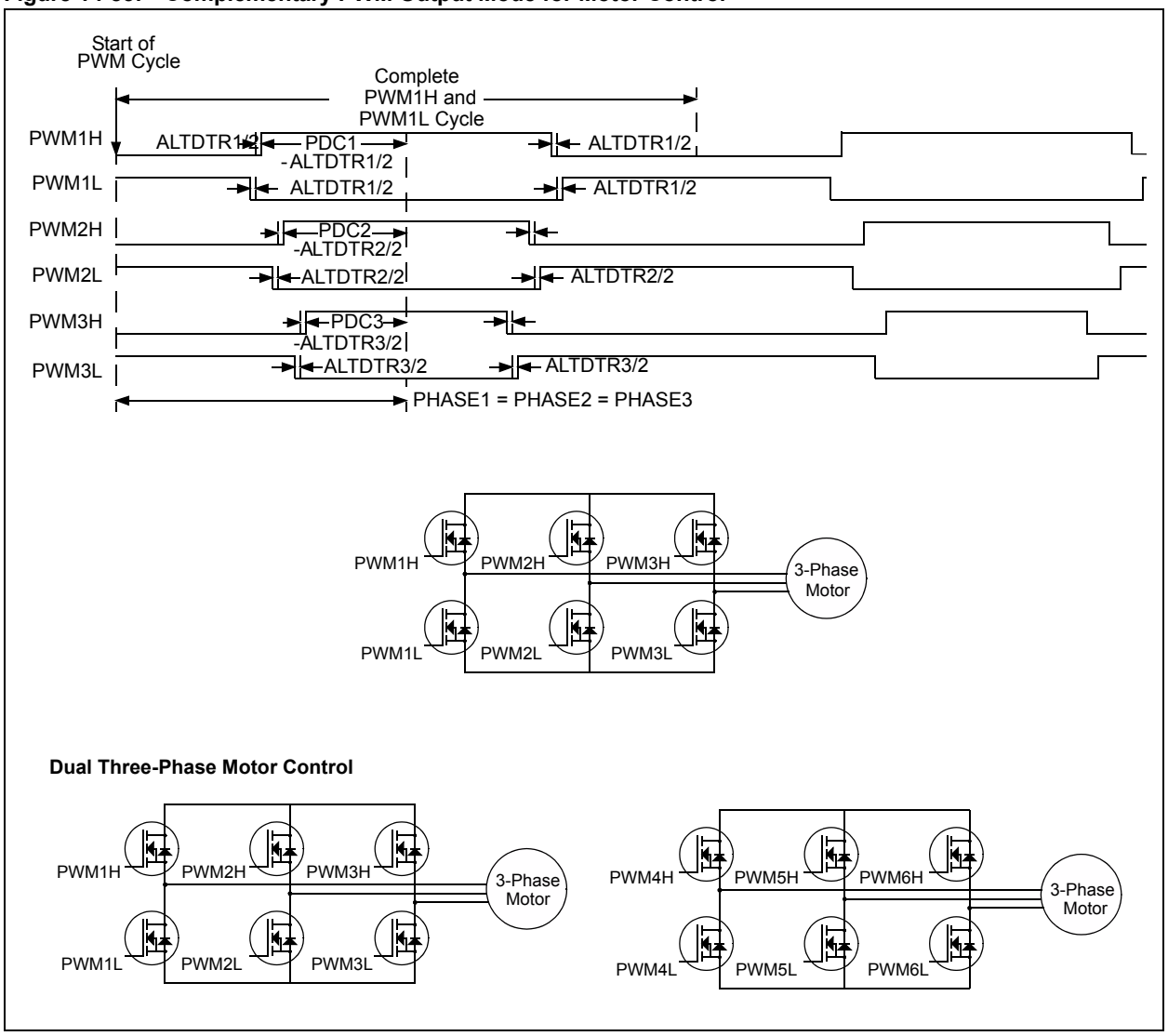


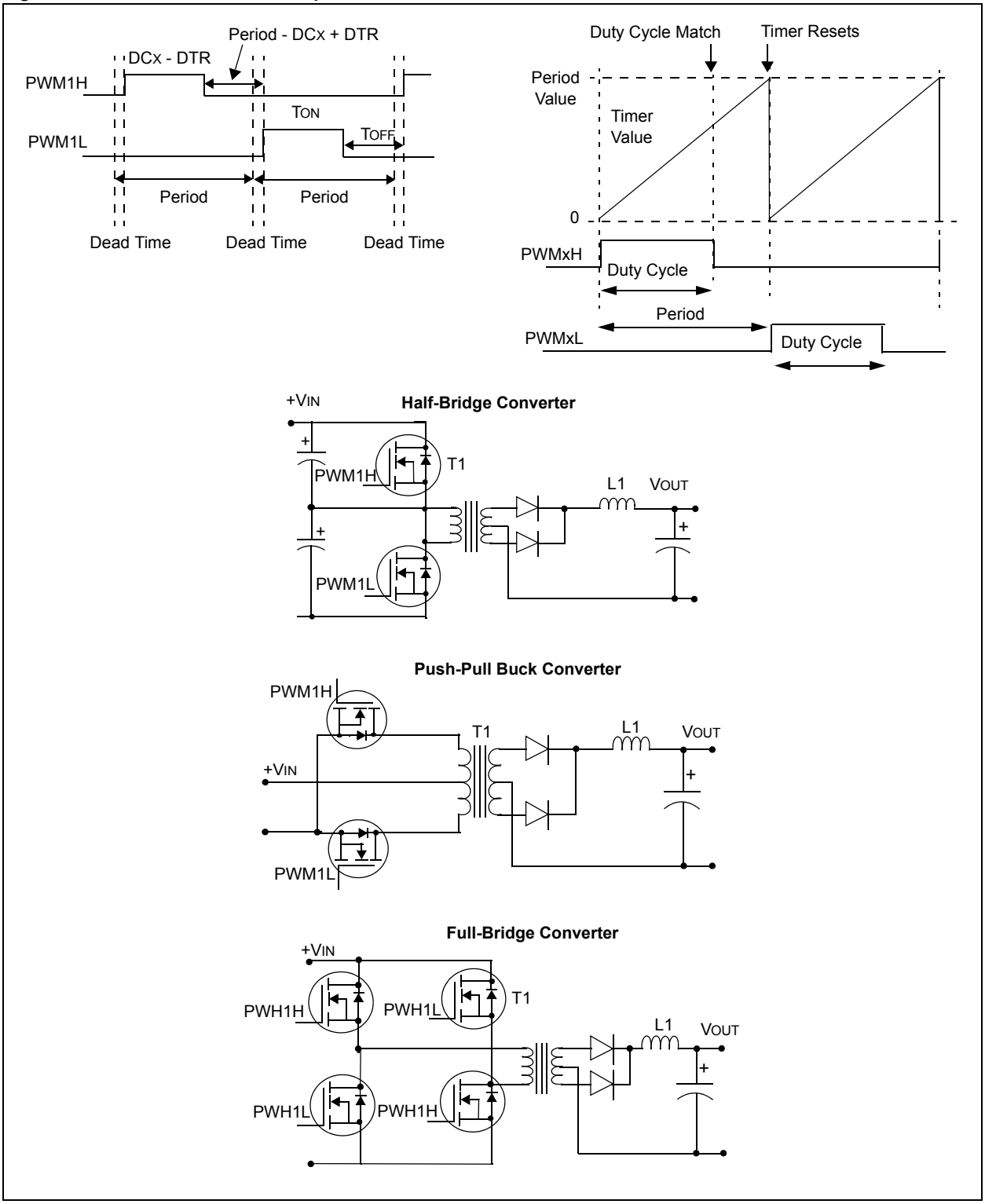
Figure 14-83: Complementary PWM Output Mode for Motor Control



14.16.2 Push-Pull Output Mode

The Push-Pull PWM mode, shown in Figure 14-84, alternately outputs the PWM signal on one of two PWM pins. In this mode, complementary PWM output is not available. This mode is useful in transformer-based power converter circuits that avoid flow of direct current that saturates their cores. The Push-Pull mode ensures that the duty cycle of the two phases is identical, thereby yielding a net DC bias of zero.

