Section 43. High-Speed PWM

HIGHLIGHTS

This section of the manual contains the following topics:

43.1 INTRODUCTION

This section describes the High-Speed PWM module and its associated operational modes. The High-Speed PWM module in the dsPIC33F device family supports a wide variety of PWM modes and is ideal for power conversion applications.

Some of the common applications of the High-Speed PWM module include the following:

- AC-to-DC converters
- Power Factor Correction (PFC)
- Inverters
- DC-to-DC converters
- Battery chargers
- Digital lighting
- Uninterrupted Power Supply (UPS)
- AC and DC motors

43.2 FEATURES

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

- Up to four PWM generators
- Two PWM outputs per PWM generator
- Individual time base and duty cycle for each PWM output
- Duty cycle, dead time, phase shift and frequency resolution of 1.04 ns at 40 MIPS
- Independent fault and current-limit inputs for eight PWM Outputs
- Redundant output
- True Independent output
- Center-Aligned PWM mode
- Output override control
- Special Event Trigger
- PWM capture feature
- Prescaler for input clock
- Dual trigger from PWM to Analog-to-Digital Converter (ADC) per PWM period
- PWMxL and PWMxH output pin swapping
- Remappable PWMxH and PWMxL pins
- Independent PWM frequency, duty cycle and phase shift changes
- Leading-Edge Blanking (LEB) functionality
- PWM Capture functionality

Note:	Duty cycle, Dead-time, Phase shift and Frequency resolution is 8.32 ns in Center-Aligned PWM Mode.

43.3 CONTROL REGISTERS

This section outlines the specific functions of each register that controls 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 Analog-to-Digital Converter (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: Master Time Base Period Register**
  - Provides the PWM time period value
- **SEVTCMP: PWM Special Event Compare Register**
  - Provides the compare value that is used to trigger the ADC module
- **MDC: PWM Master Duty Cycle Register**
  - Provides the PWM master duty cycle value
- **PWMCONx: PWM Control Register**
  - Enables or disables fault interrupt, current-limit interrupt, 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 the Dead Time mode
  - Enables or disables Center-Aligned mode
  - Controls external PWM Reset operation
  - Enables or disables immediate updates of the duty cycle, phase offset and independent time base period
- **PDCx: PWM Generator Duty Cycle Register**
  - Provides the duty cycle value for the PWMxH and PWMxL outputs, if shared 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 output, if master time base is selected
  - Provides the independent time base period for the PWMxH output, if 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
- **SDCx: PWM Secondary Duty Cycle Register**
  - Provides the duty cycle value for the PWMxL output, if independent time base is selected

- **SPHASEx: PWM Secondary Phase Shift Register**
  - 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
- **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
  - Enables or disables the primary PWM trigger event with the secondary PWM trigger event
- **IOCONx: PWM I/O Control Register**
  - Enables or disables PWM pin control feature (PWM control or GPIO)
  - 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 the Current-Limit mode
  - Selects the fault control signal source
  - Configures the fault polarity
  - Enables or disables the Fault mode
- **TRIGx: PWM Primary Trigger Compare Value Register**
  - Provides the compare value to generate the primary PWM trigger
- **STRIGx: PWM Secondary Trigger Compare Value Register**
  - Provides the compare value to generate the secondary PWM 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
- **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 LEB processing on the current-limit input signal is completed

Register 43-1: 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	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SYNCEN(1)	—	SYNCSRC<1:0>(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 **Unimplemented:** Read as ‘0’

bit 5-4 **SYNCSRC<1:0>:** Synchronous Source Selection bits(1)
00 = SYNCI1
01 = SYNCI2
10 = Reserved
11 = Reserved

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

Note 1: These bits should be changed only when PTEN = 0.

Register 43-2: 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
—	—	—	—	—	PCLKDIV<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 **PCLKDIV<2:0>:** PWM Input Clock Prescaler (Divider) Select bits(1)
- `000` = Divide by 1, maximum PWM timing resolution (power-on default)
- `001` = Divide by 2, maximum PWM timing resolution
- `010` = Divide by 4, maximum PWM timing resolution
- `011` = Divide by 8, maximum PWM timing resolution
- `100` = Divide by 16, maximum PWM timing resolution
- `101` = Divide by 32, maximum PWM timing resolution
- `110` = Divide by 64, maximum PWM timing resolution
- `111` = Reserved

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

Register 43-3: PTPER: 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>							
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>							
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>:** Master Time Base (PMTMR) Period Value bits

Note: The PWM time base has a minimum value of 0x0010, and a maximum value of 0xFFFB.

Register 43-4: 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	U-0	U-0	U-0
SEVTCMP<7:3>					—	—	—
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 **SEVTCMP<15:3>:** Special Event Compare Count Value bits

bit 2-0 **Unimplemented:** Read as ‘0’

Register 43-5: 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

Note 1: The smallest pulse-width that can be generated on the PWM output corresponds to a value of 0x0008, while the maximum pulse-width generated corresponds to a value of Period - 0x0008.

2: As the Duty Cycle gets closer to 0% or 100% of the PWM Period (0 to 40 ns, depending on the mode of operation), PWM Duty Cycle resolution will increase from 1 to 3 LSBs.

Register 43-6: 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	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
DTC<1:0>		—	—	—	CAM(2,3)	XPRES	IUE
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(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 FLTSTAT bit is cleared

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

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

bit 9 **ITB:** Independent Time Base Mode bit(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

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

bit 5-3 **Unimplemented:** Read as ‘0’

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).

Register 43-6: PWMCONx: PWM Control Register (Continued)

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

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

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

1 = Updates to the active MDC/PDCx/SDCx registers are immediate
0 = Updates to the active PDCx 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).

Register 43-7: PDCx: PWM Generator 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
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, the PDCx register controls the PWMxH duty cycle only. In the Complementary, Redundant and Push-Pull PWM modes, the PDCx register controls the duty cycle of both the PWMxH and PWMxL.

2: The smallest pulse-width that can be generated on the PWM output corresponds to a value of 0x0008, while the maximum pulse-width generated corresponds to a value of Period - 0x0008.

3: As the Duty Cycle gets closer to 0% or 100% of the PWM Period (0 to 40 ns, depending on the mode of operation), PWM Duty Cycle resolution will increase from 1 to 3 LSBs.

Register 43-8: SDCx: PWM Secondary 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
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. When used in Independent PWM mode, the SDCx register controls the PWMxL duty cycle.

2: The smallest pulse-width that can be generated on the PWM output corresponds to a value of 0x0008, while the maximum pulse-width generated corresponds to a value of Period - 0x0008.

3: As the Duty Cycle gets closer to 0% or 100% of the PWM Period (0 to 40 ns, depending on the mode of operation), PWM Duty Cycle resolution will increase from 1 to 3 LSBs.

Register 43-9: 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 PWMCONx<ITB> = `0`, the following applies based on the mode of operation:
- Complementary, Redundant and Push-Pull Output mode (IOCONx<PMOD> = `00`, `01`, or `10`) PHASEx<15:0> = Phase shift value for PWMxH and PWMxL outputs
- True Independent Output mode (IOCONx<PMOD> = `11`) PHASEx<15:0> = Phase shift value for PWMxL only

2: If PWMCONx<ITB> = `1`, the following applies based on the mode of operation:
- Complementary, Redundant, and Push-Pull Output mode (IOCONx<PMOD> = `00`, `01`, or `10`) PHASEx<15:0> = Independent time base period value for PWMxH and PWMxL
- True Independent Output mode (IOCONx<PMOD> = `11`) PHASEx<15:0> = Independent time base period value for PWMxL only
- The smallest pulse width that can be generated on the PWM output corresponds to a value of 0x0008, while the maximum pulse width generated corresponds to a value of period – 0x0008.

Register 43-10: SPHASEx: PWM Secondary 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
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 (used in Independent PWM mode only)

Note 1: If PWMCONx<ITB> = 0, the following applies based on the mode of operation:
- Complementary, Redundant and Push-Pull Output mode (IOCONx<PMOD> = 00, 01, or 10) SPHASEx<15:0> = Not used
- True Independent Output mode (IOCONx<PMOD> = 11) PHASEx<15:0> = Phase shift value for PWMxL only

2: If PWMCONx<ITB> = 1, the following applies based on the mode of operation:
- Complementary, Redundant and Push-Pull Output mode (IOCONx<PMOD> = 00, 01, or 10) SPHASEx<15:0> = Not used
- True Independent Output mode (IOCONx<PMOD> = 11) PHASEx<15:0> = Independent time base period value for PWMxL only
- The smallest pulse width that can be generated on the PWM output corresponds to a value of 0x0008, while the maximum pulse width generated corresponds to a value of period – 0x0008.

Register 43-11: 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 43-12: 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 43-13: 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

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DTM(1)	—	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

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

bit 11-8 **Unimplemented:** Read as ‘0’

bit 7 **DTM:** Dual Trigger Mode bit(1)

`1` = Secondary trigger event is combined with the primary trigger event to create PWM trigger
`0` = Secondary trigger event is not combined with the primary trigger event to create PWM trigger. Two separate PWM triggers are generated

bit 6 **Unimplemented:** Read as ‘0’

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

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

Note 1: The secondary PWM generator cannot generate PWM trigger interrupts.

Register 43-14: 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>(1)		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>		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(1)
00 = PWM I/O pin pair is in the Complementary Output mode
01 = PWM I/O pin pair is in the Redundant Output mode
10 = PWM I/O pin pair is in the Push-Pull Output mode
11 = PWM I/O pin pair is in the True Independent 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>:** Data 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>:** Data for PWMxH and PWMxL Pins if FLTMOD is Enabled bits
FCLCONx<IFLTMOD> = 0: Normal Fault mode
If Fault active, then FLTDAT<1> provides data for PWMxH
If Fault active, then FLTDAT<0> provides data for PWMxL

FCLCONx<IFLTMOD> = 1: Independent Fault mode
If Current-Limit active, then FLTDAT<1> provides data for PWMxH
If Fault active, then FLTDAT<0> provides data for PWMxL

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

Register 43-14: IOCONx: PWM I/O Control Register (Continued)

bit 3-2 **CLDAT<1:0>:** Data for PWMxH and PWMxL Pins if CLMOD is Enabled bits
FCLCONx<IFLTMOD> = 0: Normal Fault mode
If current-limit active, then CLDAT<1> provides data for PWMxH
If current-limit active, then CLDAT<0> provides data for PWMxL

FCLCONx<IFLTMOD> = 1: Independent Fault mode
CLDAT<1:0> is ignored

bit 1 **SWAP:** SWAP PWMxH and PWMxL pins bit
1 = PWMxH output signal is connected to PWMxL pins; PWMxL output signal is connected to PWMxH pins
0 = PWMxH and PWMxL 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 should not be changed after the PWM module is enabled (PTEN = 1).

Register 43-15: 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	U-0	U-0	U-0
TRGCMP<7:3>					—	—	—
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 **TRGCMP<15:3>:** 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.

bit 2-0 **Unimplemented:** Read as ‘0’

Register 43-16: 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)]					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)]					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 mode maps CLDAT<1:0> bits to the PWMxH and PWMxL outputs. The PWM Fault mode maps FLTDAT<1:0> to the PWMxH and PWMxL outputs.

bit 14-10 **CLSRC<4:0>:** Current-Limit Control Signal Source Select bits for PWM Generator #[(2)]

00000 = Fault 1
00001 = Fault 2
00010 = Fault 3
00011 = Fault 4
00100 = Fault 5
00101 = Fault 6
00110 = Fault 7
00111 = Fault 8
01000 = Reserved
•
•
•
11111 = Reserved

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 clock selection during operation will yield unpredictable results.

2: 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.

3: 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.

Register 43-16: FCLCONx: PWM Fault Current-Limit Control Register (Continued)

bit 7-3 **FLTSRC<4:0>:** Fault Control Signal Source Select bits for PWM Generator #
00000 = Fault 1
00001 = Fault 2
00010 = Fault 3
00011 = Fault 4
00100 = Fault 5
00101 = Fault 6
00110 = Fault 7
00111 = Fault 8
01000 = Reserved
•
•
•
11111 = Reserved

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

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

2: 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.

3: 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.

43
High-Speed PWM

Register 43-17: STRIGx: PWM Secondary 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
STRGCMP<15:8>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
STRGCMP<7:3>					—	—	—
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 **STRGCMP<15:3>:** Secondary Trigger Control Value bits

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

bit 2-0 **Unimplemented:** Read as ‘0’

Register 43-18: LEBCONx: Leading-Edge Blanking 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
PHR	PHF	PLR	PLF	FLTLEBEN	CLLEBEN	LEB<9:8>	
bit 15							bit 8

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

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 9-3 **LEB<6:0>:** Leading-Edge Blanking for current-limit and fault input bits
Value is 8.32 ns increments.

bit 2-0 **Unimplemented:** Read as ‘0’

43

High-Speed PWM

Register 43-19: 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>							
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	U-0	U-0	U-0
PWMCAP<7:3>					—	—	—
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 **PWMCAP<15:3>:** 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.

bit 2-0 **Unimplemented:** Read as ‘0’

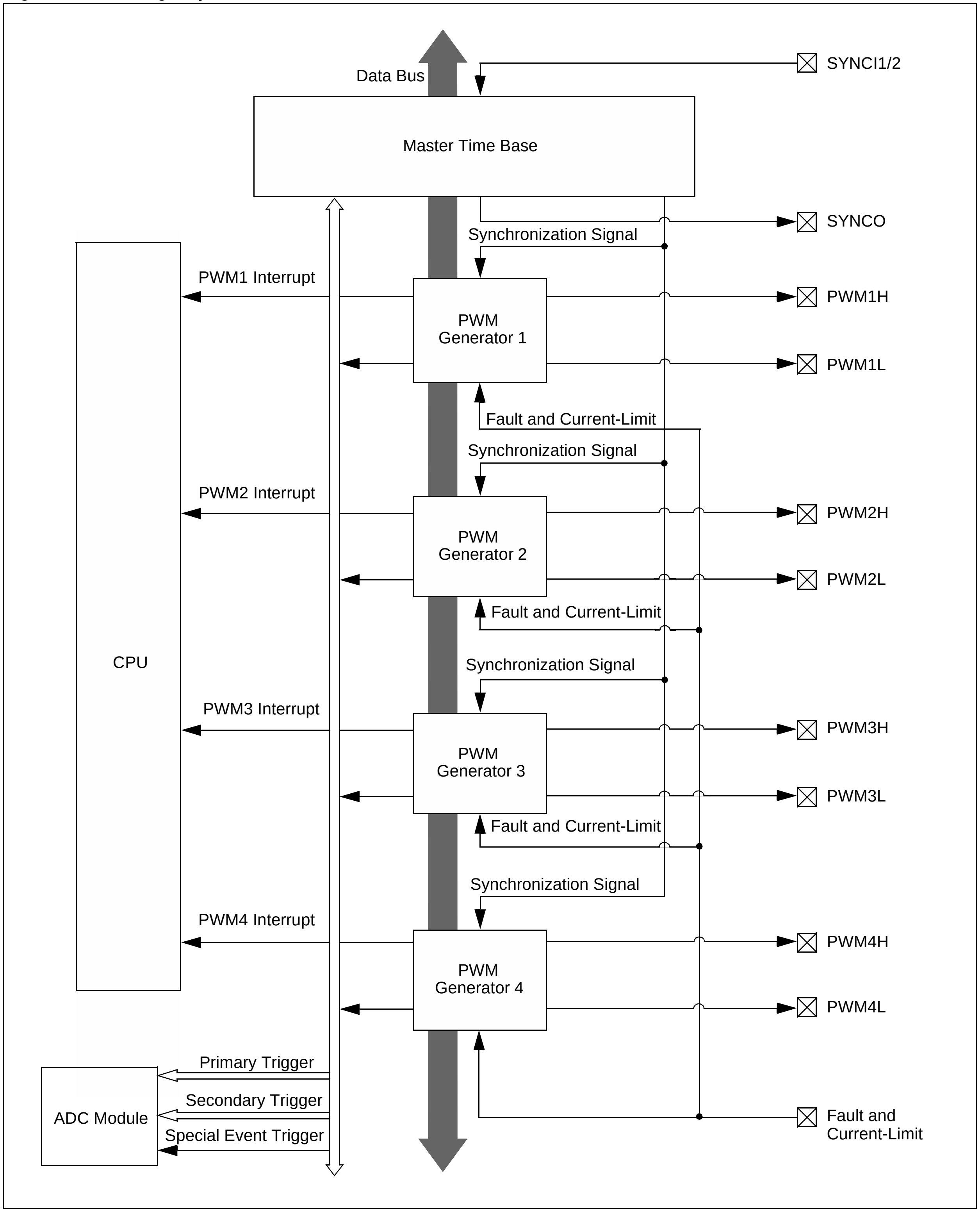
Note: If PWMCONx<ITB> = 0, the following applies based on the mode of operation:

- Complementary, Redundant and Push-Pull Output mode (IOCONx<PMOD> = 00, 01, or 10)
- PHASEx<15:0> = Phase shift value for PWMxH and PWMxL outputs
- True Independent Output mode (IOCONx<PMOD> = 11)
- PHASEx<15:0> = Phase shift value for PWMxH only
- If PWMCONx<ITB> = 1, the following applies based on the mode of operation: Complementary, Redundant and Push-Pull Output mode (IOCONx<PMOD> = 00, 01, or 10)
- PHASEx<15:0> = Independent time base period value for PWMxH and PWMxL
- True Independent Output mode (IOCONx<PMOD> = 11)
- PHASEx<15:0> = Independent time base period value for PWMxH only

43.4 ARCHITECTURE OVERVIEW

Figure 43-1 shows an architectural overview of the High-Speed PWM module and its interconnection with the CPU and other peripherals.

Figure 43-1: High-Speed PWM Module Architectural Overview



The High-Speed PWM module contains up to four 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 synchronization 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, can monitor and protect the system by placing the PWM outputs into a known “safe” state.

Each PWM can generate 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.

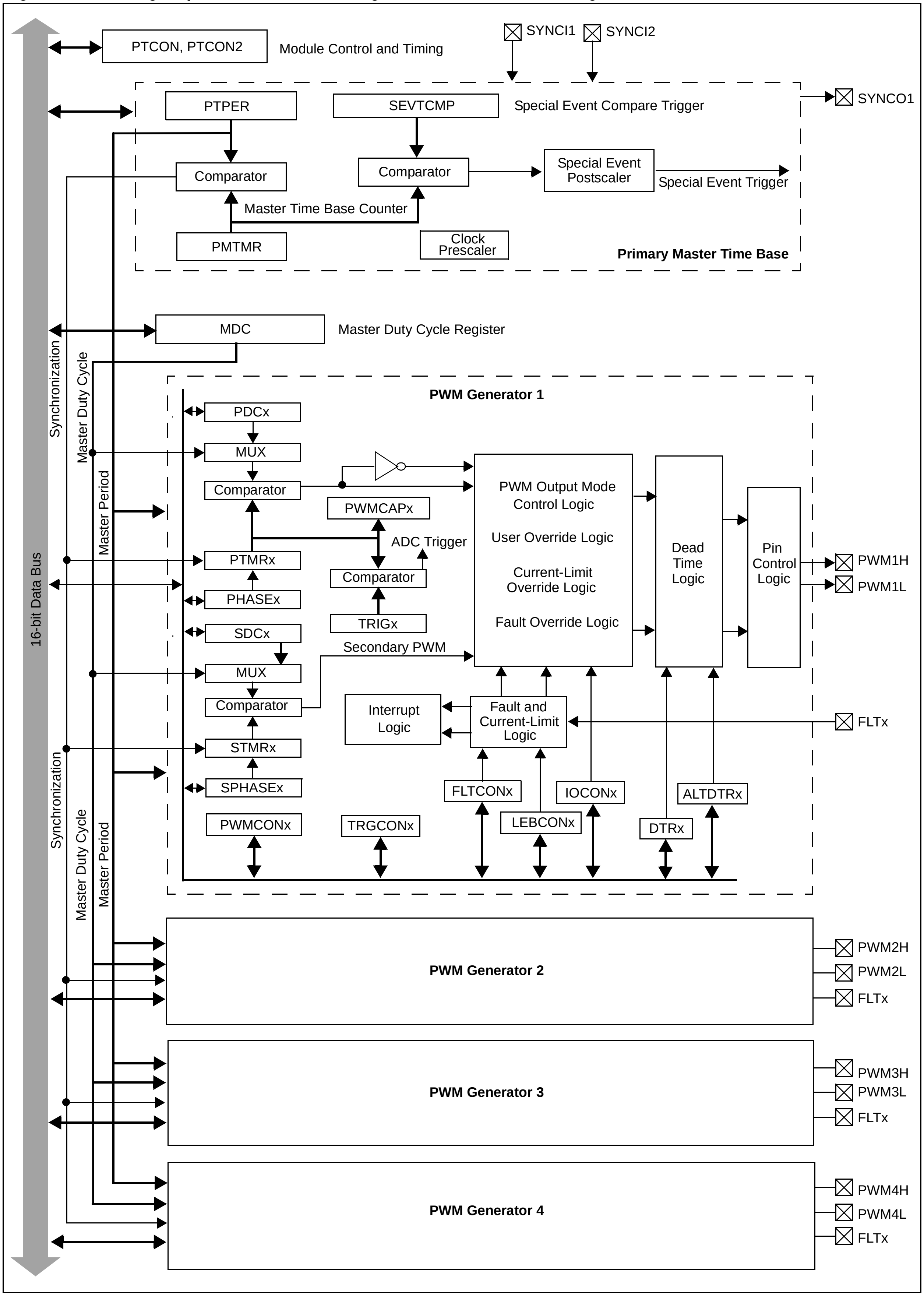
The High-Speed PWM module can synchronize itself with an external signal or can act as a synchronizing source to any external device. The SYNCI1 and SYNCI2 pins are the input pins, which can synchronize the High-Speed PWM module with an external signal. The SYNCO pin is an output pin that provides a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

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

- 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 High-Speed PWM module function is described in detail in subsequent sections. Figure 43-2 shows the interconnections between various registers in the High-Speed PWM module.

Figure 43-2: High-Speed PWM Module Register Interconnection Diagram



43.5 MODULE DESCRIPTION

43.5.1 PWM Clock Selection

The auxiliary clock generator must be used to generate the clock for the PWM module independent of the system clock. The Primary Oscillator Clock (POSCCLK) and Internal FRC Clock (FRCCLK) can be used with an auxiliary PLL to obtain the Auxiliary Clock (ACLK). The auxiliary PLL has a fixed 16x multiplication factor.

The Auxiliary Clock Control (ACLKCON) register selects the reference clock and enables the auxiliary PLL and output dividers for obtaining the necessary auxiliary clock. Equation 43-1 shows the relationship between the reference clock input frequency and the Auxiliary Clock (ACLK) frequency.

Equation 43-1:

$$ACLK = (REFCLK * M)/N$$

where,

$REFCLK$ = Internal FRC Clock frequency (7.37 MHz), if the Internal FRC is selected as clock source

$REFCLK$ = Primary Oscillator Clock frequency (POSCCLK), if the Primary Oscillator is selected as clock source

M = 16, if the auxiliary PLL is enabled by setting the ENAPLL (ACLKCON<15>) bit

M = 1, if the auxiliary PLL is disabled

N = Postscaler ratio selected by the Auxiliary Postscaler (APSTSCLR<2:0>) bits in the Auxiliary Clock Control (ACLKCON<2:0>) register.

Note: The nominal input clock to the PWM should be 120 MHz. Refer to the “Electrical Characteristics” section in the specific device data sheet for the full operating range.

Example 43-1: PWM Clock Code

```
/* Setup for the PWM clock to use the FRC as the REFCLK */
/* ((FRC * 16) / APSTSCLR) = (7.37 * 16) / 1 = 117.9 MHz */
ACLKCONbits.FRCSEL = 1;   /* FRC is input to Auxiliary PLL */
ACLKCONbits.SELACLK = 1;  /* Auxiliary Oscillator provides the clock source */
ACLKCONbits.APSTSCLR = 7; /* Divide Auxiliary clock by 1 */
ACLKCONbits.ENAPLL = 1;   /* Enable Auxiliary PLL */
while(ACLKCONbits.APLLCK != 1); /* Wait for Auxiliary PLL to Lock */
```

The auxiliary clock for the PWM module can be derived from the system clock while the device is running in the primary PLL mode. Equation 43-2 gives the relationship between the Primary PLL Clock (PLLCLK) frequency and the Auxiliary Clock (ACLK) frequency.

Equation 43-2:

$$ACLK = (PLLCLK)/N$$

where,

N = Postscaler ratio selected by the Auxiliary Postscaler (APSTSCLR<2:0>) bits in the Auxiliary Clock Control (ACLKCON<2:0>) register.

Note: If the primary PLL is used as a source for the auxiliary clock, then the primary PLL should be configured up to a maximum operation of 30 MHz or less.

Example 43-2: Using Primary Oscillator for Setting the ACLK

```
/* Setup for the PWM clock to use the primary oscillator as the REFCLK */
/*((primary oscillator* 16) / APSTSCLR) = (8 * 16) / 1 = 120 MHz */
ACLKCONbits.ARCSEL = 1;    /* Primary Oscillator is the Clock Source */
ACLKCONbits.FRCSEL = 0;    /* Input clock source is determined by ASRCSEL bit setting */
ACLKCONbits.SELACLK = 1;   /* Auxiliary Oscillator provides the clock source */
ACLKCONbits.APSTSCLR = 7;  /* Divide Auxiliary clock by 1 */
ACLKCONbits.ENAPLL = 1;    /* Enable Auxiliary PLL */
while(ACLKCONbits.APLLCK != 1); /* Wait for Auxiliary PLL to Lock */
```

Refer to **Section 42. "Oscillator (Part IV)"** (DS70307), for more information on configuring the auxiliary clock generator.

43.5.2 Time Base

Each PWM output in a PWM generator can use either the master time base or an independent time base. The High-Speed PWM module input clock has prescaler (divider) options of 1:1 to 1:64, which can be selected using the PWM Input Clock Prescaler (Divider) Select (PCLKDIV) bits in the PWM Clock Divider Select (PTCON2<2:0>) register. 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 1.04 ns.

For example:

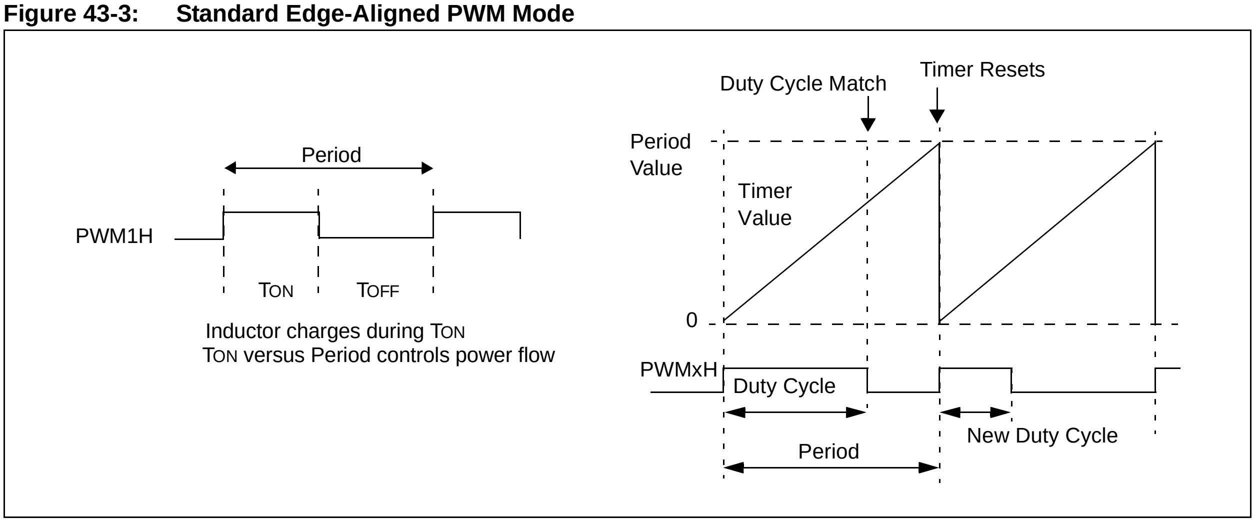
- If a Prescaler option of 1:2 is selected, the PWM duty cycle and period resolution can be set at 2.08 ns. Therefore, the High-Speed PWM module's power consumption would be reduced by approximately 50 percent of the maximum speed operation.
- If a Prescaler option of 1:4 is selected, the PWM duty cycle and period resolution can be set at 4.16 ns. Therefore, the High-Speed PWM module's power consumption would be reduced by approximately 75 percent of the maximum speed operation.

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

43.5.3 Standard Edge-Aligned PWM

Standard edge-aligned PWM waveforms are shown in Figure 43-3. To create the edge-aligned PWM, a timer or counter circuit counts upward from zero to a specified maximum value, called the Period. Another register contains the duty cycle value, which is constantly compared with the timer (period) 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 is greater than or equal to the period value, the timer resets itself, and the process repeats.

Figure 43-3: Standard Edge-Aligned PWM Mode



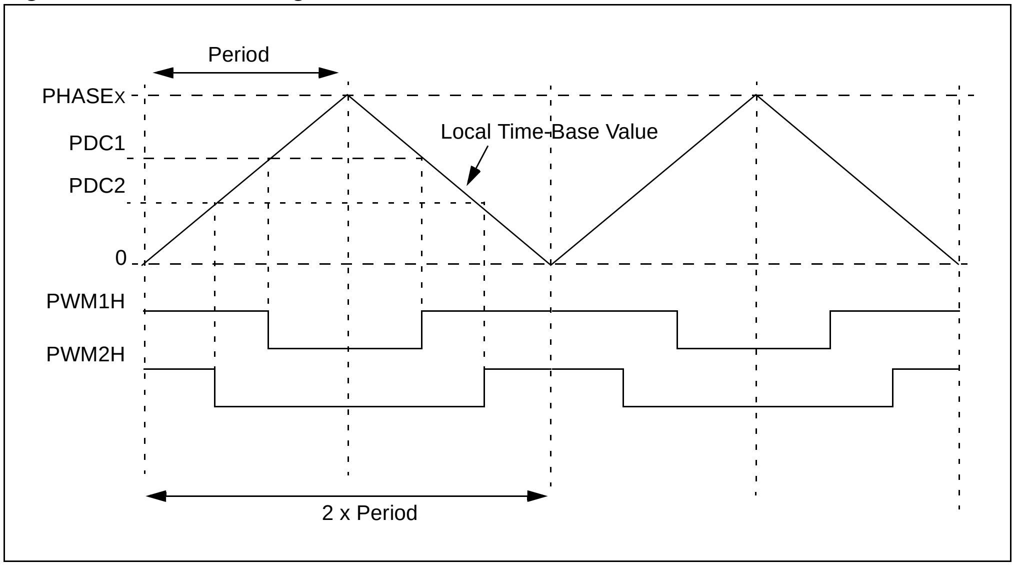
43.5.4 Center-Aligned PWM

The Center-Aligned PWM waveforms shown in Figure 43-4, 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 Center-Aligned Mode Enable (CAM) bit in the PWM Control (PWMCONx<2>) register is set.

When operating in Center-Aligned mode, the effective PWM period will be twice the value that is 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 43-4: Center-Aligned PWM Mode



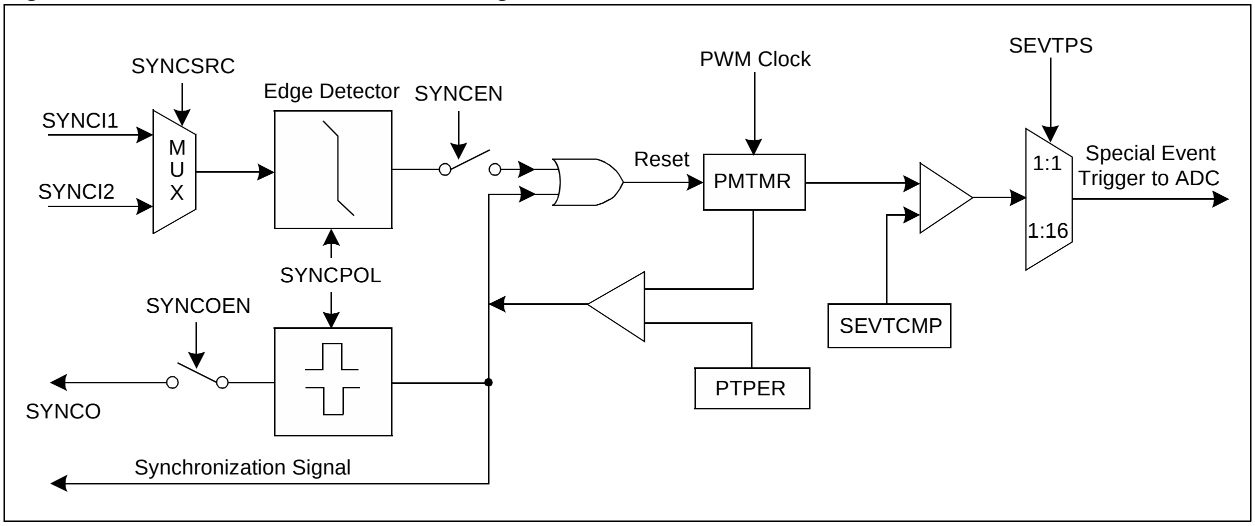
Example 43-3: Edge-Aligned or Center-Aligned Mode Selection

```
/* Select Edge-Aligned or Center-Aligned PWM Time Base */
/* Choose one of the modes given below */

PWMCON1bits.CAM = 0; // For Edge-Aligned Mode

PWMCON1bits.CAM = 1; // For Center-Aligned Mode
```

43.5.5 Master Time Base/Synchronous Time Base

Figure 43-5 shows PWM functionality in the master time base.

Figure 43-5: 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 Time Period (PTPER) register. 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.

43.5.6 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 Synchronous Source Selection (SYNCSRC<1:0>) bits in the PWM Time Base Control (PTCON<5:4>) register. The Synchronize Input Polarity (SYNCPOL) bit in the PWM Time Base Control (PTCON<9>) register 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 External Time Base Synchronization Enable (SYNCEN) bit in the PWM Time Base Control (PTCON<7>) register. The pulse-width of the external synchronization signal (SYNCI1/SYNCI2) should be more than 200 ns to ensure the reliable detection by the master time base.

The external device can also be synchronized with the master time base using the synchronization output signal (SYNCO). The SYNCO signal is generated when the Period register (PTPER) resets the PMTMR register. The polarity of the SYNCO signal is determined by the SYNCPOL bit in the PTCON register. The SYNCO signal can be enabled or disabled by selecting the Primary Time Base Sync Enable (SYNCOEN) (PTCON<8>) bit in the PTCON register.

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

Example 43-4: 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 43-5: 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
```

43.5.7 Special Event Trigger

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

The master Special Event Trigger value is loaded into the PWM Special Event Compare (SEVTCMP) register. In addition, the PWM Special Event Trigger Output Postscaler Select (SEVTPS) bits in the PWM Time Base Control (PTCON<3:0>) register control 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 control bits in the PTCON register.

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 (SEIEN) bit
- If the compare value in SEVTCMP register is a value from zero to a maximum value of PTPER register

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

- Any device Reset
- When PTEN = 0

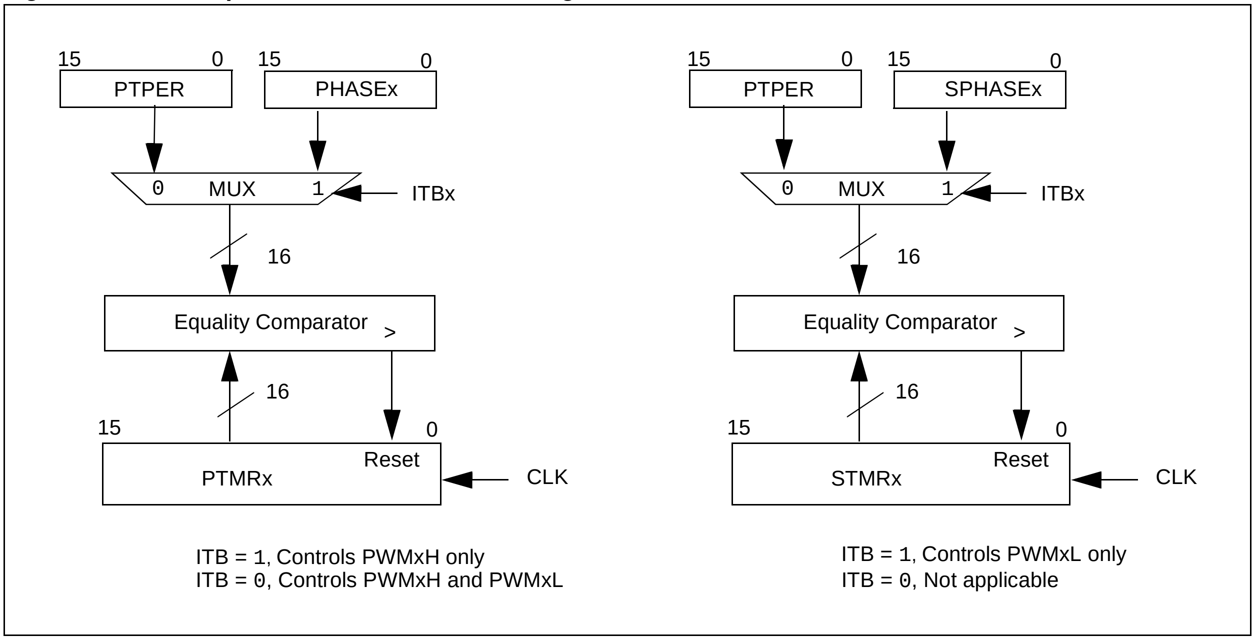
Example 43-6: ADC Special Event Trigger Configuration

```
/* ADC Special Event Trigger configuration */

SEVTCMP = 1248;                 // Special Event Trigger value set at 25%
                                // of 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
```

43.5.8 Independent PWM Time Base

Figure 43-6 shows PWM functionality in the independent time base.

Figure 43-6: 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 (PHASEx) register.
- 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 PHASEx register. 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 application.

43.6 PWM GENERATOR

This section describes the functionality of the PWM generator.

43.6.1 PWM Period

The PWM period value defines the switching frequency of the PWM pulses. The PWM period value can be controlled either by the Master Time Period (PTPER) register, 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 of the PWM output signals (PWMxH and PWMxL).
- In the True Independent Output mode, the PHASEx register controls the PWM period of the PWMxH output signal and the SPHASEx register controls the PWM period of the PWMxL output signal.

Refer to **43.9 "PWM Operating Modes"**, for detailed information about various PWM modes and their features.

When the High-Speed PWM module operates in the 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 the 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 application. The PWM time period (PTPER) can be determined using Equation 43-3.

Equation 43-3: PERIOD, PHASEx and SPHASEx Register Value Calculation

$$PTPER,\ PHASEx,\ SPHASEx = \frac{REFCLK}{7.37\ MHz} * \frac{Desired\ PWM\ Period}{1.04\ ns * PWM\ Input\ Clock\ Prescaler}$$

where, $Desired\ PWM\ Period = \dfrac{1}{PWM\ Frequency}$

Based on Equation 43-3, 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 43-7.

Example 43-7: PWM Time Period Calculation

$$PTPER = \frac{10\mu s}{1.04ns \times 2} = 4808 = 0x12C8$$

where, $PWM\ Frequency$ = 100 kHz

$PWM\ Input\ Clock\ Prescaler$ = 1:2

$REFCLK = FRC = 7.37\ MHz$

The maximum available PWM period resolution is 1.04 ns. The PWM Input Clock Prescaler (Divider) Select (PCLKDIV<2:0>) bits in the PWM Clock Divider Select (PTCON2<2:0>) register determine the type of PWM clock. The timer/counter is enabled or disabled by setting or clearing the PWM Module Enable (PTEN) bit in the PWM Clock Divider Select (PTCON<15>) register. The PMTMR timer can also be cleared using the PTEN bit.

If the Enable Immediate Period Updates (EIPU) bit in the PWM Clock Divider Select (PTCON<10>) register 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 43-8: Clock Prescaler Selection

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

PTCON2bits.PCLKDIV = 1;
```