Figure 14-84: Push-Pull PWM Output Mode



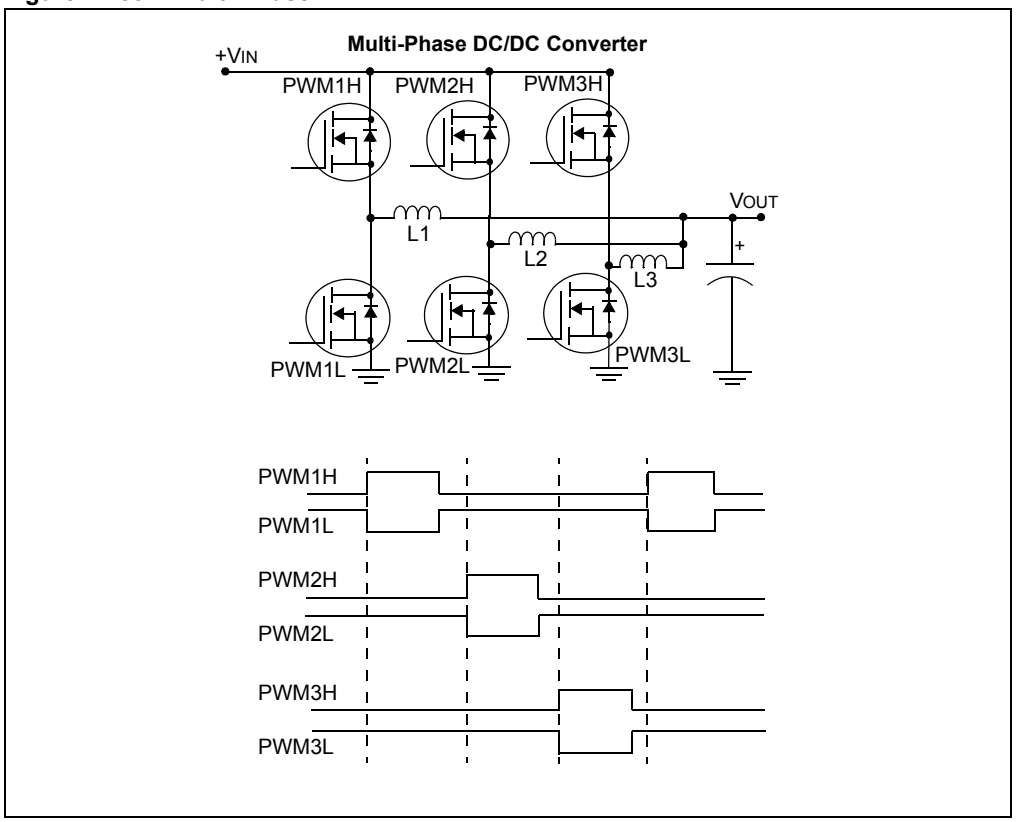
14.16.3 Multi-Phase PWM

The Multi-Phase PWM, shown in Figure 14-85, uses phase shift values in the PHASEx registers to shift the PWM outputs with respect to the primary time base. Because the phase shift values are added to the primary time base, the phase shifted outputs occur earlier than a PWM signal that specifies zero phase shifts. In Multi-Phase mode, the specified phase shift is fixed by the application's design. Phase shift is available in all PWM modes that use the master time base.