Example 43-9: 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 43-10: 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;
```


43.6.2 PWM Duty Cycle Control

The duty cycle determines the period of time that 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 Immediate Update Enable (IUE) bit in the PWM Control (PWMCONx<0>) register. If the IUE bit is ‘0’, the active register updates at the start of the next PWM cycle.

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

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

43.6.2.1 MASTER DUTY CYCLE (MDC)

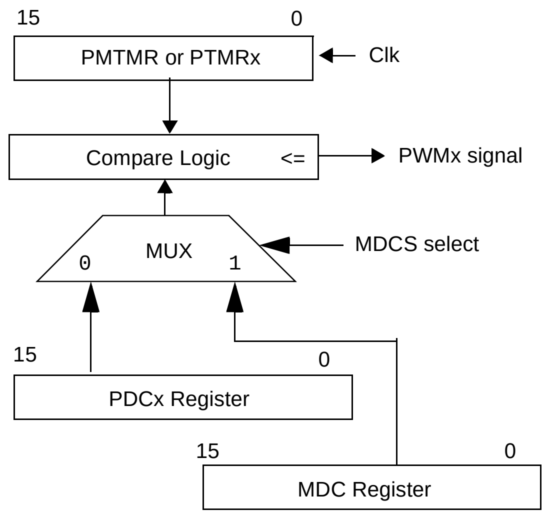
The master duty cycle is controlled by the master time base generator. The Master Duty Cycle Register Select (MDCS) bit in the PWM Control (PWMCONx<8>) register determines whether the duty cycle of each of the PWMxH and PWMxL outputs are controlled by the PWM Master Duty Cycle (MDC) register or the PWM 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.

43.6.2.2 PRIMARY DUTY CYCLE (PDCx)

The primary duty cycle is controlled by the independent time base when the Independent Time Base Mode (ITB) bit in the PWM Control (PWMCONx<9>) register is set to ‘1’. The PDCx register is an input register that provides the duty cycle value for the primary PWM output (PWMxH) signal.

Figure 43-7: Primary Duty Cycle Comparison

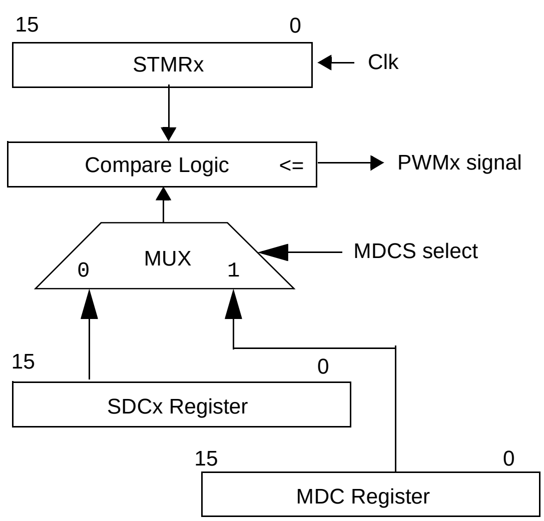


Note: In Independent Output mode, PDCx affects PWMxH only.

43.6.2.3 SECONDARY DUTY CYCLE (SDCx)

The secondary duty cycle is controlled by the independent time base in the PWMCONx register 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 (PWMxL) signal.

Figure 43-8: Secondary Duty Cycle Comparison



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

The Duty Cycle can be determined using Equation 43-4.

Equation 43-4: MDC, PDCx and SDCx Calculation

$$MDC, PDCx, SDCx = \frac{REFCLK}{7.37\ MHz} * \frac{Desired\ Duty\ Cycle}{1.04\ ns * PWM\ Input\ Clock\ Prescaler}$$

where, $Desired\ PWM = \dfrac{1}{PWM\ Frequency}$

> **Note 1:** If a duty cycle value is smaller than the minimum value (0x0008), a signal will have zero duty cycle. A value of 0x0008 is the minimum usable duty cycle value that produces an output pulse from the PWM generators.
>
> **2:** A duty cycle value greater than (Period - 0x0008) produces 100 percent duty cycle.
>
> **3:** 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 43-4, 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.

43.6.2.4 DUTY CYCLE RESOLUTION

The PWM duty cycle and period resolution is 1.04 ns per Least Significant Byte (LSB) with the PWM clock configured for the highest prescaler setting. The PWM duty cycle bit resolution can be determined using Equation 43-5:

Equation 43-5: Bit Resolution Calculation

$$Bit\ Resolution = log_2\left[\frac{REFCLK}{7.37\ MHz} * \frac{PWM\ Period}{1.04\ ns * PWM\ Input\ Clock\ Prescaler}\right]$$

Table 43-1 shows the duty cycle bit resolution versus PWM frequencies at highest PWM clock frequency.

Table 43-1: PWM Frequency and Duty Cycle Resolution

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

At the highest clock frequency, the clock period is 1.04 ns. The PWM resolution becomes coarser by configuring other PWM clock prescaler settings.

Example 43-11: PWM Duty Cycle Selection

```
/* Select either Master Duty cycle or Independent Duty cycle */

PWMCON1bits.MDCS = 0; // PDC1 provides duty cycle value

PWMCON1bits.MDCS = 1; // MDC provides duty cycle value
```

Example 43-12: 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
```

43.6.3 Dead Time Generation

Dead time refers to a programmable period of time (specified by the Dead Time Register (DTRx) or the Alternate Dead Time (ALTDTRx) registers), 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.

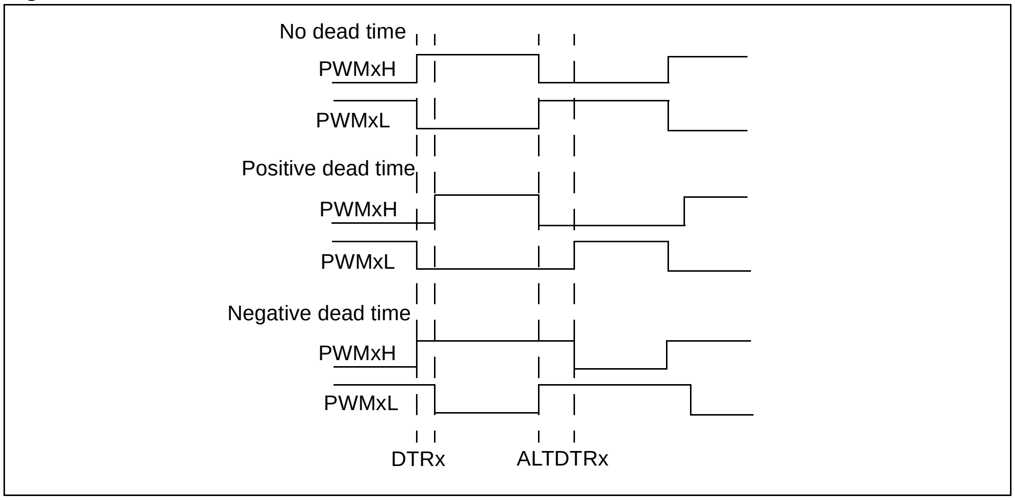
Dead time generation can be provided when any of the PWM I/O pin pairs are operating in the Complementary Output mode. Many power converter circuits require dead time because power transistors cannot switch instantaneously. To prevent current shoot-through, some amount of time must be provided between 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.

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 43-9: Dual Dead Time Waveforms



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

43.6.4 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 starts 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 43-6:

Equation 43-6: Dead Time Calculation

$$DTRx = \frac{REFCLK}{7.37\ MHz} * \frac{Desired\ Dead\ Time}{1.04\ ns * PWM\ Input\ Clock\ Prescaler}$$

There are three Dead Time Control modes:

- Positive Dead Time

  The Positive Dead Time mode describes a period of time when both the 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, the DTRx registers specify the dead time for the PWMxH output, and the ALTDTRx register specifies the dead time for the PWMxL output.

- Negative Dead Time

  The Negative Dead Time mode describes a period of time when both the 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, 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.

- 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 in the PWMCONx register.

43.6.5 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 1.04 ns duty cycle resolution, the dead time resolution is 1.04 ns. At the highest PWM resolution, the maximum dead time value is 17.03 µs.

43.6.6 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).

43.6.7 Dead Time Resolution

At the highest clock rate, the dead time resolution is 1.04 ns under normal operating conditions. However, there are some exceptions:

1. For fault current-limit, or user override events, the highest possible dead time resolution is 8.32 ns (bit 3 in the DTRx and ALTDTRx registers) at maximum CPU speed and prescaler.

   > **Note:** When current-limit or fault override data is set to '0', dead time is not applied, and the "zero" override data is applied immediately.

2. If devices do not implement the High-Resolution PWM option and the PWM clock prescaler resolution is 1.04, 2.08 or 4.16 ns, the highest possible dead time resolution is 8.32 ns.

Example 43-13: PWM Dead Time Control

```
/* Select Dead Time control */
/* Choose one of these options */

PWMCON1bits.DTC = 0;  // Positive Dead Time applied for all modes
PWMCON1bits.DTC = 1;  // Negative Dead Time applied for all modes
```

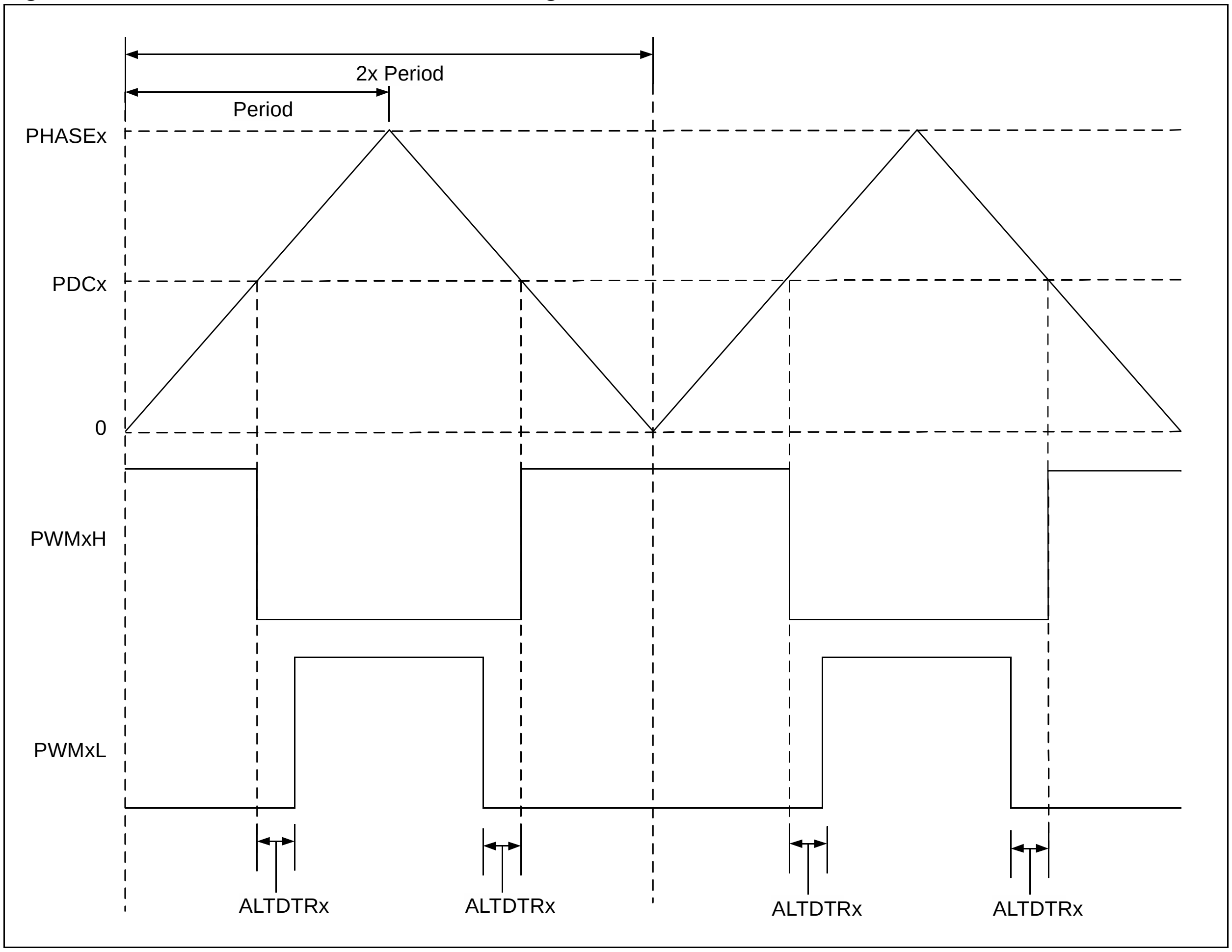
Example 43-14: PWM Dead Time Initialization

```
/* Dead Time value for PWM generator */
/* Refer to Equation 43-5 */

DTR1 = 63;      // Dead Time value is 65.5 ns
ALTDTR1 = 63;   // Alternate Dead time value is 65.5 ns
```

43.6.8 Dead Time Insertion in Center-Aligned Mode

While using the 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 43-10.

Figure 43-10: Dead Time Insertion in Center-Aligned Mode

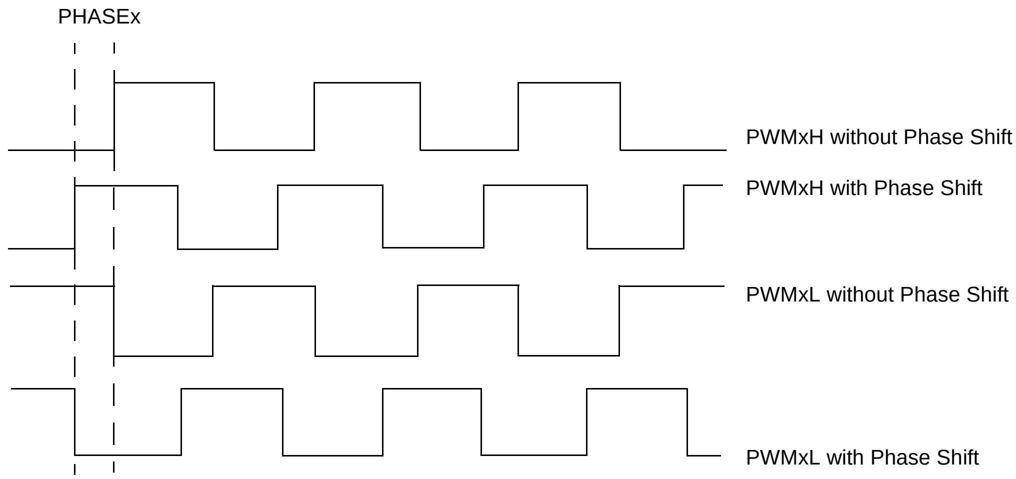


43.6.9 Phase Shift

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 contents of SPHASEx register are used as an initialization value for the STMRx register.

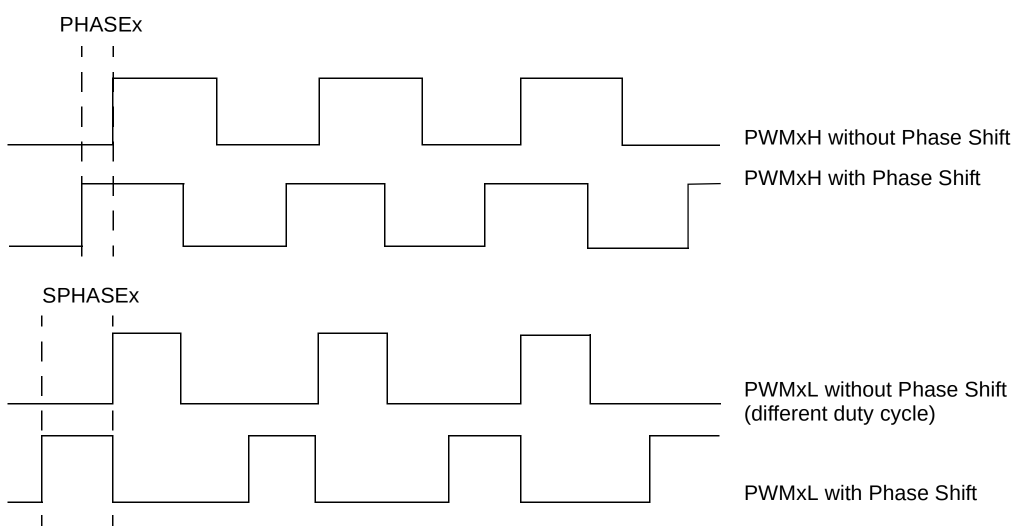
Figure 43-11 and Figure 43-12 provide example waveforms for phase shifting in Complementary mode and Independent Output mode, respectively.

Figure 43-11: Phase Shifting (Complementary Mode)



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

Figure 43-12: Phase Shifting (Independent Output Mode)



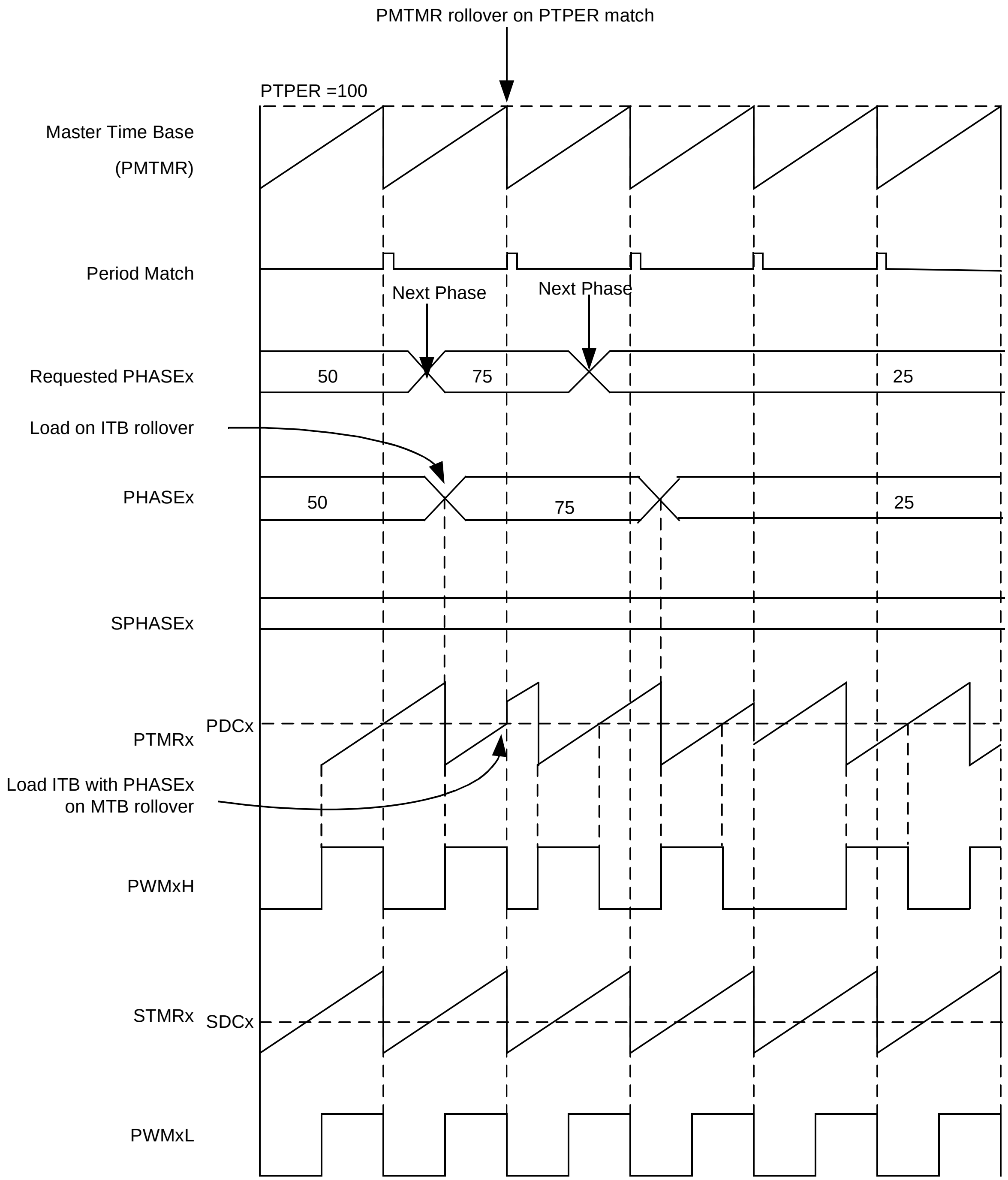
Note: In Independent Output mode, SPHASEx controls the phase shift for PWMxL and PHASEx controls the phase shift of PWMxH.

In addition, there are two shadow registers for the PHASEx and SPHASEx registers that are updated whenever new values are written by the user 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 43-13 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 43-13: Phase Shift Waveforms



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

Example 43-15: 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 of 104 ns

SPHASEx = 100; // Secondary phase shift value of 104 ns
```

The bit resolution of PWM duty cycle, phase, and dead time with respect to different input clock prescaler selections are shown in Table 43-2.

Table 43-2: Duty Cycle, Phase, Dead Time Bit Resolution versus Prescaler Selection

PWM Clock Prescaler	Bit Resolution						
	64 ns	32 ns	16 ns	8 ns	4 ns	2 ns	1 ns
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	—	—	—	—	—	—

43.7 PWM TRIGGER

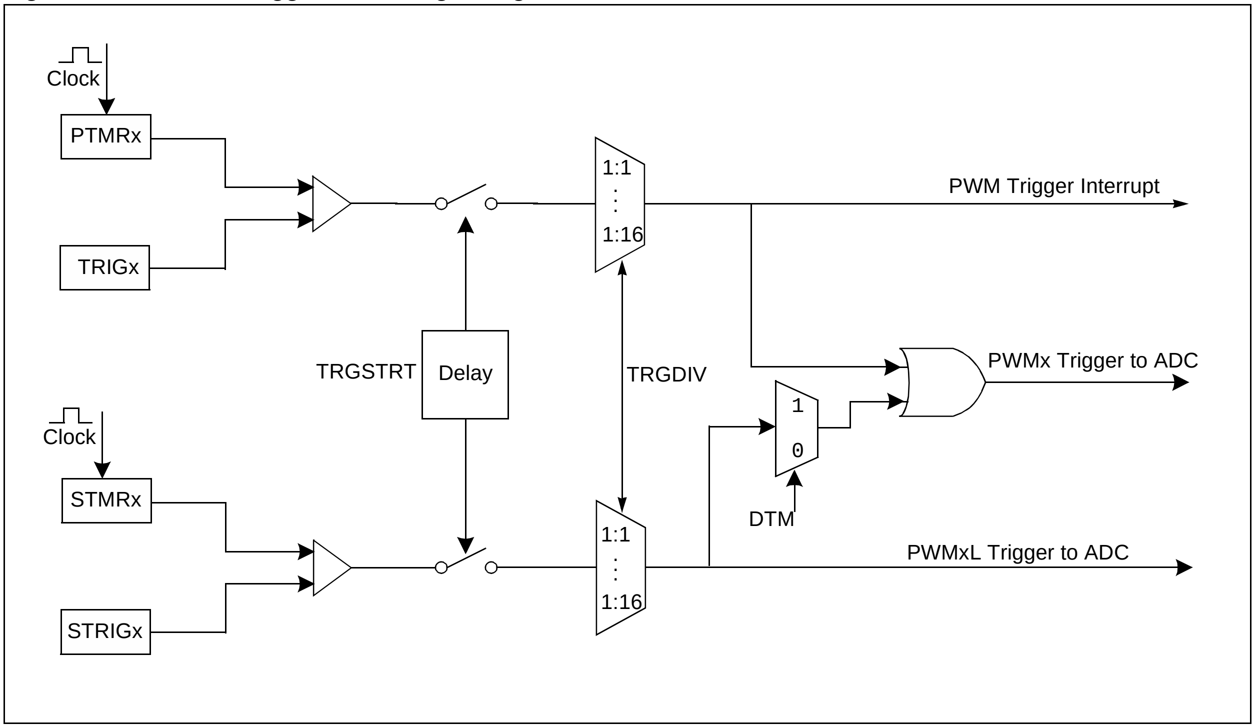
For the ADC module, the TRIGx and STRIGx registers specify the triggering point for the PWMxH and PWMxL outputs, respectively. An ADC trigger signal will be generated when the independent time base counter (PTMRx or STMRx) register value matches with the specified TRIGx or STRIGx register value.

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 Trigger Postscaler Start Enable Select (TRGSTRT<5:0>) bits in the PWM Trigger Control (TRGCONx <5:0>) register that specify how many PWM cycles to wait before generating the first ADC trigger.

Figure 43-14 shows the logic for ADC triggering by the High-Speed PWM module.

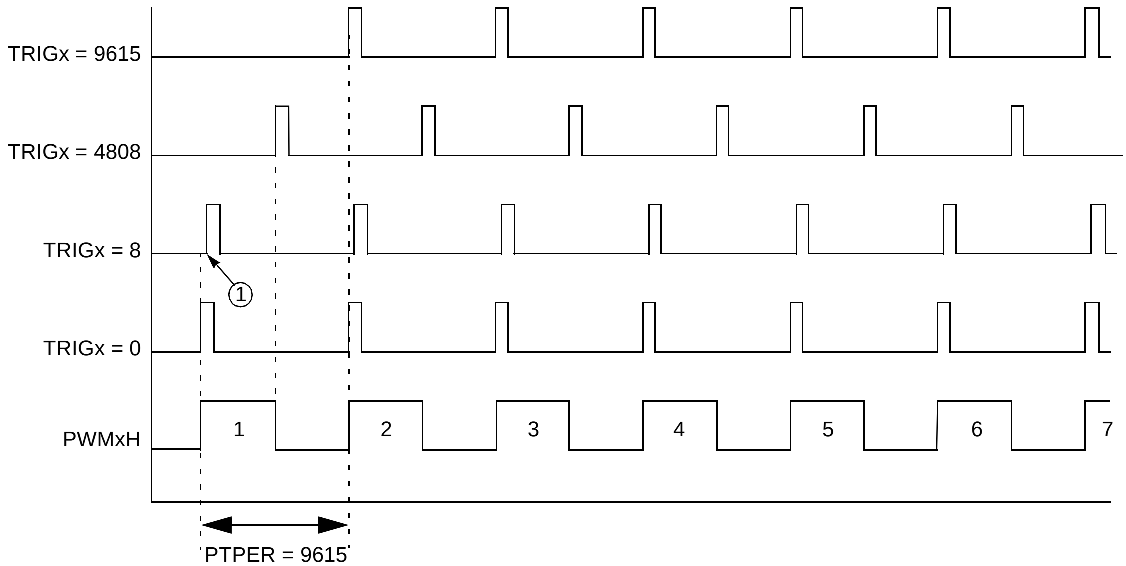
Figure 43-14: PWM Trigger for Analog-to-Digital Conversion



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 43-15 through Figure 43-22.

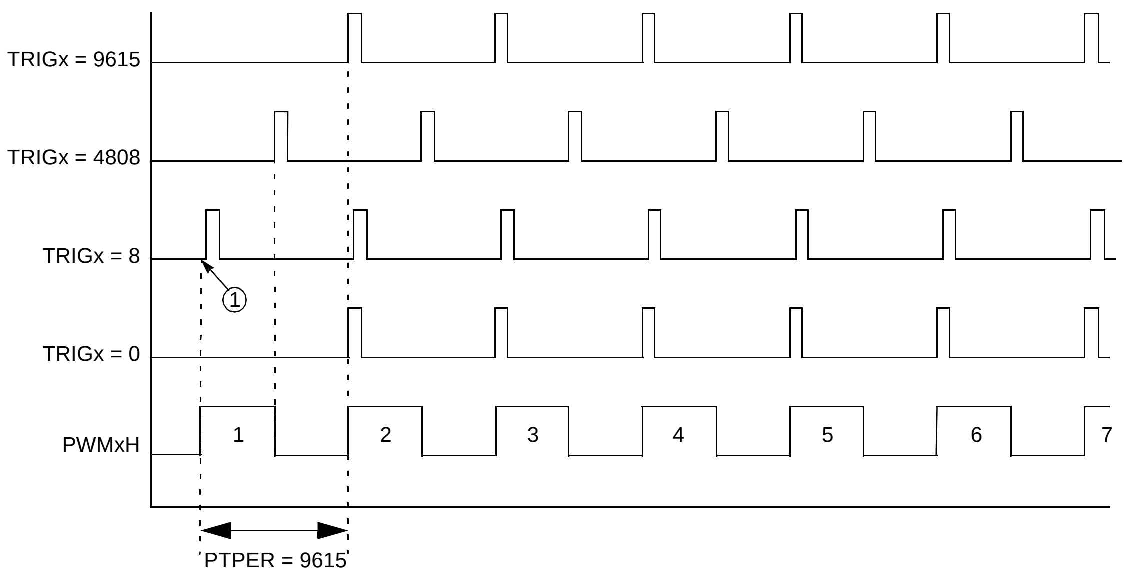
Note:	A trigger can only be generated on the first PWM interval when the TRGDIV<3:0> bits are set to '0'.

Figure 43-15: PWM Trigger Signal in Relation to the PWM Output (TRGDIV = 0, TRGSTRT = 0)



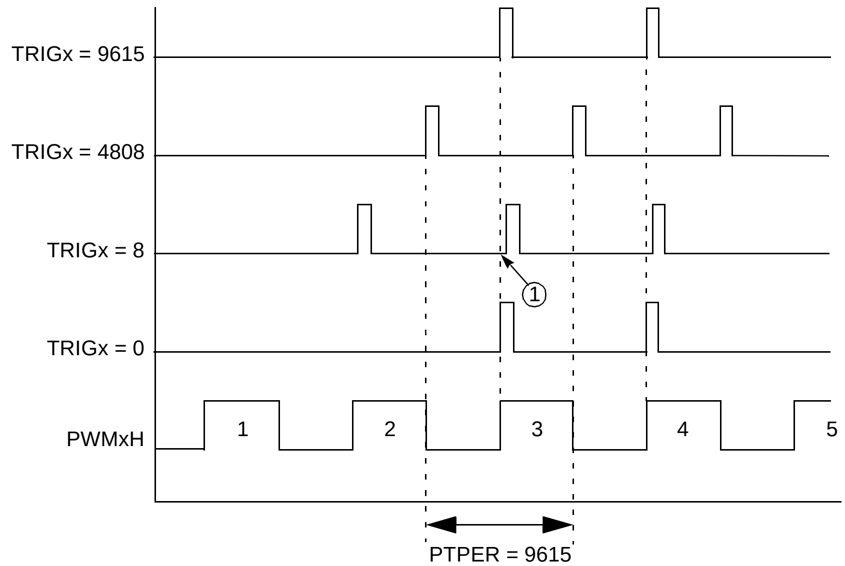
Note 1: This trigger is offset by 8 counts (TRIGx = 8).

Figure 43-16: PWM Trigger Signal in Relation to the PWM Output (TRGDIV = 0, TRGSTRT = 1)



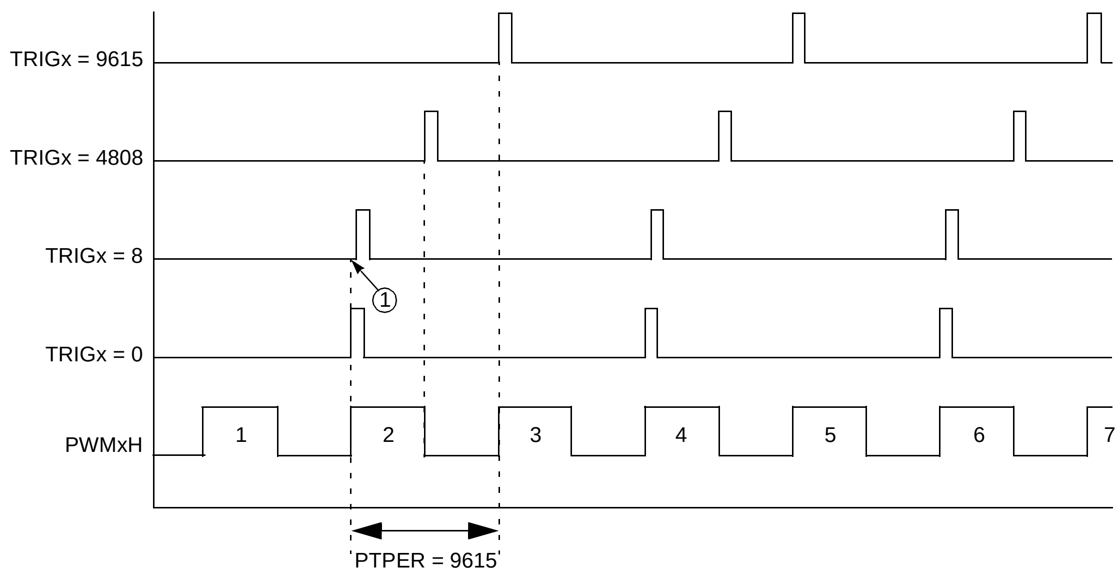
Note 1: This trigger is offset by 8 counts (TRIGx = 8).

Figure 43-17: PWM Trigger Signal in Relation to the PWM Output (TRGDIV = 0, TRGSTRT = 2)



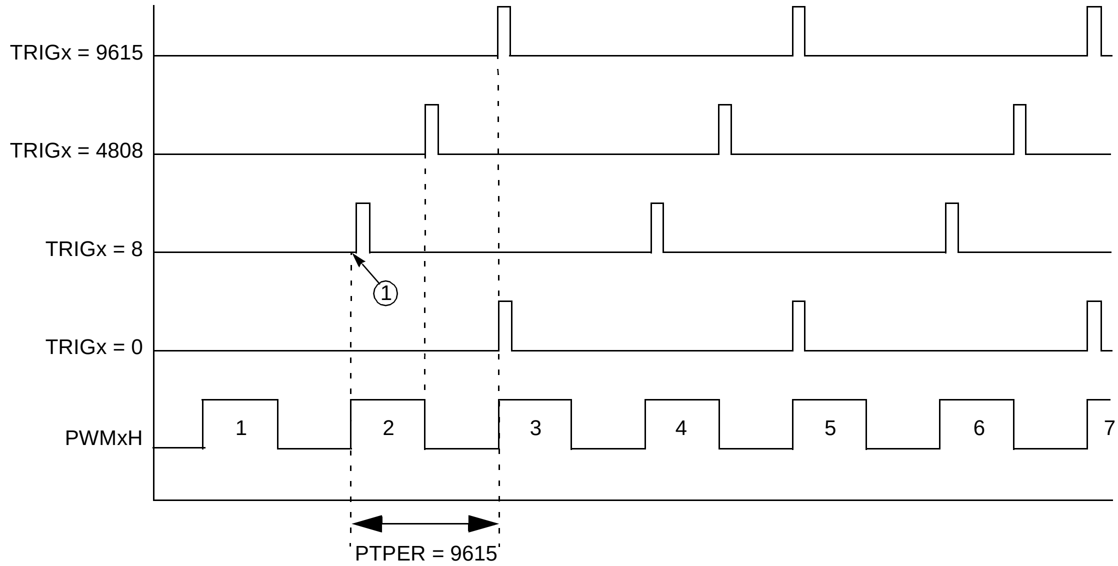
Note 1: This trigger is offset by 8 counts (TRIGx = 8).

Figure 43-18: PWM Trigger Signal in Relation to the PWM Output (TRGDIV = 1, TRGSTRT = 0)



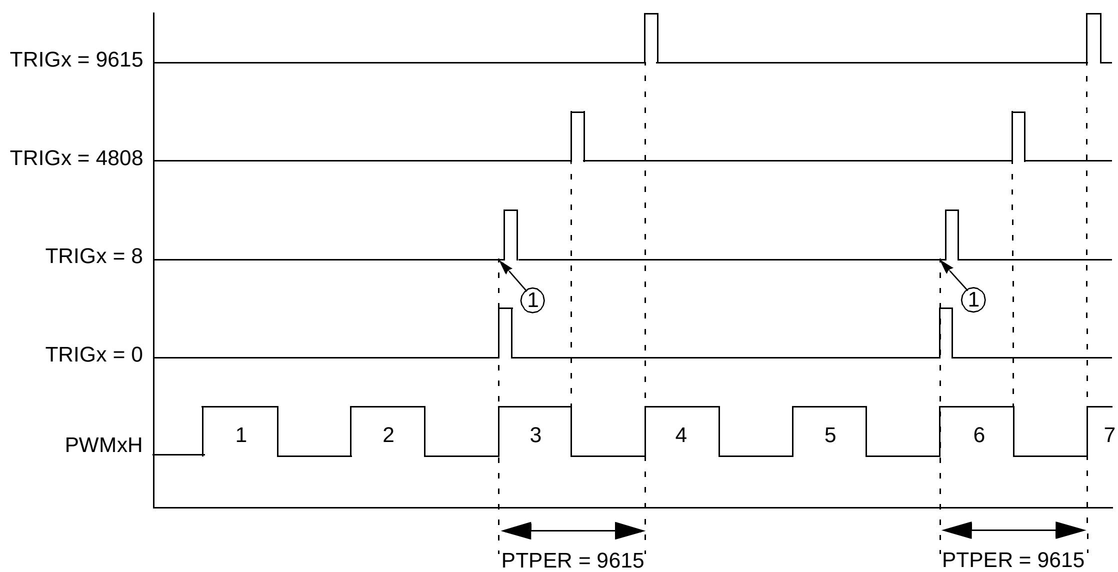
Note 1: This trigger is offset by 8 counts (TRIGx = 8).

Figure 43-19: PWM Trigger Signal in Relation to the PWM Output (TRGDIV = 1, TRGSTRT = 1)



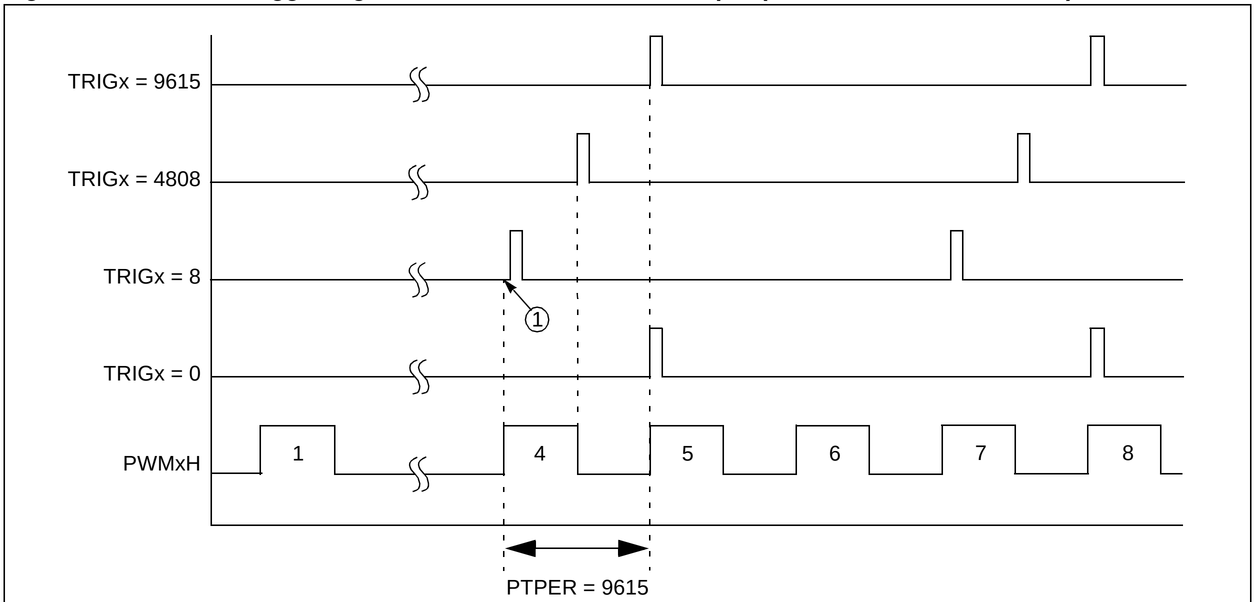
Note 1: This trigger is offset by 8 counts (TRIGx = 8).

Figure 43-20: PWM Trigger Signal in Relation to the PWM Output (TRGDIV = 2, TRGSTRT = 0)



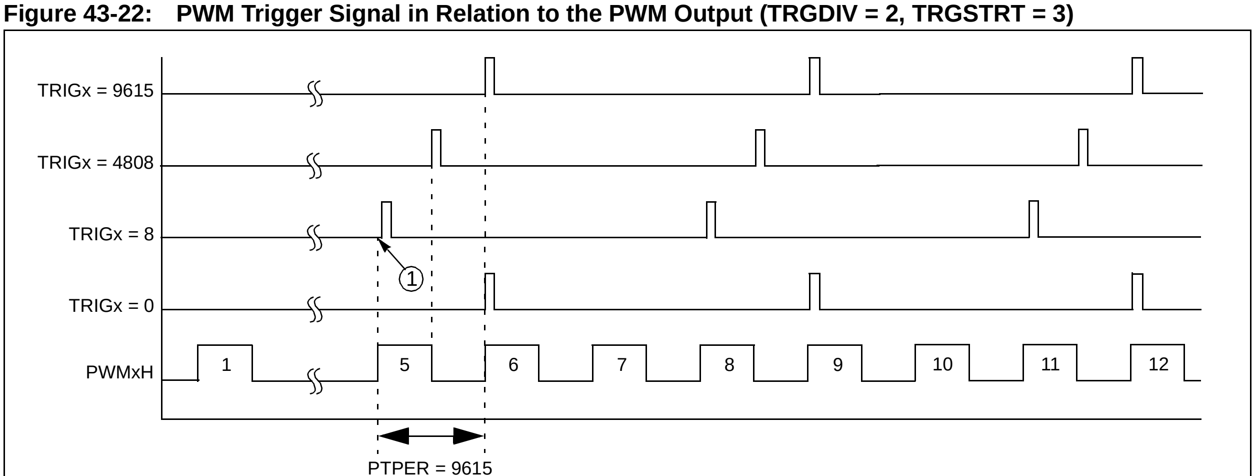
Note 1: This trigger is offset by 8 counts (TRIGx = 8).

Figure 43-21: PWM Trigger Signal in Relation to the PWM Output (TRGDIV = 2, TRGSTRT = 2)



Note 1: This trigger is offset by 8 counts (TRIGx = 8).

Figure 43-22: PWM Trigger Signal in Relation to the PWM Output (TRGDIV = 2, TRGSTRT = 3)



Note 1: This trigger is offset by 8 counts (TRIGx = 8).

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

When the Dual Trigger Mode (DTM<7>) bit in the TRGCONx register is set to ‘1’, the ADC TRIGx output is a Boolean OR of the ADC trigger pulses for the TRIGx and the STRIGx time base comparisons.

The DTM mode of operation allows the user application to take two ADC samples on the same pin within a single PWM cycle.

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 application can ensure that the time it provides is enough to complete two ADC operations within a single PWM cycle.

The trigger pulse is generated regardless of the state of the Trigger Interrupt Enable (TRGIEN) bit in the PWM Control (PWMCONx<10>) register. If the TRGIEN bit in the PWMCONx register is set to ‘1’, an interrupt request is generated.

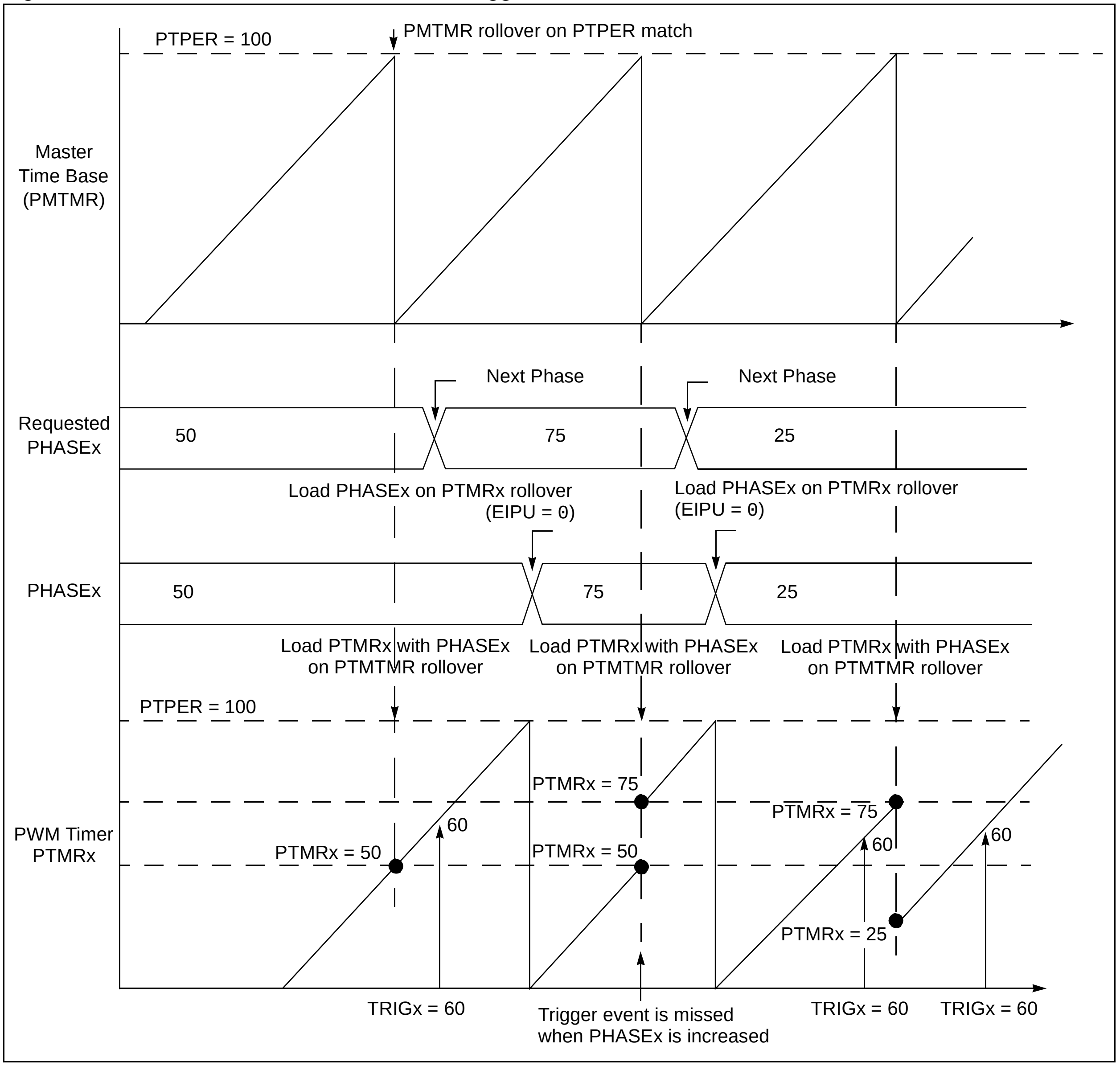
Note:	The secondary trigger comparison does not generate interrupts regardless of the state of the DTM bit.

Example 43-16: Independent PWM ADC Triggering