Multi-phase PWM is often used in DC-to-DC converters that handle fast load current transients, and need to meet smaller space requirements. A multi-phase converter is essentially a parallel array of buck converters that are operated slightly out of phase with each other. The multiple phases create an effective switching speed equal to the sum of the individual converters.

If a single phase is operating at a PWM frequency of 333 kHz, the effective switching frequency for the circuit, shown in Figure 14-86, is 1 MHz. This high switching frequency greatly reduces input and output capacitor size requirements. It also improves load transient response and ripple figures.

Figure 14-85: Multi-Phase PWM



14.16.3.1 INTERLEAVED POWER FACTOR CORRECTION (IPFC)

The interleaving of multiple boost converters in PFC circuits is becoming popular in the recent applications. Figure 14-86 and Figure 14-87 illustrate the typical interleaved PFC circuit configuration and the interleaved PFC operational waveforms, respectively.

By staggering the channels at uniform intervals, multichannel interleaved PFC can reduce the input current ripple significantly due to the ripple cancellation effect. The smaller input current ripple indicates the low Differential Mode (DM) noise filter. It is generally believed that the reduced Differential mode noise magnitude makes the Differential mode filter smaller. The output capacitor voltage ripples are also reduced significantly as a function of the duty cycle.

Figure 14-86: Interleaved PFC Diagram

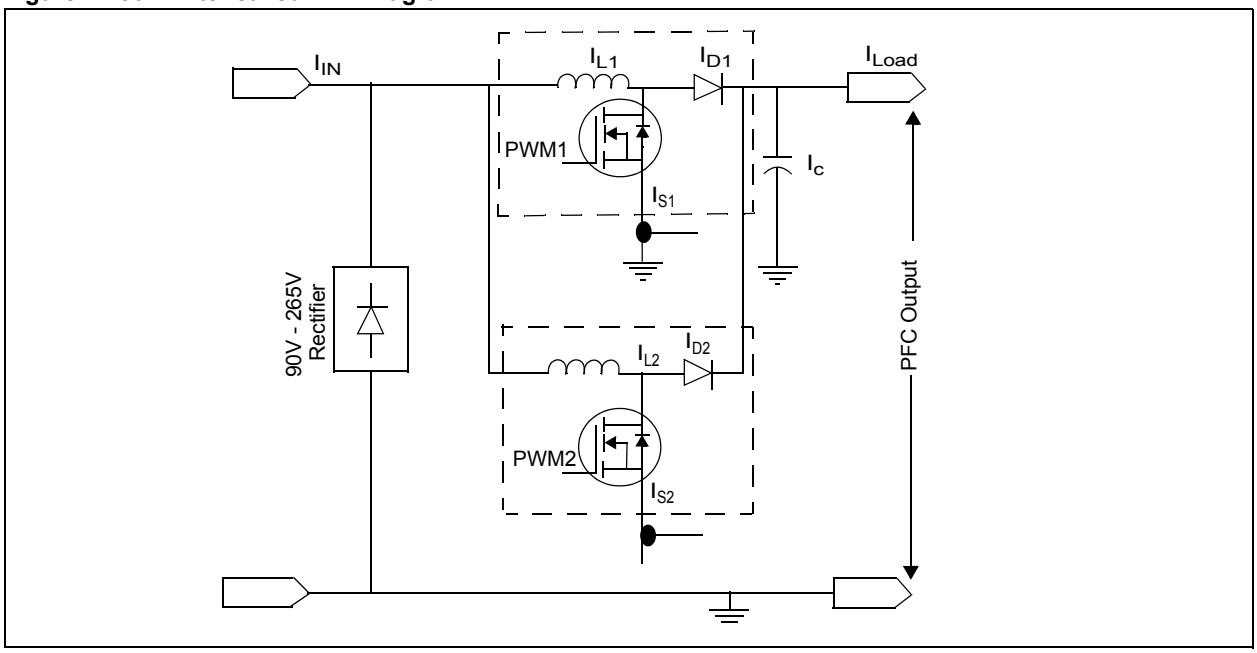


Figure 14-87: Interleaved PFC Operational Waveforms

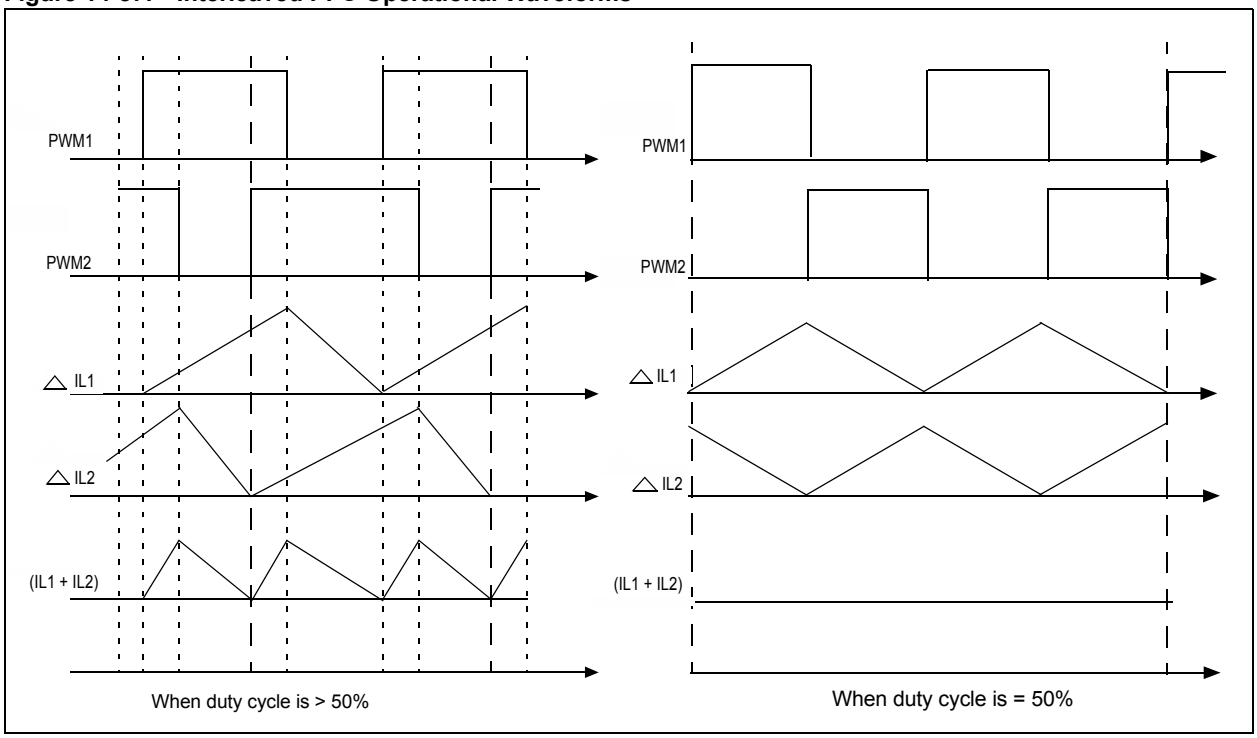


Figure 14-88: Three-Phase Motor Control with Interleaved PFC Diagram

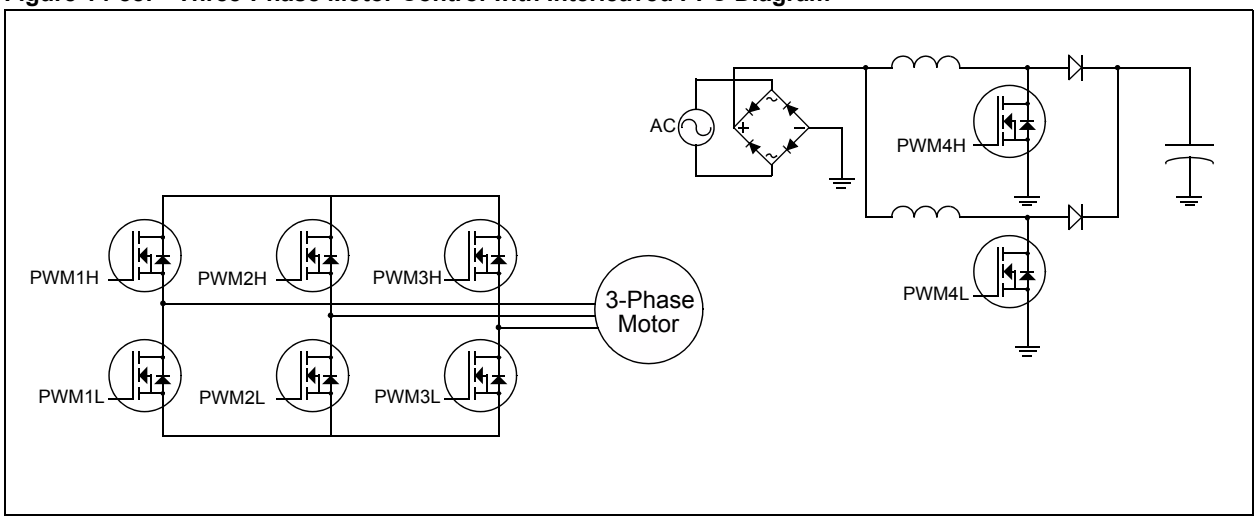
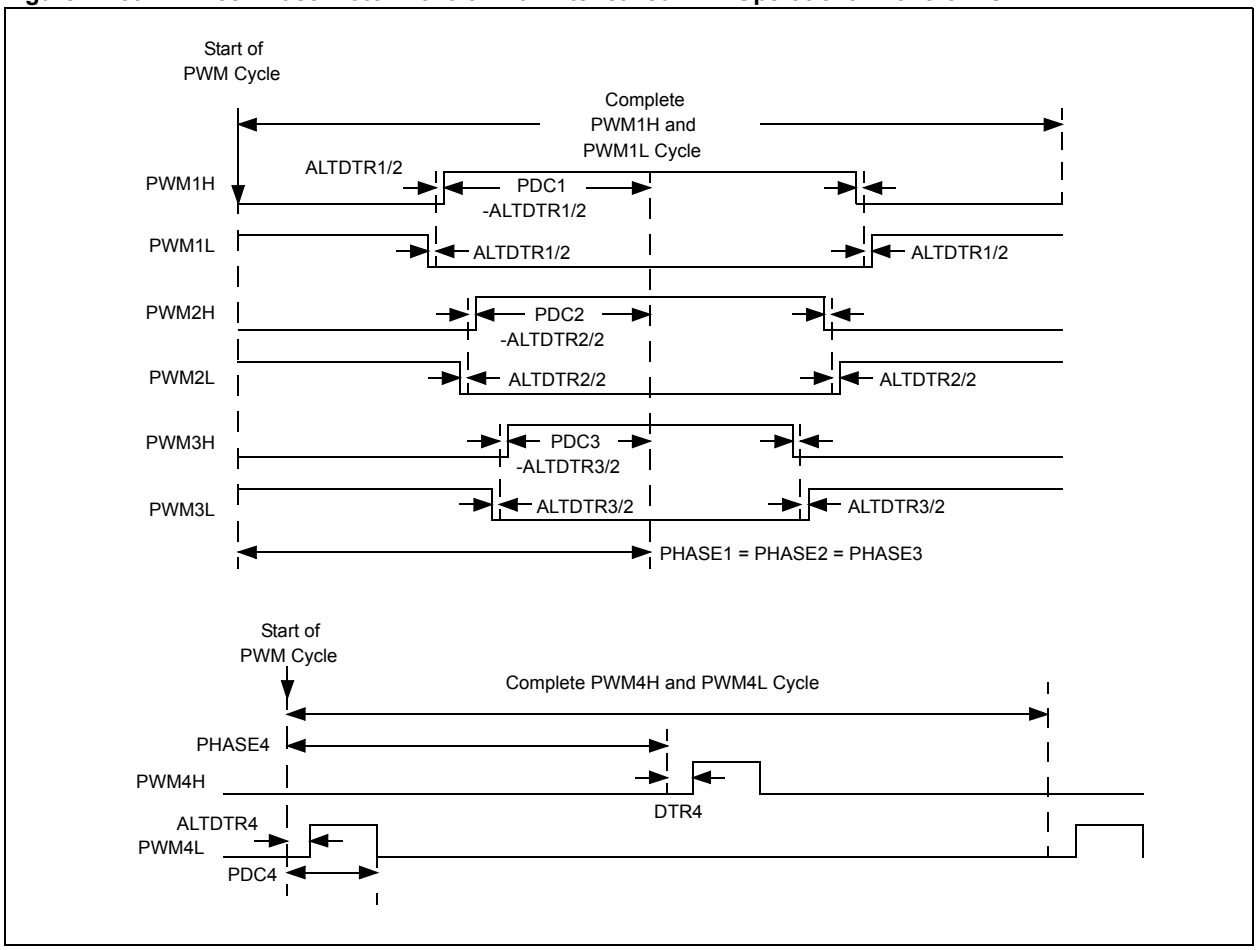