```
/* Independent PWM ADC triggering */

TRIG1 = 1248;              // Point at which the ADC module is to be triggered by primary PWM
STRIG1 = 2496;             // Point at which the ADC module is to be triggered by secondary PWM

TRGCON1bits.TRGDIV = 0; // Trigger output divider set to trigger ADC on every trigger match event
TRGCON1bits.DTM = 1;       // Primary and Secondary triggers combined to create ADC trigger
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: The TRGSTAT bit is only cleared by clearing the TRGIEN bit. It is not cleared automatically.

Figure 43-23: 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 application must ensure that this does not affect any control loop execution.

43.8 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 four interrupt request signals, the interrupt controller receives an interrupt request signal from the primary time base on special events.

The four 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 application by checking the TRGSTAT, FLTSTAT and CLSTAT bits in the PWMCONx register.

43.8.1 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 Special Event Comparison (SEVTCMP) register 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 Special Event Interrupt Enable (SEIEN) bit in the PWM Time Base Control (PTCON<11>) register. In each PWM generator, the individual time base interrupts generated by the trigger logic are controlled by the Trigger Interrupt Enable (TRGIEN) bits in the PWM Control (PWMCON<10>) register.

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.

43.9 PWM OPERATING MODES

This section describes the following operation modes, which are supported by the High-Speed PWM module.

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

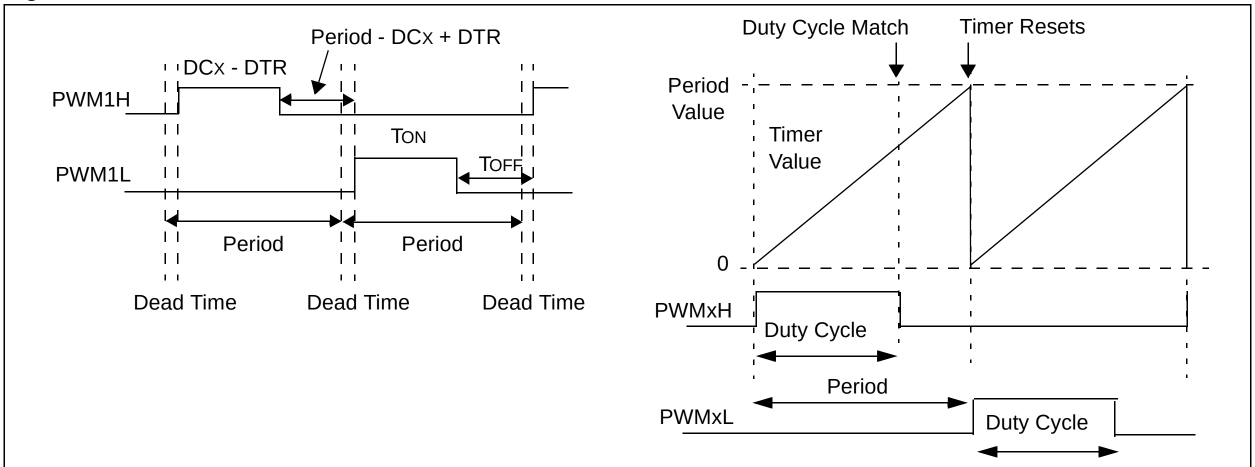
These operating modes can be selected using the PWM I/O Pin Mode (PMOD<1:0>) bits in the PWM I/O Control (IOCONx<11:10>) register.

43.9.1 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 **43.16 "Application Information"**.

Figure 43-24 shows PWM outputs in the Push-Pull PWM mode.

Figure 43-24: Push-Pull PWM Mode

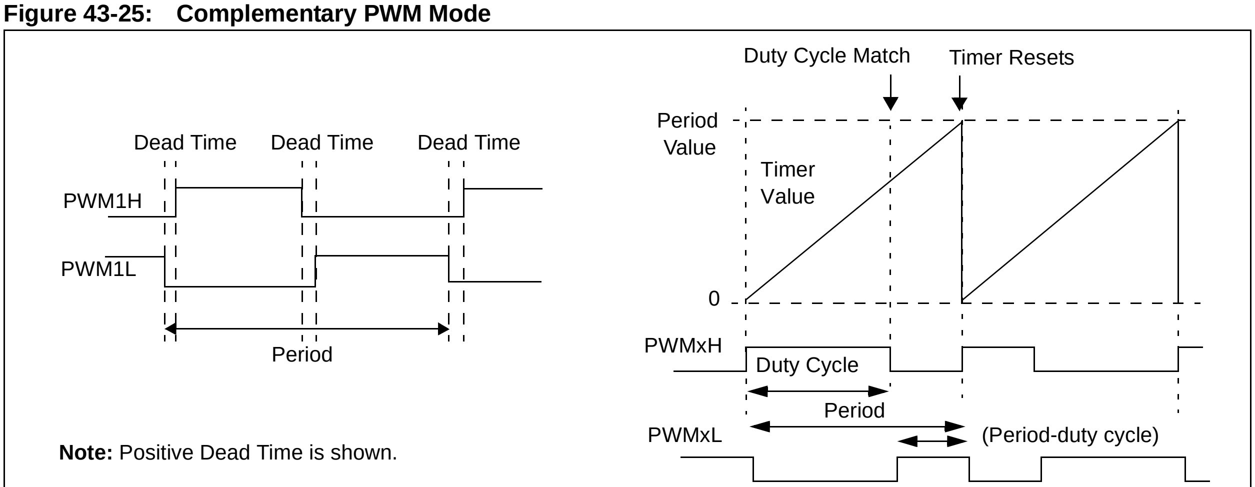


43.9.2 Complementary PWM Mode

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 **43.16 "Application Information"**.

Figure 43-25 shows the PWM outputs when the module operates in Complementary PWM mode.

Figure 43-25: Complementary PWM Mode



Note: Positive Dead Time is shown.

43.9.3 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 application specifies an override value.

Figure 43-26 shows the Redundant PWM Output mode.

Figure 43-26: Redundant PWM Output Mode

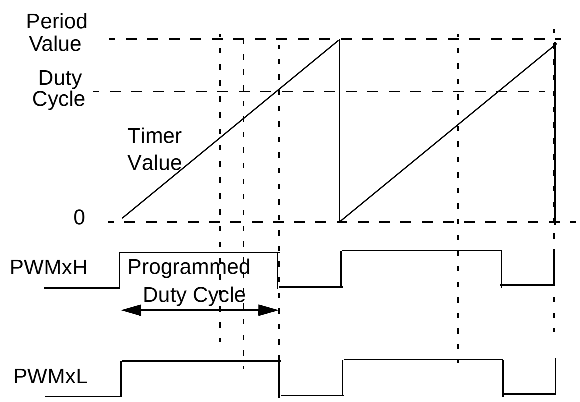


Table 43-3 provides PWM register functionality for the PWM modes.

Table 43-3: Complementary, Push-Pull and Redundant Output Mode Register Functionality

Time Base	Master Time Base		Independent Time Base Shared Time Base	
Function	**PWMxH**	**PWMxL**	**PWMxH**	**PWMxL**
PWM Period	PTPER	PTPER	PHASEx	PHASEx
PWM Duty Cycle	MDC	MDC	MDC/PDCx(1)	MDC/SDCx(1)
PWM Phase Shift	PHASEx	PHASEx	N/A	N/A
ADC Trigger	SEVTCMP/TRIGx(2)	SEVTCMP/TRIGx(2)	TRIGx	TRIGx

Note 1: In the independent time base, the PWMxH duty cycle is controlled by either MDC or PDCx, and the PWMxL duty cycle is controlled by MDC or SDCx.

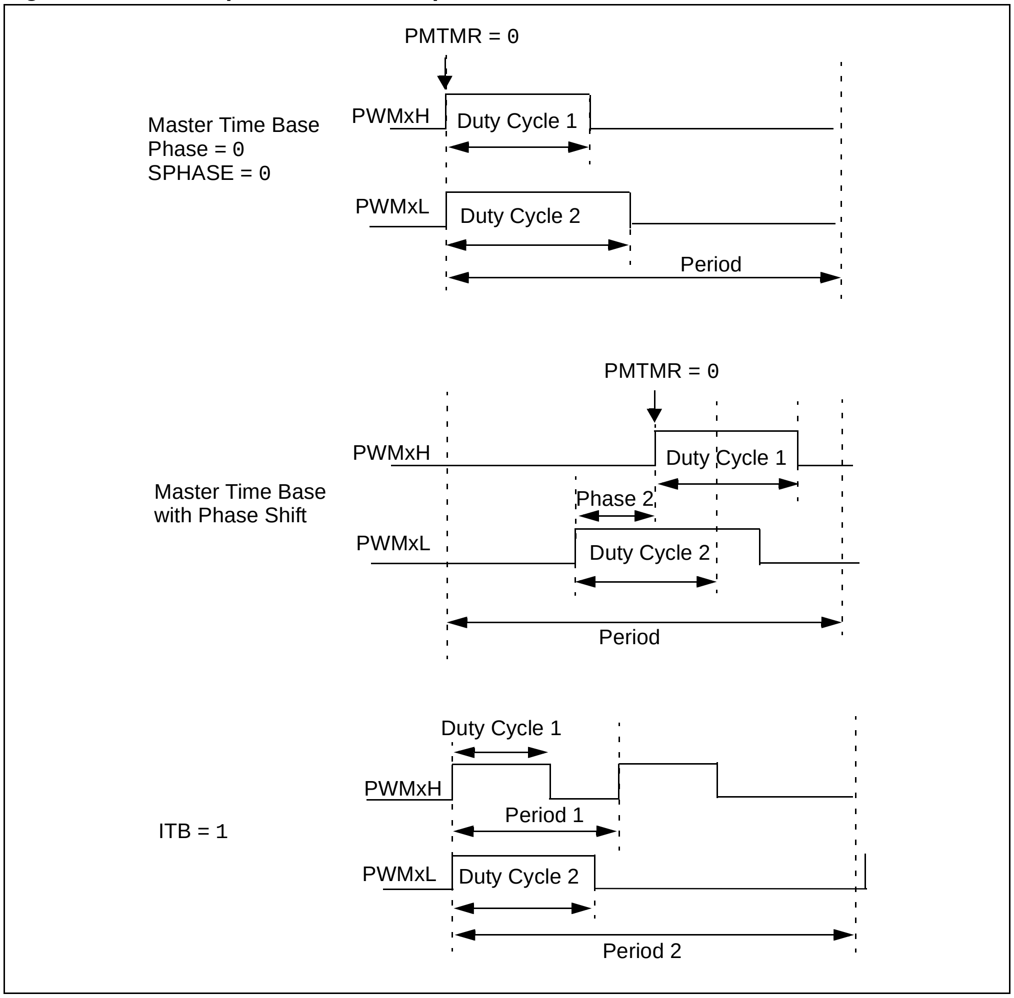
2: In the master time base, the ADC can be triggered by either the SEVTCMP or TRIGx register.

43.9.4 True Independent PWM Output Mode

In True Independent PWM Output mode, the PWM outputs (PWMxH and PWMxL) can have different duty cycles and are phase shifted relative to each other. The PDCx register specifies the duty cycle for the PWMxH output. The SDCx and SPHASEx registers specify the duty cycle and phase shift for the PWMxL output. This mode can be configured by having ITB = `0` and PMOD = 3. If ITB = `1` and PMOD = 3, the two PWM outputs are operating in true independent time period, and duty cycle. This mode of operation is referred to as True Independent mode. The output of the PHASEx and PDCx registers and ITB = `1` control the PWM period and duty cycle of the PWMxH output.

Note:	In True Independent mode, the PWM signals may not be phase related to each other.

Figure 43-27: Independent PWM Output Mode



Example 43-17: PWM Output Pin Mode Selection

```
/* Select PWM I/O pin Mode – Choose one of the following output modes */

IOCON1bits.PMOD = 0; // For Complementary Output mode

IOCON1bits.PMOD = 1; // For Redundant Output mode

IOCON1bits.PMOD = 2; // For Push-Pull Output mode

IOCON1bits.PMOD = 3; // For True Independent Output mode
```


Table 43-4 provides PWM register functionality for the Independent Output mode.