Figure 14-89: Three-Phase Motor Control with Interleaved PFC Operational Waveforms



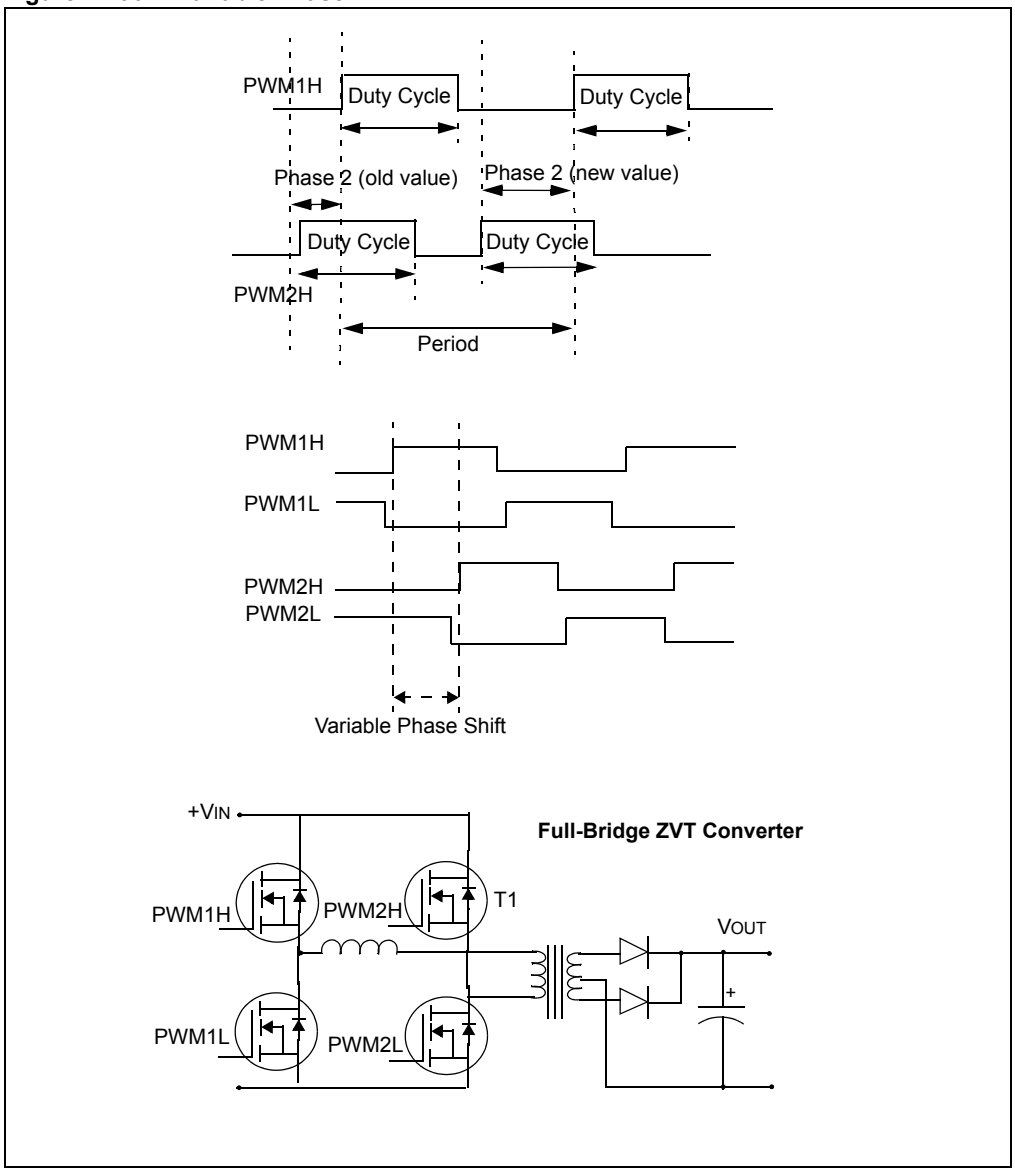
14.16.4 Variable Phase PWM

The Variable Phase PWM shown in Figure 14-90, constantly changes the phase shift among PWM channels to control the flow of power, which is in contrast with most PWM circuits that vary the duty cycle of PWM signal to control power flow. In variable phase applications, the PWM duty cycle is often maintained at 50 percent. The phase shift value is available to all PWM modes that use the master time base.