Table 43-4: Independent Output Mode Register Functionality

Time Base	Master Time Base		Independent Time Base Shared Time Base	
Function	**PWMxH**	**PWMxL**	**PWMxH**	**PWMxL**
PWM Period	PTPER	PTPER	PHASEx	SPHASEx(3)
PWM Duty Cycle	MDC	MDC	MDC/PDCx(1)	MDC/SDCx(1)
PWM Phase Shift	PHASEx	SPHASEx	N/A	N/A
ADC Trigger	SEVTCMP/TRIGx(2)	SEVTCMP/TRIGx(2)	TRIGx	TRIGx

Note 1: In the Independent output base, the PWMxH duty cycle is controlled by either MDC or PDCx, and the PWMxL duty cycle is controlled by MDC or SDCx.

2: In the master time base, the ADC can be triggered by either the SEVTCMP or TRIGx register.

3: The SPHASEx register is only used in Independent Output mode.

43.10 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 control bits (FLTSRC<4:0>) in the Fault Current-Limit Control (FCLCONx<7:3>) registers. These bits specify the source for its fault input signal.
- Each PWM generator has the Fault Interrupt Enable (FLTIEN) bit in the PWM Control (PWMCONx<12>) register. This bit enables the generation of fault interrupt requests.
- Each PWM generator has an associated Fault Polarity (FLTPOL) bit in the PWM Fault Current-Limit Control (FCLCONx<2>) register. 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 Independent Fault Mode (IFLTMOD) bit is enabled, the FLTDAT<1:0> (High/Low) bits in the PWM I/O Control (IOCONx<5:4>) register provides data values to be assigned to the PWMxH and PWMxL outputs.
  - In the Fault mode, the FLTDAT<1:0> (High/Low) bits provide the data values to be assigned to the PWMxH and PWMxL outputs.

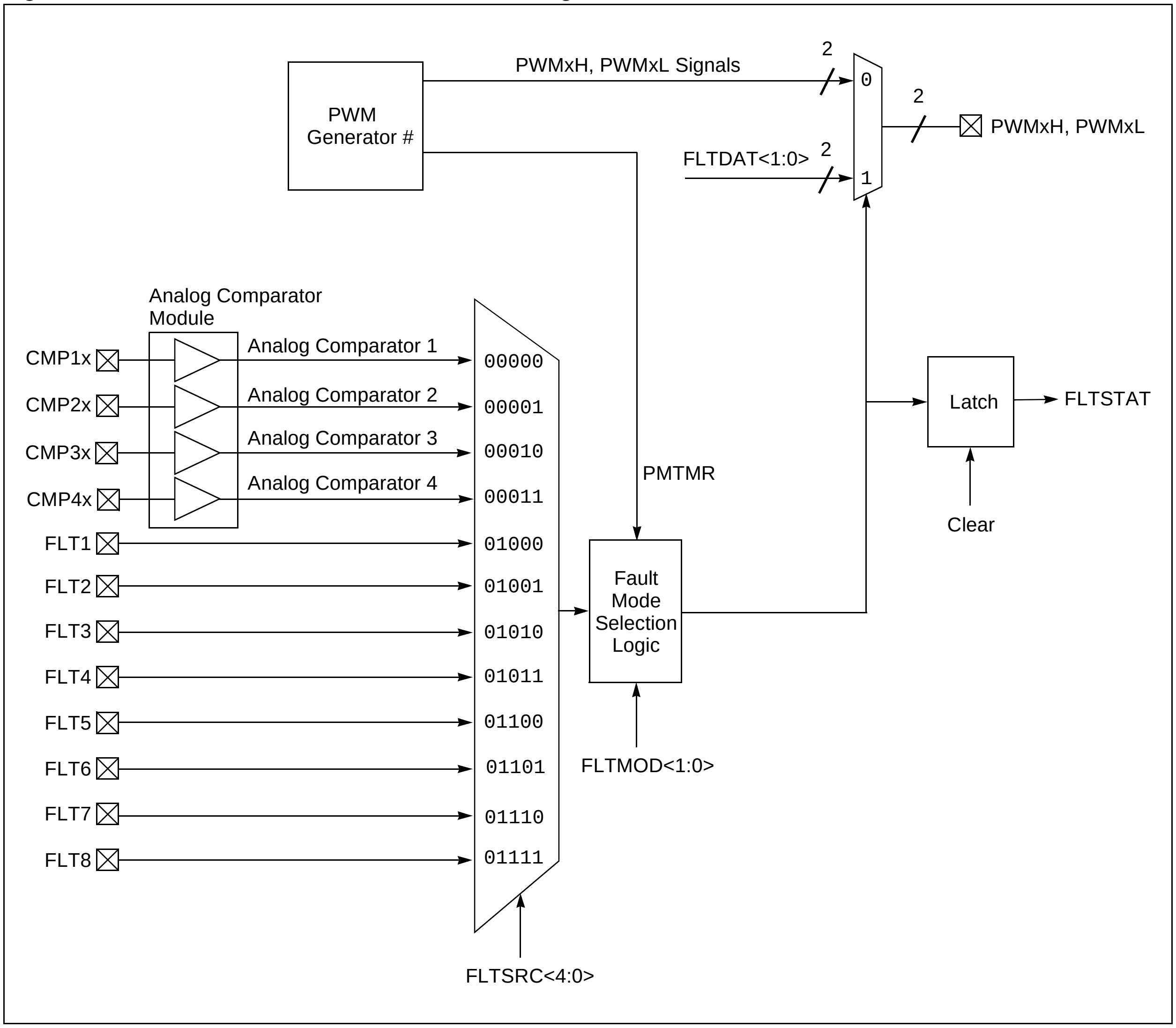
The following list describes major functions of the fault input pin:

- A fault can override the PWM outputs. The Fault Override Data (FLTDAT<1:0>) bits in the IOCONx register 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 in the PWMCONx register controls the fault interrupt signal generation. The user 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 in the FCLCONx registers, 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 application to poll the state of the fault pins in software.

Figure 43-28: PWM Fault Control Module Block Diagram



43.10.1 PWM Fault Generated by the Analog Comparator

To use the comparator output as one of the fault sources, remap the comparator output to a general I/O pin and remap one of the external faults as an input to the same pin. Remapping can be to a general I/O (GPIO) or to a virtual pin.

Virtual pins are identical in functionality to all other RPx pins, with the exception of pinouts. The four virtual pins are internal to the devices and are not connected to a physical device pin.

For example, the output of the Analog Comparator and the PWM fault input can both be configured for RP32. This configuration allows the Analog Comparator to trigger PWM faults without the use of an actual physical pin on the device. Refer to the specific device data sheet, for more information on virtual pins.

Example 43-19 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. The following output and input functions are used:

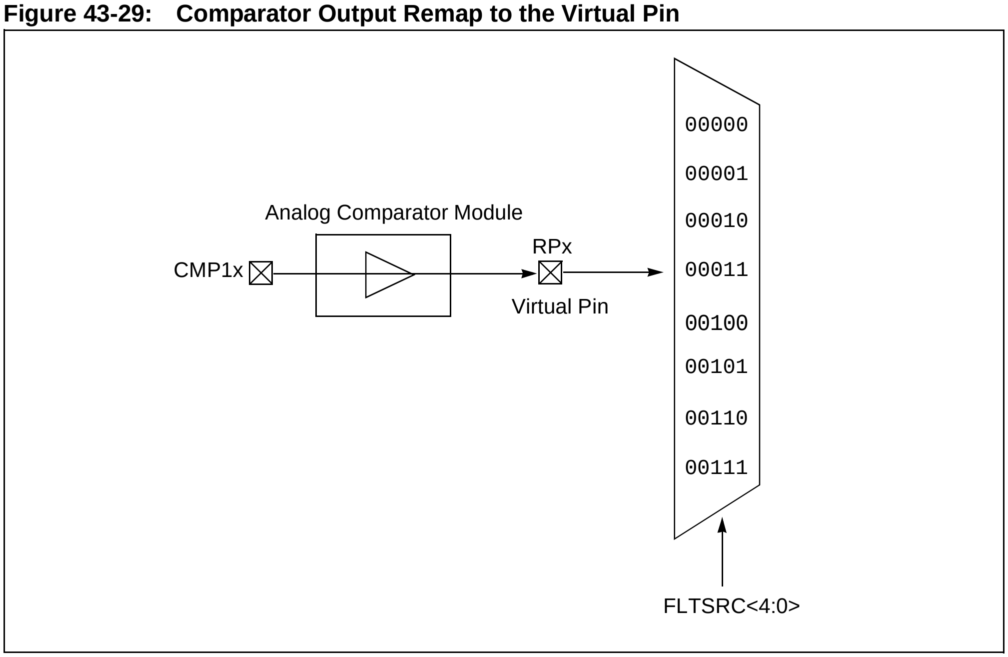
- Output Function: Analog Comparator 1
- Input Function: PWM Fault Pin 1

Example 43-18: Configuring Analog Comparator as a Fault Source to the PWM

```
//*************************************************************
// Unlock Registers
//*************************************************************
__builtin_write_OSCCONL(OSCCON & ~(1<<6));
//************************************************************
// Configure Comparator Output Function
//*************************************************************
//*******************************
// Assign ACMP1 To Pin RP32
//***************************
RPOR16bits.RP32R = 0b100111;
//*************************************************************
// Configure Fault Input Function
//*************************************************************
//***************************
// Assign Fault1 To Pin RP32
//***************************
RPINR29bits.FLT1R=32;
//*************************************************************
// Lock Registers
//*************************************************************
__builtin_write_OSCCONL(OSCCON | (1<<6));
```

Note:	The comparator output can be remapped to a general I/O.

Figure 43-29: Comparator Output Remap to the Virtual Pin



Note:	In RPx, if x = 32, 33, 34, 35 then the comparator output is remapped to a virtual pin, which is unavailable to the user.

43.10.2 Fault Interrupts

The FLTIENx bits in the PWMCONx registers 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 Fault Interrupt Status (FLTSTAT) bit in the PWM Control (PWMCONx<15>) registers. 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.

43.10.2.1 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 Fault Interrupt Status (FLTSTAT) bit in the PWM Control (PWMCONx<15>) registers 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 Fault Interrupt Status Enable FLTIEN (PWMCONx<12>) bit.
- **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 using the Fault Mode (FLTMOD) control bits in the PWM Fault Current-Limit Control (FCLCONx<1:0>) register.

43.10.3 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, 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 **43.12.4 “Fault and Current-Limit Override Logic Issues with Dead Time Logic”**, for more information on data sensitivity and behavior in response to current-limit or Fault events.

43.10.4 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 in the PWMCONx register.
   c) Enable the PWM fault interrupt by setting FLTMOD<1:0> = `0b00` in the FCLCONx register.
3. Disable PWM faults by setting FLTMOD<1:0> = `0b00` in the FCLCONx register.
4. Enable the latched PWM Fault mode by setting FLTMOD<1:0> = `0b00` in the FCLCONx register.

43.10.5 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.

43.10.6 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.

43.10.7 PWM Current-Limit Pins

The key functions of the PWM current-limit pins are as follows:

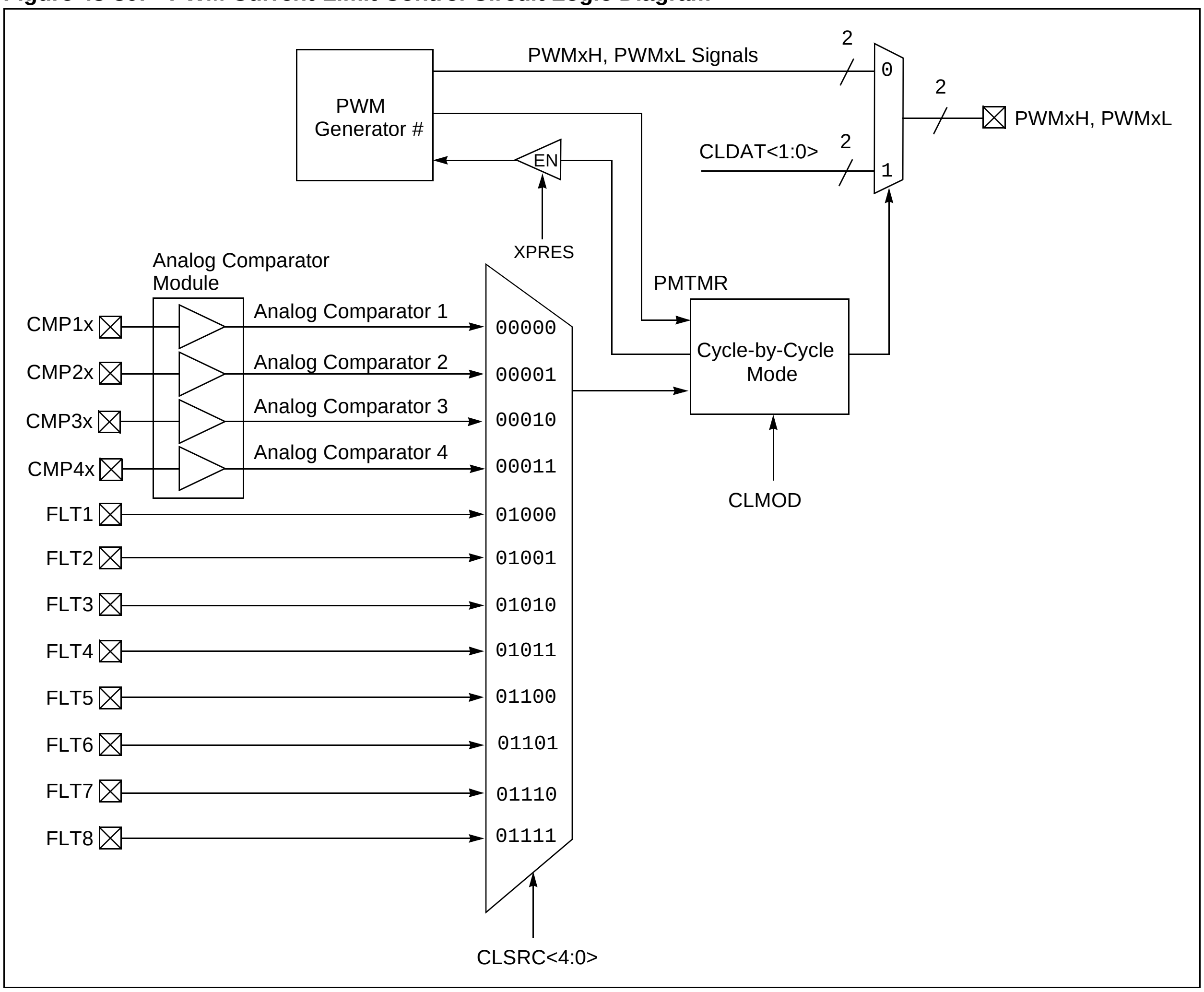
- Each PWM generator can select its own current-limit input source up to eight fault and current-limit pins. To configure the analog comparator as one of the current-limit sources, refer to **43.10.1 "PWM Fault Generated by the Analog Comparator"**.
- Each PWM generator has (CLSRC<4:0>) control bits in the Fault Current-Limit Control (FCLCONx<14:10>) registers. These bits specify the source for its fault input signal.
- Each PWM generator has a corresponding Current-Limit Interrupt Enable (CLIEN) bit in the PWM Control (PWMCONx<11>) register. This bit enables the generation of current-limit interrupt requests.
- Each PWM generator has an associated Current-Limit Polarity (CLPOL) bit in the PWM Fault Current-Limit Control (FCLCONx<9>) register.
- 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 Current-Limit Override Data (CLDAT<1:0>) bits in the PWM I/O Control (IOCONx<3:2>) register 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 Current-Limit Interrupt Enable (CLIEN) bit in the PWM Control (PWMCONx<11>) registers controls the current-limit interrupt signal generation. The user application can specify interrupt generation even if the Current-Limit Mode Enable (CLMOD) bit in the PWM Fault Current-Limit Control (FCLCONx<8>) register 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 Current-Limit Mode Enable (CLMOD) bit for the PWM Generator is '0'
  - The External PWM Reset Control (XPRES) bit in the PWMCONx register 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 43-30: PWM Current-Limit Control Circuit Logic Diagram



43.10.8 Current-Limit Interrupts

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

43.10.9 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 bits in the PWM I/O Control (IOCONx<3:2>) registers, 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 bits in the PWM I/O Control (IOCONx<5:4>) registers.

43.10.10 PWM Fault and Current-Limit Trigger Outputs to ADC

The current-limit and fault source selection (CLSRC<4:0> and FLTSRC<4:0>) bits in the PWM Fault Current-Limit control (FCLCONx<14:10> and FCLCONx<7:3>) registers control the fault selection to each PWM generator module. The control multiplexers select the desired fault and current-limit signals for their respective modules. The selected fault and current-limit signals which are also available to the ADC module as trigger signals, initiate ADC sampling and conversion operations.

Example 43-19: 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 = 0;      // Current-limit input source is Analog Comparator 1
FCLCON1bits.FLTSRC = 3;     // Fault input source is Analog Comparator 4

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
```

43.11 SPECIAL FEATURES

The following special features are available in the High-Speed PWM module:

- Leading-Edge Blanking (LEB)
- Individual time base capture
- PWM pin swapping
- PWM output pin control and override
- PWM immediate update

43.11.1 Leading-Edge Blanking (LEB)

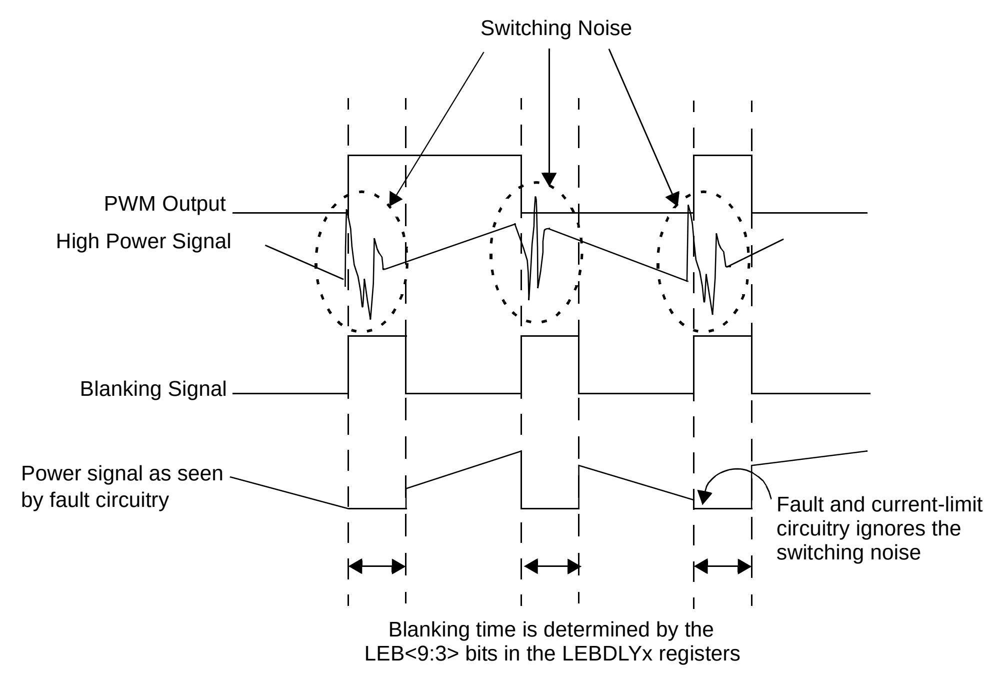
Each PWM generator supports LEB of the current-limit and fault inputs through the LEB (LEBCONx<9:3>) bits and the PHR (LEBCONx<15>), PHF (LEBCONx<14>), PLR (LEBCONx<13>), PLF (LEBCONx<12>, FLTLEBEN (LEBCONx<11>), and CLLEBEN (LEBCONx<10>) bits in the Leading-Edge Blanking Control registers. The purpose 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. The LEB bits support the blanking (ignoring) of the current-limit and fault inputs for a period of 0 to 1023 ns in 8.4 ns increments following any specified rising or falling edge of the PWMxH and PWMxL 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 43-31: Leading-Edge Blanking



43.11.2 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 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 application can calculate the slope of the current rise in the inductor. The secondary independent time base does not have an associated Capture register.

43.11.3 PWM Pin Swapping

The SWAP bit in the IOCONx register, if set to ‘1’, enables the user 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 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 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 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., 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).

43.11.3.1 EXAMPLE 1: PIN SWAPPING WITH SMPS POWER CONTROL

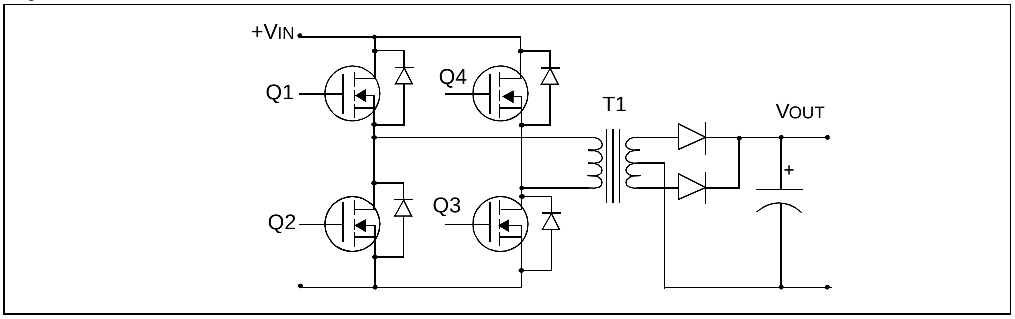
The SMPS Power Control example describes dynamic swapping. In power conversion applications, the transistor modulation technique can be changed between the full-bridge Zero Voltage Transition (ZVT) and standard full-bridge “on-the-fly” transition to meet different load and efficiency requirements. As shown in Figure 43-32, 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 43-32: SMPS Power Control



43.11.3.2 EXAMPLE 2: PIN SWAPPING WITH MOTOR CONTROL

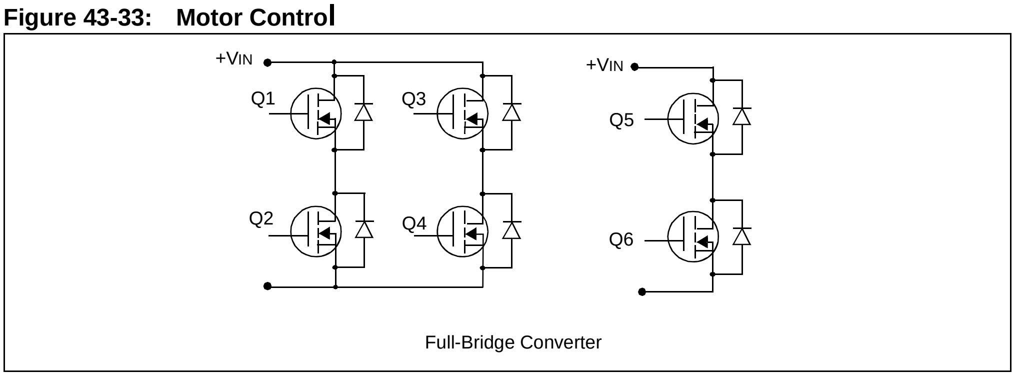
The Motor Control example describes static swapping. Consider a generic motor control system, which is capable of driving two different types of motors, such as DC motors and three-phase AC induction motors.

Brushed DC motors typically use a full-bridge transistor configuration, as shown in Figure 43-33. 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 43-33: 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 43-20: PWM Pin Swapping

```
/* PWM Pin Swapping feature */

IOCON1bits.SWAP = 1;    // PWMxH output signal is connected to the PWMxL pin
                        // and vice versa
```

43.12 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 43-21: PWM Output Pin Assignment

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

Example 43-22: PWM Output Pins State Selection

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

Example 43-23: Enabling the High-Speed PWM Module

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

PTCONbits.PTEN = 1;
```

43.12.1 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 PENH (IOCONx<15>) and PENL (IOCONx<14>) bits are set, the High-Speed PWM module controls the PWMx output pins. The PWM Output Override bits allow the user application to manually drive the PWM I/O pins to specified logic states, independent of the duty cycle comparison units.

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

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

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

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

43.12.3 Override Synchronization

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

43.12.4 Fault and Current-Limit Override Logic Issues with Dead Time Logic

In the event of a Fault and Current-Limit condition, the data in the FLTDAT<1:0> (IOCONx<5:4>) bits or CLDAT<1:0> (IOCONx<3:2>) bits determine the state of the PWM I/O pins.

If any of the FLTDAT<1:0> (IOCONx<5:4>) bits or CLDAT<1:0> (IOCONx<3:2>) bits are '`0`', the PWMxH and/or PWMxL outputs are driven low immediately, bypassing the dead time logic. This behavior turns off the PWM outputs immediately without any additional delays. This may impact many power conversion 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> (IOCONx<5:4>) bits or CLDAT<1:0> (IOCONx<3:2>) bits are '`1`', the PWMxH and/or PWMxL outputs pass 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.

43.12.5 Asserting Outputs Via Current-Limit

In response to a Current-Limit event, the CLDAT (IOCONx<3:2>) bits 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.

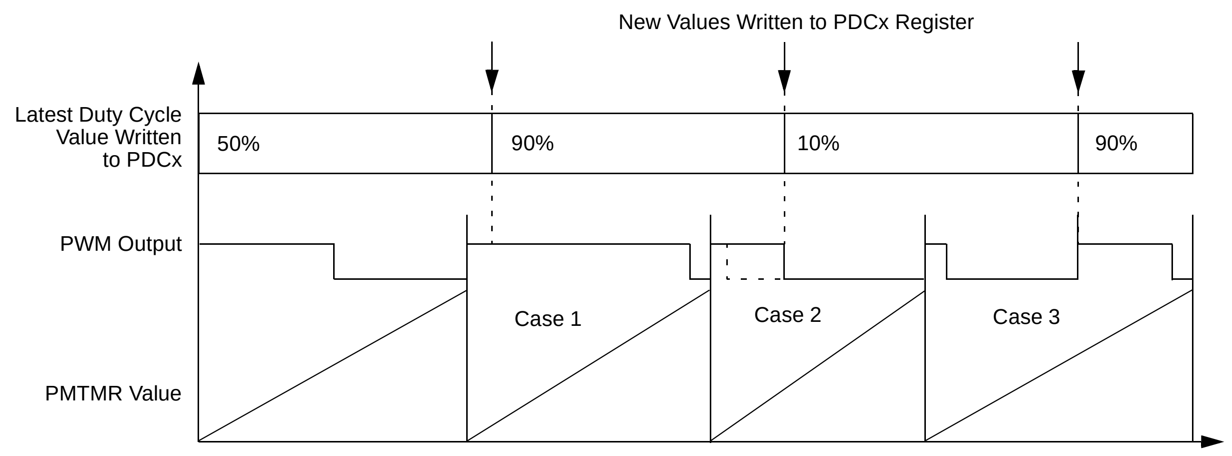
43.13 IMMEDIATE UPDATE OF PWM DUTY CYCLE

The high performance PWM control-loop application requires a maximum duty cycle update rate. Setting the Immediate Update Enable (IUE) bit in the PWM Control (PWMCONx<0>) registers 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 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 43-34: Duty Cycle Update Times When Immediate Updates are Enabled (IUE = 1)