The variable phase PWM is used in newer power conversion/motor control topologies that are designed to reduce switching losses. In the standard PWM methods, when 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 during the time when the transistor turns ON or OFF and the power loss (V * I * TSW * FPWM) becomes appreciable at high frequencies.

The Zero Voltage Switching (ZVS) and Zero Current Switching (ZVC) circuit topologies attempt to use quasi-resonant techniques that shift either the voltage or the current waveforms relative to each other to change the value of voltage or the current to zero when the transistor turns ON or OFF. If either the current or the voltage is zero, no switching loss occurs.

Figure 14-90: Variable Phase PWM



14.16.5 Current Reset PWM

The Current Reset PWM shown in Figure 14-91 is a variable frequency mode, where the actual PWM period is less than or equal to the specified period value. The independent time base is reset externally after the PWM signal has been deasserted. The Current Reset PWM mode can be used in Constant PWM On-Time mode. To operate in PWM Current Reset, the PWM generator should be in Independent Time Base. If an external Reset signal is not received, the PWM period uses the PHASEx register value by default.

Note: In Current Reset mode, the local time base resetting is based on the leading edge of the current-limit input signal after completion of the PWMxH/L duty cycle.

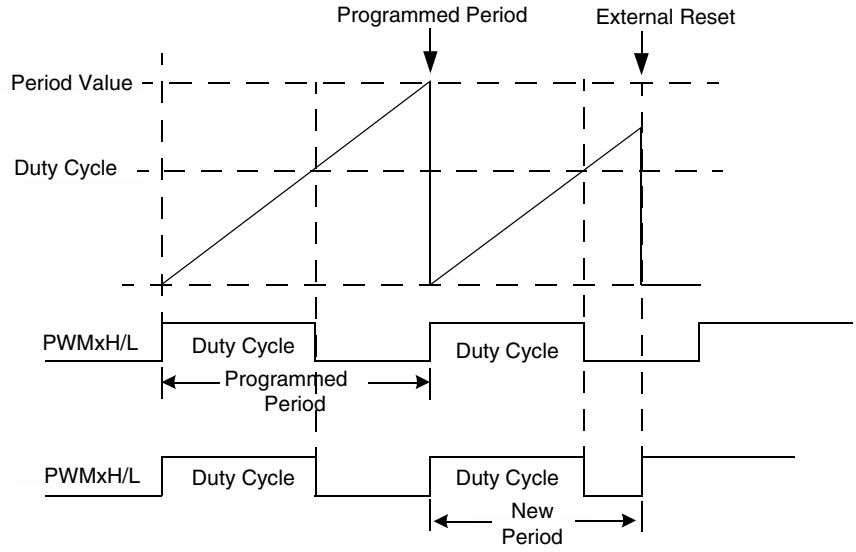
In Current Reset mode, the PWM frequency varies with the load current. This is different from most PWM modes because the user-assigned application sets the maximum PWM period and an external circuit measures the inductor current. When the inductor current falls below a specified value, the external current 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 is called Constant On-Time Variable Frequency PWM output and is used in Critical Conduction mode PFC applications.

This should not be confused with the cycle-by-cycle current-limiting PWM output, where the PWM output 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. Here, the PWM frequency is fixed for a given time base period.

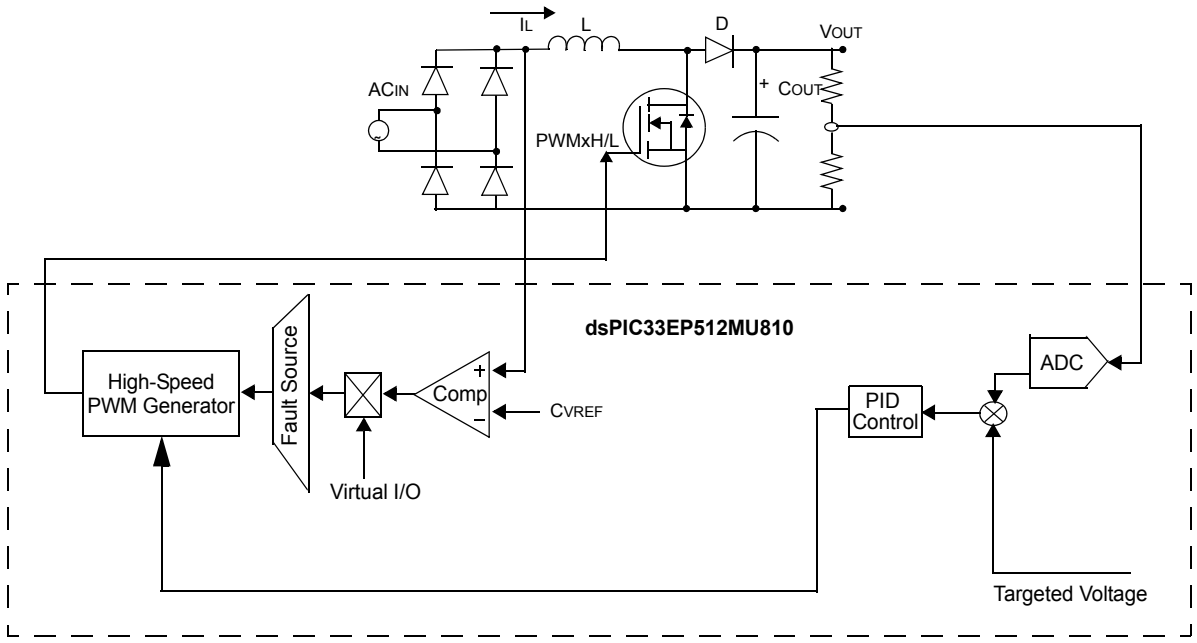
The advantages of Current Reset PWM mode in PFC applications are as follows:

- As the PFC boost inductor does not require to store energy at the end of each switching cycle, a smaller inductor can be used. The usage of the smaller inductor leads to reduced cost.
- Commutation of Boost diode from ON to OFF happens at zero current. The slower diodes can be used to reduce the cost.
- Inner current feedback loop is much faster, as the feedback is received for every cycle

Figure 14-91: Current Reset PWM



External current comparator resets PWM counter. PWM cycle restarts earlier than the programmed period. This is a Constant On-Time Variable Frequency PWM mode.

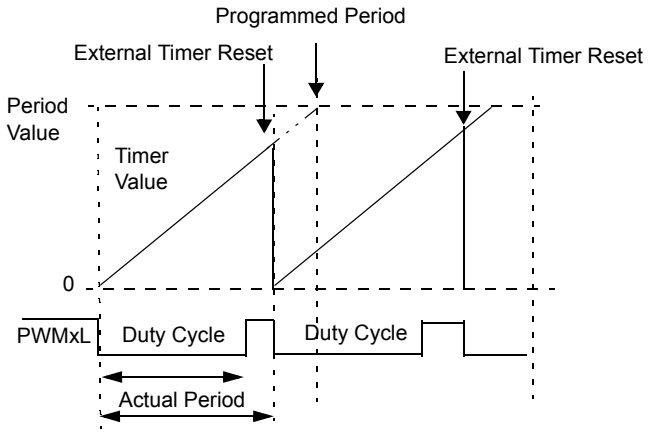


14.16.6 Constant Off-Time PWM

Constant Off-Time PWM, shown in Figure 14-92, is a variable-frequency PWM output where the actual PWM period is less than or equal to the specified period value. The PWM time base resets externally after the PWM signal duty cycle value has been reached and the PWM signal has been deasserted. This is implemented by enabling the On-Time PWM output called Current Reset PWM and using the complementary PWM output (PWMxL).

The Constant Off-Time PWM can be enabled only when the PWM generator operates in independent time base. If an external Reset signal is not received, by default, the PWM period uses the value specified in PHASEx register.

Figure 14-92: Constant Off-Time PWM

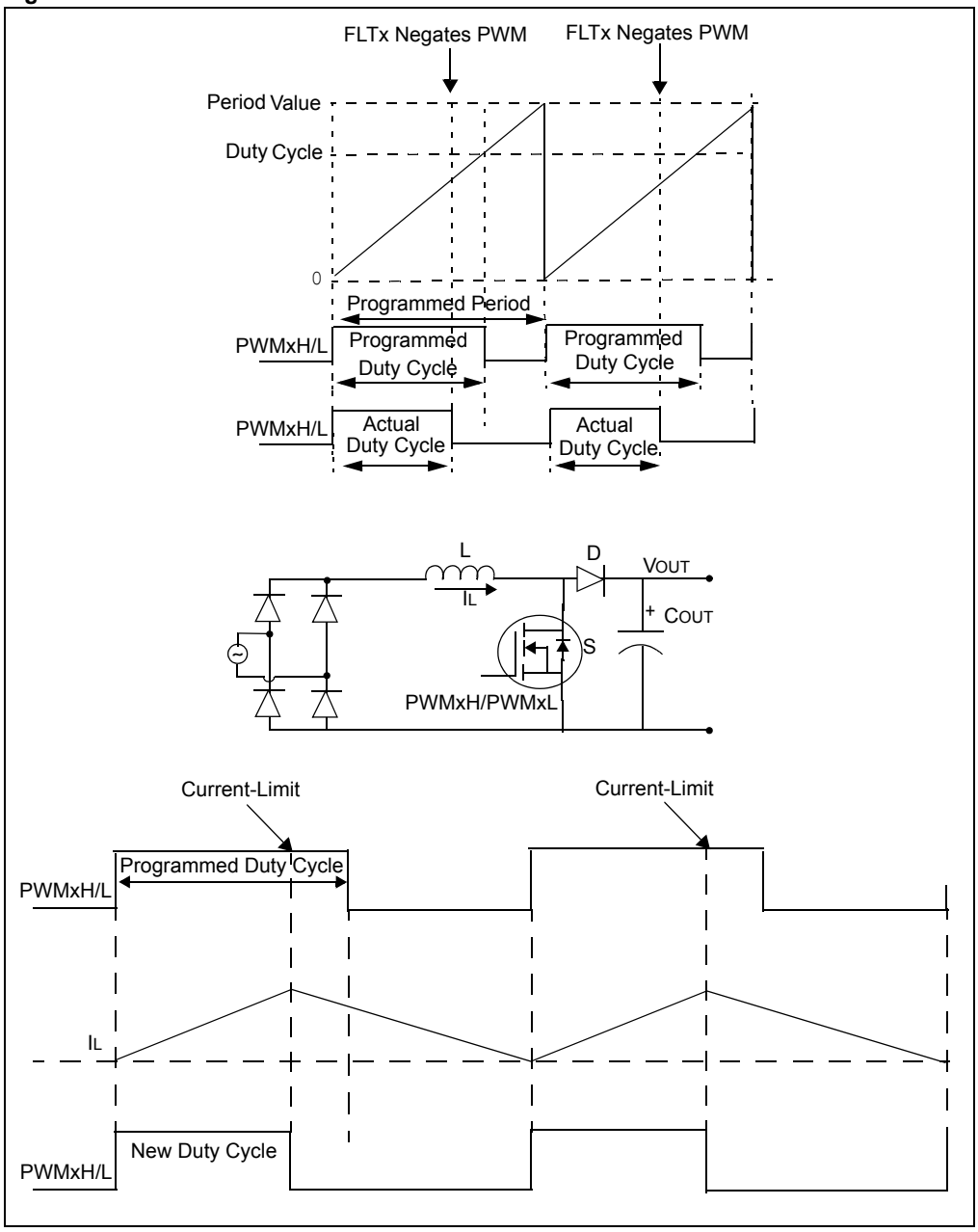


Note: Duty Cycle represents OFF-Time.

14.16.7 Current-Limit PWM

The cycle-by-cycle current-limit shown in Figure 14-93, truncates the asserted PWM signal when the selected external fault signal is asserted. The PWM output values are specified by the CLDAT<1:0> bits (IOCONx<3:2>). The override outputs remain in effect until the beginning of the next PWM cycle. This is sometimes used in the Power Factor Correction (PFC) circuits, where the inductor current controls the PWM On-Time. This is a constant frequency PWM.

Figure 14-93: Current-Limit PWM



14.16.8 Discontinuous or Burst Mode Implementation

In applications where the load current drawn from the converter is much smaller than its nominal current/converter operating at no load, the power drawn from the source can be reduced by forcing the converter to deassert the PWM output by using manual override. Typically, the converter PWM output can be turned off over a period of time based on the output voltage regulation, which can reduce the no load power requirements significantly.

14.17 REGISTER MAP

Table 14-7 maps the bit functions for the High-Speed PWM control registers.

Table 14-7: High-Speed 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
PTCON2	—	—	—	—	—	—	—	—	—	—	—	—	—	PCLKSEL<2:0>			0000
PTPER	PTPER<15:0>																FFF8
STCON	—	—	—	SESTAT	SEIEN	EIPU	SYNCPOL	SYNCOEN	SYNCEN	SYNCSRC<2:0>			SEVTPS<3:0>				0000
STCON2	—	—	—	—	—	—	—	—	—	—	—	—	—	PCLKSEL<2:0>			0000
STPER	STPER<15:0>																FFF8
MDC	MDC<15:0>																0000
SEVTCMP	SEVTCMP<15:0>																0000
SSEVTCMP	SSEVTCMP<15:0>																0000
CHOP	CHPCLKEN	—	—	—	—	—	CHOP<9:0>										0000
PWMKEY	PWMKEY<15:0>																0000
PWMCONx	FLTSTAT	CLSTAT	TRGSTAT	FLTIEN	CLIEN	TRGIEN	ITB	MDCS	DTC<1:0>		DTCP	—	MTBS	CAM	XPRES	IUE	0000
IOCONx	PENH	PENL	POLH	POLL	PMOD<1:0>		OVRENH	OVRENL	OVRDAT<1:0>		FLTDAT<1:0>		CLDAT<1:0>		SWAP	OSYNC	0000
FCLCONx	IFLTMOD	CLSRC<4:0>					CLPOL	CLMOD	FLTSRC<4:0>					FLTPOL	FLTMOD<1:0>		0000
PDCx	PDCx<15:0>																0000
PHASEx	PHASEx<15:0>																0000
SDCx	SDCx<15:0>																0000
SPHASEx	SPHASEx<15:0>																0000
DTRx	—	—	DTRx<13:0>														0000
ALTDTRx	—	—	ALTDTRx<13:0>														0000
TRIGx	TRGCMP<15:0>																0000
TRGCONx	TRGDIV<3:0>				—	—	—	—	—	—	TRGSTRT<5:0>						0000
LEBCONx	PHR	PHF	PLR	PLF	FLTLEBEN	CLLEBEN	—	—	—	—	BCH	BCL	BPHH	BPHL	BPLH	BPLL	0000
LEBDLYx	—	—	—	—	LEB<11:0>												0000
PWMCAPx	PWMCAP1<15:0>																0000
AUXCONx	—	—	—	—	BLANKSEL<3:0>				—	—	CHOPSEL<3:0>				CHOPHEN	CHOPLEN	0000

Legend: x = unknown value on Reset, — = unimplemented, read as '0'. Reset values are shown in hexadecimal.
Note 1: Not all bits are available for all devices. Please refer to the specific data sheet for details.

14.18 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 dsPIC33E/PIC24E product family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the High-Speed PWM module are:

Title	Application Note #
No related application notes are available at this time.	N/A

Note: Please visit the Microchip web site (www.microchip.com) for additional Application Notes and code examples for the dsPIC33E/PIC24E family of devices.

14.19 REVISION HISTORY

Revision A (September 2010)

This is the initial released revision of this document.

Revision B (June 2011)

This revision includes the following updates:

- Added the PWMKEY register description to **14.3 "Control Registers"**
- Added the PWMKEY: PWM Unlock Key register (see Register 14-11)
- Added **14.5.3 "Write Protection"**, which provides information on the write protection feature for the IOCONx and FCLCONx registers
- Added **14.10.1 "Class B Fault"**, which describes the fault implemented for Class B safety features
- Added the PWMKEY register to the HIgh-Speed PWM register map (see Table 14-7)
- Changes to formatting and minor text updates have been incorporated through the document

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

ISBN: 978-1-61341-340-1

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2009

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hangzhou
Tel: 86-571-2819-3180
Fax: 86-571-2819-3189

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-6578-300
Fax: 886-3-6578-370

Taiwan - Kaohsiung
Tel: 886-7-213-7830
Fax: 886-7-330-9305

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

05/02/11