Example 43-24: 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
```

43.14 POWER-SAVING MODES

This section discusses the operation of the High-Speed PWM module in Sleep mode and Idle mode.

43.14.1 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.

43.14.2 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 application disable the PWM outputs prior to entering Idle mode. If the PWM module is controlling a power conversion application, the action of putting the device into Idle mode 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.

43.15 EXTERNAL CONTROL OF INDIVIDUAL TIME BASE(S)

External signals can reset the primary dedicated time bases, if the External PWM Reset Control (XPRES) bit in the (PWMCONx<1>) register is set. This mode of operation is called Current Reset PWM mode. If the user application sets the ITB bit, a PWM generator operates in Independent Time Base mode. If the user 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 Current-Limit Control Signal Source Select (CLSRC<4:0>) bits in the PWM Fault Current-Limit Control (FCLCONx<14:10>) register causes the independent time base to reset. The active edge of the selected current-limit signal is specified by the CLPOL bit in the FCLCONx register.

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.

43.16 APPLICATION INFORMATION

Typical applications that use different PWM operating modes and features are:

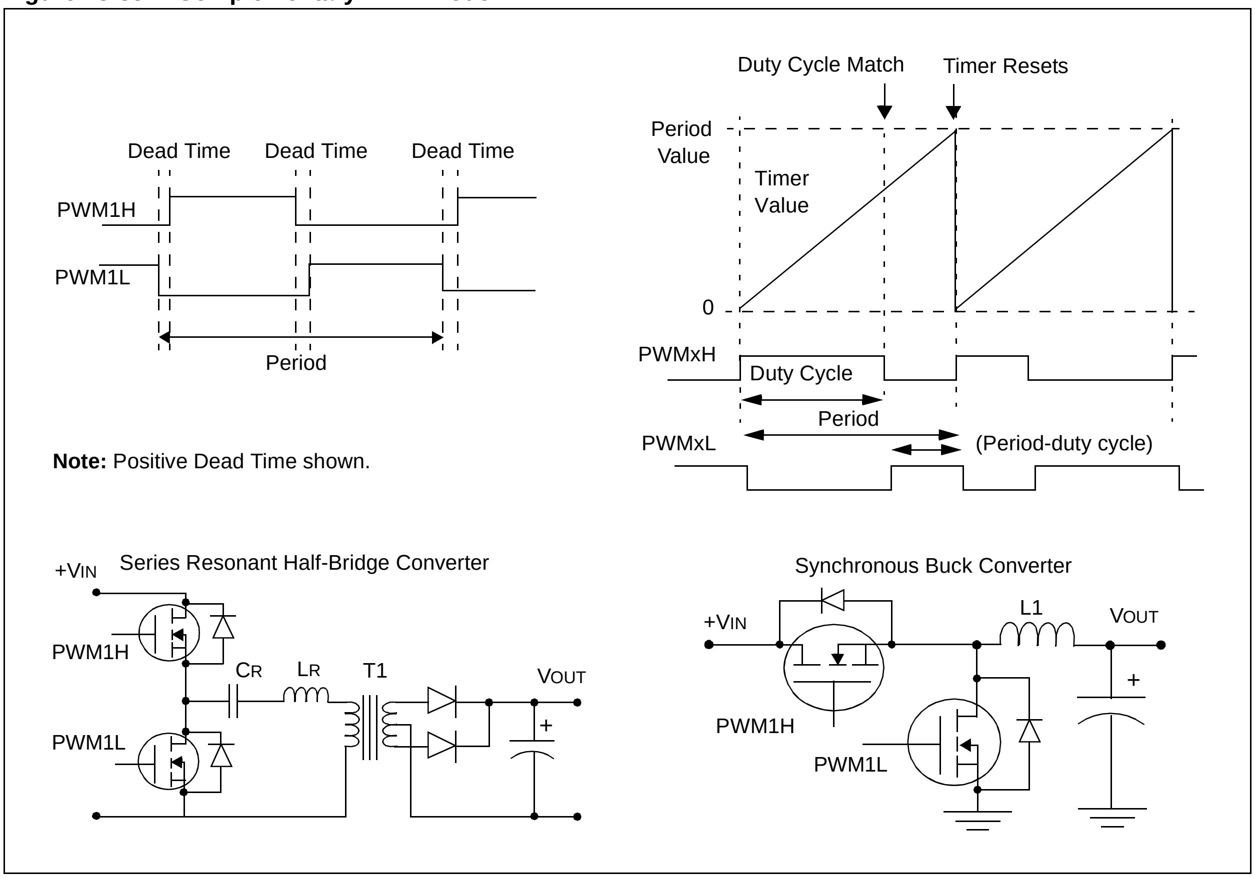
- Complementary Output Mode
- Push-Pull Output Mode
- Multi-Phase PWM
- Variable Phase PWM
- Current Reset PWM
- Constant Off-Time PWM
- Current-Limit PWM

Each application is described in the following sections.

43.16.1 Complementary Output Mode

The Complementary PWM mode shown in Figure 43-35 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).

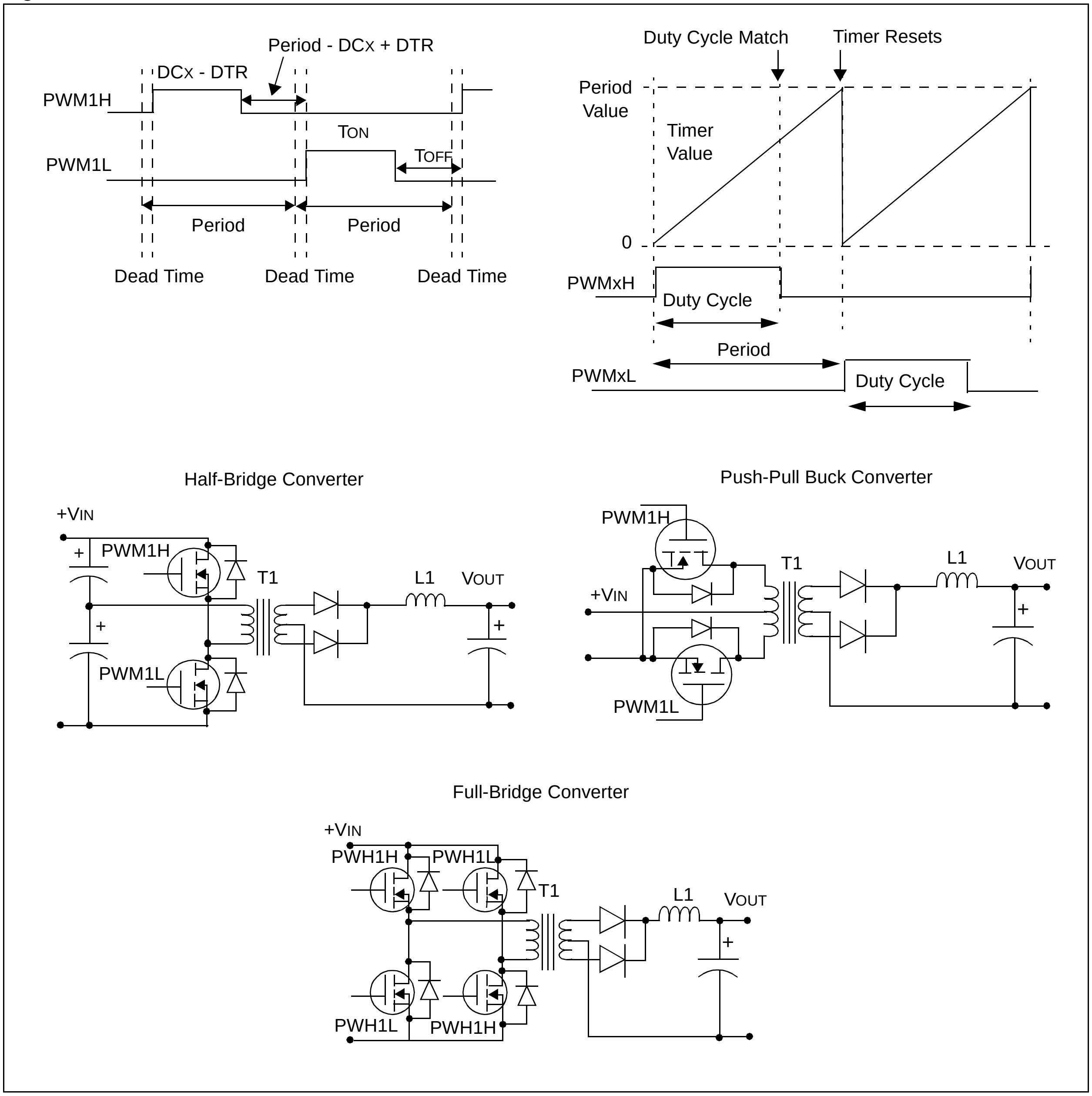
Figure 43-35: Complementary PWM Mode



43.16.2 Push-Pull Output Mode

The Push-Pull PWM mode, shown in Figure 43-36, 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, thus yielding a net DC bias of zero.

Figure 43-36: Push-Pull PWM Mode



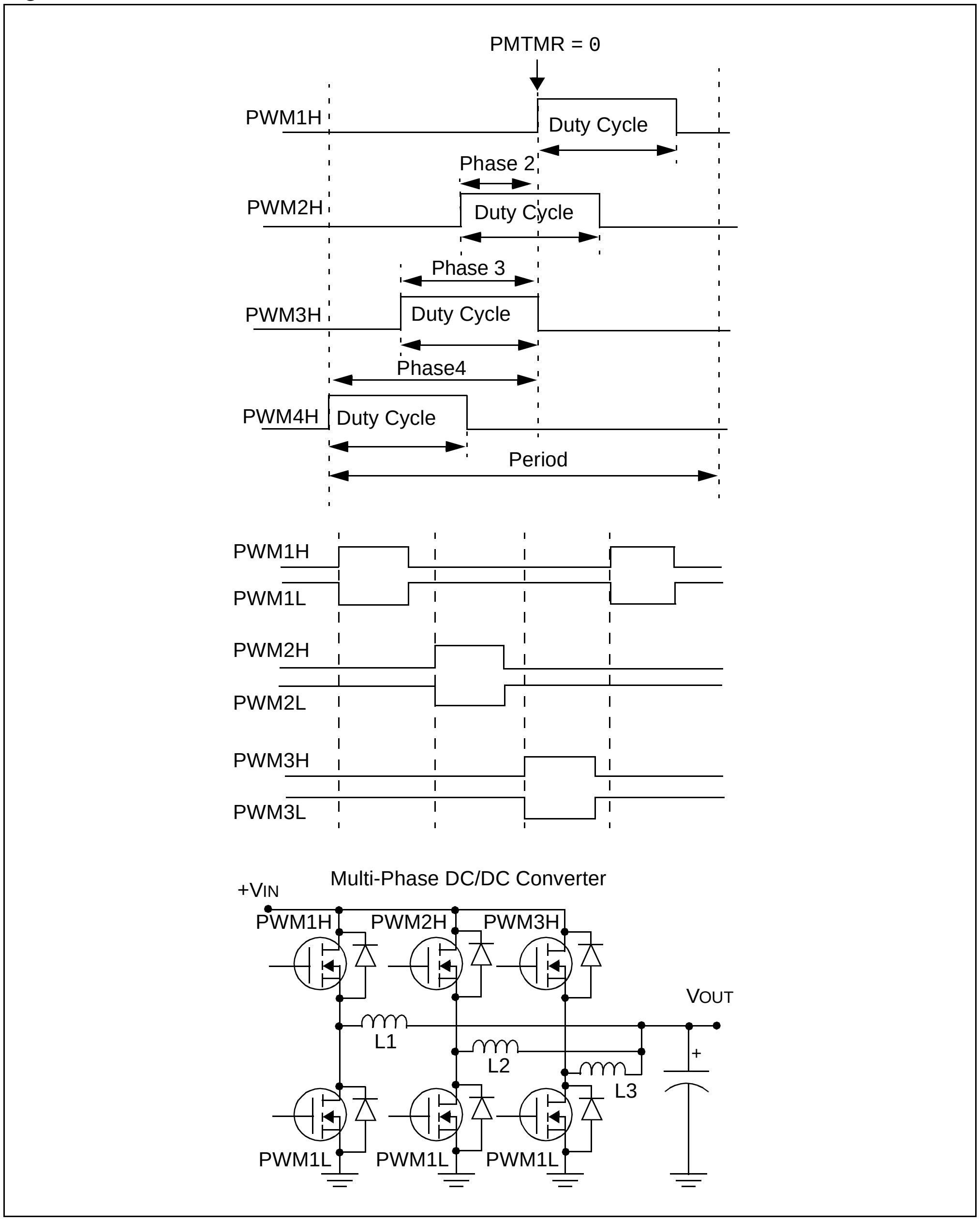
43.16.3 Multi-Phase PWM

The Multi-Phase PWM, shown in Figure 43-37, 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 43-38, 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 43-37: Multi-Phase PWM



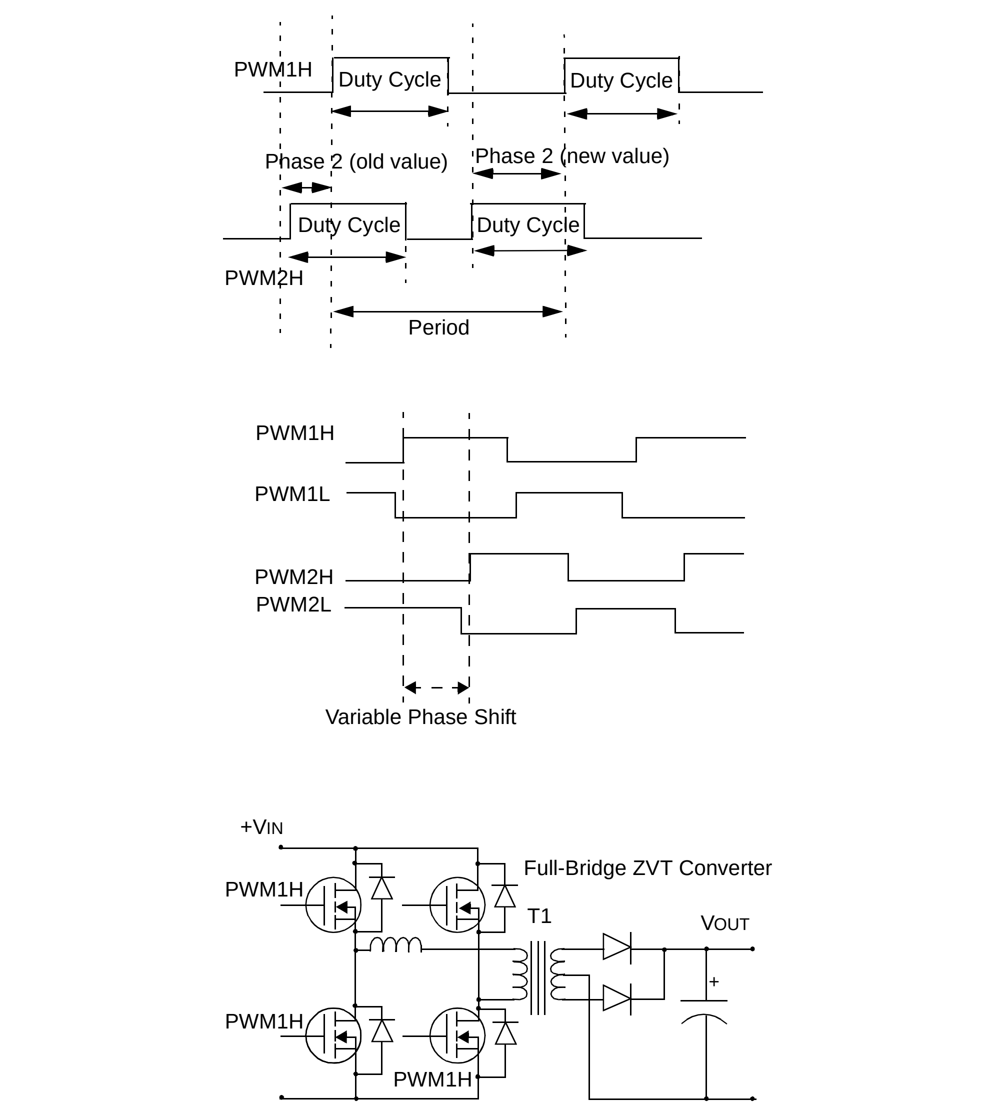
43.16.4 Variable Phase PWM

The Variable Phase PWM, shown in Figure 43-38, 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.

Variable phase PWM is used in newer power conversion 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 ($V * I * T_{SW} * F_{PWM}$) 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 43-38: Variable Phase PWM



43.16.5 Current Reset PWM

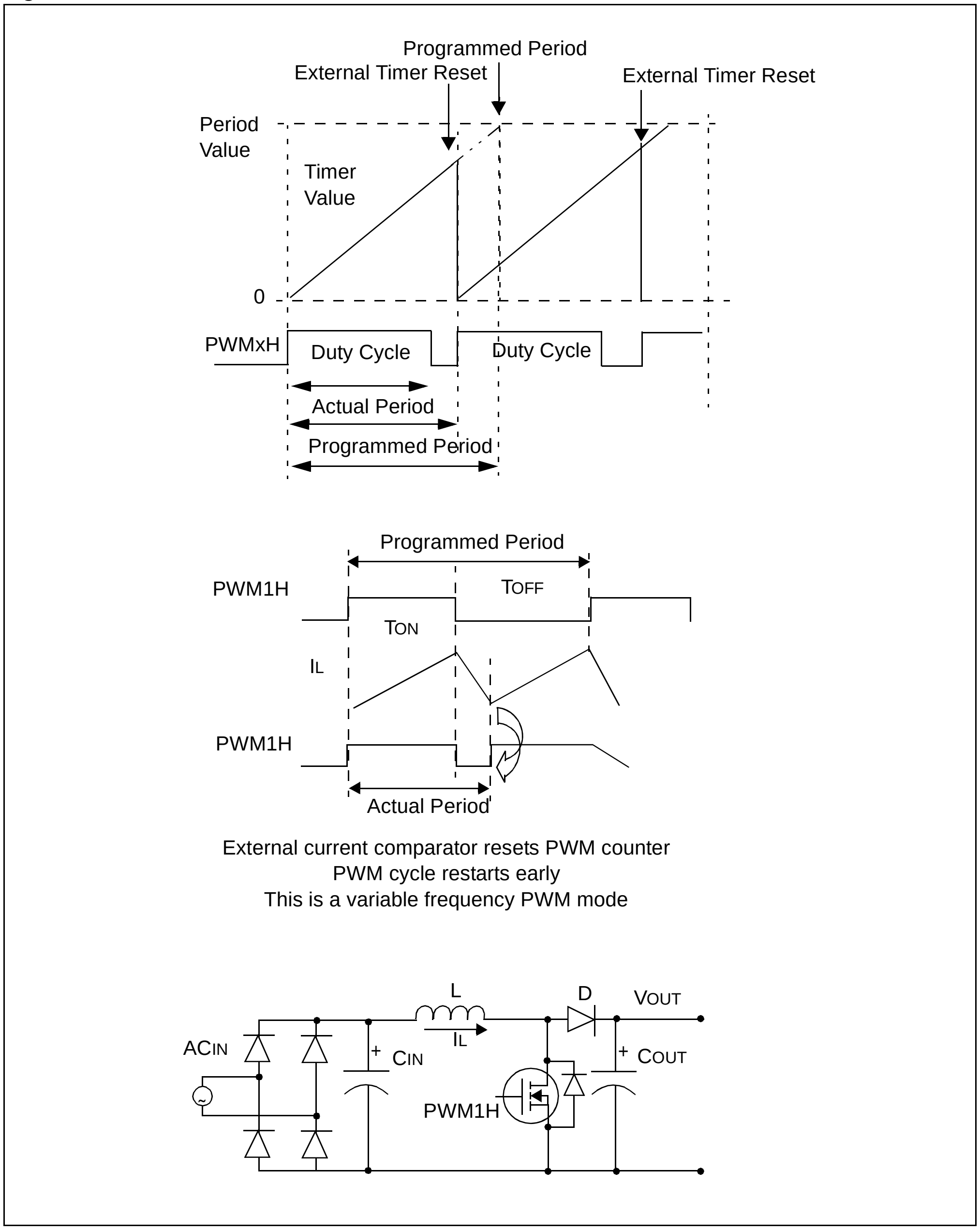
The Current Reset PWM, shown in Figure 43-39, is a variable frequency mode where the actual PWM period is less than or equal to the specified period value. The independent time base is reset externally some time after the PWM signal has been deasserted. This is called Constant PWM On-Time mode. To operate 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 the 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 duty cycle.

In Current Reset mode, the PWM frequency varies with the load current. This is different than most PWM modes because the user 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 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 sometimes called Constant On-Time PWM output.

This should not be confused with 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.

Figure 43-39: Current Reset PWM

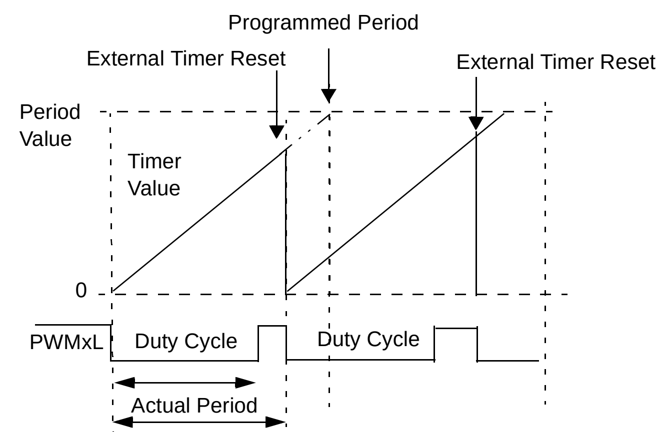


43.16.6 Constant Off-Time PWM

Constant Off-Time PWM, shown in Figure 43-40, 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 43-40: Constant Off-Time PWM

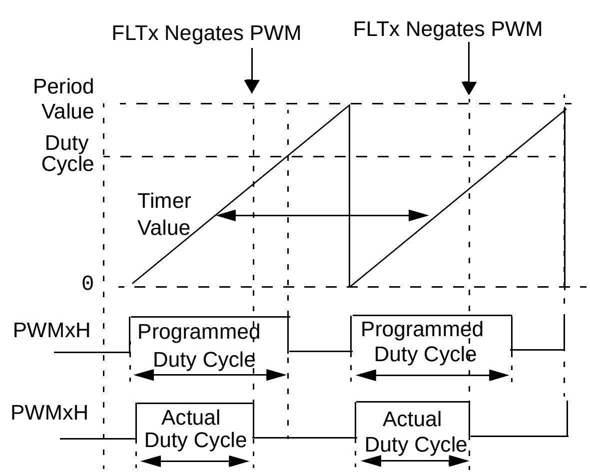


Note: Duty Cycle represents OFF-Time.

43.16.7 Current-Limit PWM

The cycle-by-cycle current-limit shown in Figure 43-41 truncates the asserted PWM signal when the selected external fault signal is asserted. The PWM output values are specified by the current-limit override bits (CLDAT<1:0>) in the PWM I/O Control (IOCONx<3:2>) registers. The override outputs remain in effect until the beginning of the next PWM cycle. This is sometimes used in Power Factor Correction (PFC) circuits where the inductor current controls the PWM On-Time. This is a constant frequency PWM.

Figure 43-41: Current-Limit PWM



43.17 REGISTER MAP

Table 43-5 maps the bit functions for the High-Speed PWM control registers.

Table 43-5: 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<1:0>		SEVTPS<3:0>				0000
PTCON2	—	—	—	—	—	—	—	—	—	—	—	—	—	PCLKDIV<2:0>			0000
PTPER	PTPER<15:0>																FFF8
SEVTCMP	SEVTCMP<15:3>													—	—	—	0000
MDC	MDC<15:0>																0000
PWMCON1	FLTSTAT	CLSTAT	TRGSTAT	FLTIEN	CLIEN	TRGIEN	ITB	MDCS	DTC<1:0>		—	—	—	CAM	XPRES	IUE	0000
PWMCON2	FLTSTAT	CLSTAT	TRGSTAT	FLTIEN	CLIEN	TRGIEN	ITB	MDCS	DTC<1:0>		—	—	—	CAM	XPRES	IUE	0000
PWMCON3	FLTSTAT	CLSTAT	TRGSTAT	FLTIEN	CLIEN	TRGIEN	ITB	MDCS	DTC<1:0>		—	—	—	CAM	XPRES	IUE	0000
PWMCON4	FLTSTAT	CLSTAT	TRGSTAT	FLTIEN	CLIEN	TRGIEN	ITB	MDCS	DTC<1:0>		—	—	—	CAM	XPRES	IUE	0000
IOCON1	PENH	PENL	POLH	POLL	PMOD<1:0>		OVRENH	OVRENL	OVRDAT<1:0>		FLTDAT<1:0>		CLDAT<1:0>		SWAP	OSYNC	0000
IOCON2	PENH	PENL	POLH	POLL	PMOD<1:0>		OVRENH	OVRENL	OVRDAT<1:0>		FLTDAT<1:0>		CLDAT<1:0>		SWAP	OSYNC	0000
IOCON3	PENH	PENL	POLH	POLL	PMOD<1:0>		OVRENH	OVRENL	OVRDAT<1:0>		FLTDAT<1:0>		CLDAT<1:0>		SWAP	OSYNC	0000
IOCON4	PENH	PENL	POLH	POLL	PMOD<1:0>		OVRENH	OVRENL	OVRDAT<1:0>		FLTDAT<1:0>		CLDAT<1:0>		SWAP	OSYNC	0000
FCLCON1	IFLTMOD	CLSRC<4:0>					CLPOL	CLMOD	FLTSRC<4:0>					FLTPOL	FLTMOD<1:0>		0000
FCLCON2	IFLTMOD	CLSRC<4:0>					CLPOL	CLMOD	FLTSRC<4:0>					FLTPOL	FLTMOD<1:0>		0000
FCLCON3	IFLTMOD	CLSRC<4:0>					CLPOL	CLMOD	FLTSRC<4:0>					FLTPOL	FLTMOD<1:0>		0000
FCLCON4	IFLTMOD	CLSRC<4:0>					CLPOL	CLMOD	FLTSRC<4:0>					FLTPOL	FLTMOD<1:0>		0000
PDC1	PDC1<15:0>																0000
PDC2	PDC2<15:0>																0000
PDC3	PDC3<15:0>																0000
PDC4	PDC4<15:0>																0000
PHASE1	PHASE1<15:0>																0000
PHASE2	PHASE2<15:0>																0000
PHASE3	PHASE3<15:0>																0000
PHASE4	PHASE4<15:0>																0000
DTR1	—	—	DTR1<13:0>														0000
DTR2	—	—	DTR2<13:0>														0000
DTR3	—	—	DTR3<13:0>														0000
DTR4	—	—	DTR4<13:0>														0000
ALTDTR1	—	—	ALTDTR1<13:0>														0000
ALTDTR2	—	—	ALTDTR2<13:0>														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.

High-Speed PWM 43

Table 43-5: 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
ALTDTR3	—	—	ALTDTR3<13:0>														0000
ALTDTR4	—	—	ALTDTR4<13:0>														0000
SDC1	SDC1<15:0>																0000
SDC2	SDC2<15:0>																0000
SDC3	SDC3<15:0>																0000
SDC4	SDC4<15:0>																0000
SPHASE1	SPHASE1<15:0>																0000
SPHASE2	SPHASE2<15:0>																0000
SPHASE3	SPHASE3<15:0>																0000
SPHASE4	SPHASE4<15:0>																0000
TRIG1	TRGCMP<15:3>													—	—	—	0000
TRIG2	TRGCMP<15:3>													—	—	—	0000
TRIG3	TRGCMP<15:3>													—	—	—	0000
TRIG4	TRGCMP<15:3>													—	—	—	0000
TRGCON1	TRGDIV<3:0>				—	—	—	—	DTM	—	TRGSTRT<5:0>						0000
TRGCON2	TRGDIV<3:0>				—	—	—	—	DTM	—	TRGSTRT<5:0>						0000
TRGCON3	TRGDIV<3:0>				—	—	—	—	DTM	—	TRGSTRT<5:0>						0000
TRGCON4	TRGDIV<3:0>				—	—	—	—	DTM	—	TRGSTRT<5:0>						0000
STRIG1	STRGCMP<15:3>													—	—	—	0000
STRIG2	STRGCMP<15:3>													—	—	—	0000
STRIG3	STRGCMP<15:3>													—	—	—	0000
STRIG4	STRGCMP<15:3>													—	—	—	0000
PWMCAP1	PWMCAP1<15:3>													—	—	—	0000
PWMCAP2	PWMCAP2<15:3>													—	—	—	0000
PWMCAP3	PWMCAP3<15:3>													—	—	—	0000
PWMCAP4	PWMCAP4<15:3>													—	—	—	0000
LEBCON1	PHR	PHF	PLR	PLF	FLTLEBEN	CLLEBEN	LEB<9:3>							—	—	—	0000
LEBCON2	PHR	PHF	PLR	PLF	FLTLEBEN	CLLEBEN	LEB<9:3>							—	—	—	0000
LEBCON3	PHR	PHF	PLR	PLF	FLTLEBEN	CLLEBEN	LEB<9:3>							—	—	—	0000
LEBCON4	PHR	PHF	PLR	PLF	FLTLEBEN	CLLEBEN	LEB<9:3>							—	—	—	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.

43.18 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 dsPIC33F 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.

Note:	Please visit the Microchip web site (www.microchip.com) for additional Application Notes and code examples for the dsPIC33F family of devices.

43.19 REVISION HISTORY

Revision A (February 2008)

This is the initial released revision of this document.

Revision B (September 2008)

This revision incorporates the following updates:

- Equations:
  - Updated Equation 43-3 in **43.6 “PWM Generator”**.
  - Updated Equation 43-4 in **43.6.2.3 “Secondary Duty Cycle (SDCx)”**
- Examples:
  - Added an example for PWM Clock Code in **43.5.1 “PWM Clock Selection”**.
- Figures:
  - Updated the labels in Figure 43-4.
  - Included new figure in **43.6.7 “Dead Time Resolution”** (see Figure 43-10).
  - Updated the fault source values in Figure 43-28 and Figure 43-30.
- Headings:
  - Added Auxiliary PLL as a new section (see **43.5.1 “PWM Clock Selection”** in **43.5 “Module Description”**.
  - The description for Dead Time Distortion has been corrected in **43.6.6 “Dead Time Distortion”**.
  - Added a new section on Dead-Time Insertion in Center-Aligned Mode (see **43.6.8 “Dead Time Insertion in Center-Aligned Mode”**.
  - Added a new sub-section for PWM Fault Generator (see **43.10.1 “PWM Fault Generated by the Analog Comparator”**) in **43.10 “PWM Fault Pins”**.
- Notes:
  - Added a note on nominal input clock to the PWM in **43.5.1 “PWM Clock Selection”**.
  - Added a note for the boundary conditions of the PWM resolution in the following registers:
    - MDC: PWM Master Duty Cycle Register (see Note 2 in Register 43-5)
    - PDCx: PWM Generator Duty Cycle Register (see Note 2 in Register 43-7)
    - SDCx: PWM Secondary Duty Cycle Register (see Note 2 in Register 43-8)
  - Added a note for using Fault 1 for Current-Limit mode (CLSRC<4:0> = `b0000`) in Register 43-16 (see Note 2).
  - Added a note for configuring the auxiliary clock in **43.5.1 “PWM Clock Selection”**.
  - Added a note on resetting the local time base in **43.16.5 “Current Reset PWM ”**.
- Registers:
  - The register descriptions for the **PDCx: PWM Generator Duty Cycle Register** and **SDCx: PWM Secondary Duty Cycle Register** have been corrected.
  - The bit descriptions for bit 14-10 and bit 7-3 in Register 43-16 have been corrected.
  - Updated the bit field value of LEB as LEB<4:0> and LEB<6:5> in LEBCONx: Leading-Edge Blanking Control Register (see Register 43-18).
  - The Read/Write state for the bit 3 through bit 15 have been corrected in PWMCAPx: Primary PWM Time Base Capture Register (see Register 43-19).
- Sections:
  - The terms Complementary Output Mode and Complementary PWM Mode have been corrected as Complementary Mode in the entire document.
  - The terms Push-Pull Output Mode and Push-Pull Mode have been corrected as Push-Pull Mode in the entire document.
- Additional minor corrections such as language and formatting updates are incorporated throughout the document.