PWM_8B6C
Block User Guide
V01.14

Original Release Date: 12 MAR 1998
Revised: 14 Mar 2002

Motorola, Inc.

Revision History

Table 0-1 Revision History

Version Number	Revision Date	Effective Date	Author	Description of Changes
00.00		3-12-98		First pass release
00.10		3-15-98		- Updates of Section 1 based on peer review and internal spec review. - Added initial information into Section 2.
00.20		4-7-98		- Updates of Section 1 and Section 2 per MSIL review. - Updated cover page per latest spec template review.
01.00		4-15-98		- Updated per Rev. 3.0 TSCS Module Spec Template. - Changed the Port and DDR register names to match the latest HCS12 naming convention. - Added the reset state under each counter, period, and duty register in the Register Description sections. - Added Design for Testability sub-section in Section 2 to describe scan implementation. - Updated Module I/O signal names in Section 2 per the latest HCS12 signal naming convention. - Frozen PWM spec sent to Delco.
01.01		4-30-98		- Added Document Number (12MRE31052W) to cover page of spec per QS9000 requirements. - Changed PWMEN register to PWME and also changed the bit names from PWENx to PWMEx to follow the enable naming convention. - Changed PWMCEN register to PWMCAE and also changed the bit names from CENx to CAEx to avoid having the same register name as the PWMC module. - Section 2 Module I/O list changed to have only 1 input buffer enable signal (pwm_ibe_t4) for the entire port. The reset signal name was also changed to vsc_reset_t4 per the latest bus definition document. - Added further clarification on DISCRW test bit in Section 2. If set, duty and period registers are not loaded with the buffer value.
01.02		5-14-98		- Updated per Rev. 3.1, 3.2 and 3.3 TSCS Module Spec Template. - Removed Table 1-1 PWM Register Address Summary. Added the Address Offset along side the registers in Figure 1-2 PWM Register Map. - Added WARNING regarding writing to the test registers in special modes. - Added footnote regarding the counter value in the Period=0 boundary case. - Removed “weasel” words--may and should.

Version Number	Revision Date	Effective Date	Author	Description of Changes
01.03		5-27-98		- Added clarification on how the counter counts in left and center aligned output modes. - Added further clarification on the Period=0 boundary case. Added that the counter=$00. - Added further clarification on what occurs on writes to the counter--output is changed according to the polarity bit. - Added Caution regarding the first PWM cycle after the channel is enabled can be irregular. - Replaced bit 'RDP' with 'RDPPWM' and bit 'PUPP' with 'PUPPWME' to match port control bit naming conventions. - Added 'iam8bit' signal in Table 2-1. - Added Table 2-2, Engineering Electrical Specs. - Added statement in Section 2 regarding DISCRW bit in PWMTST register. When bit is set, the output is not changed according to the polarity bit. - Added statement in Section 2 regarding DISCRM bit in PWMTST register. When bit is set, the duty and period registers do not get loaded with the buffer value. - Corrected left and center aligned max PWM output frequencies in Table A-2. - Created Table A-3 for the PWM Period/Duty Resolution Characteristics. - Miscellaneous clean up.
01.04		7-1-98		* Changed reset state of PWMPERx and PWMDTYx registers to FF. * In section 2 : changed some bus interface signal names : vsc_wait_t2 changed to vsc_wait_t3 vsc_bdmact_t2 changed to vsc_bdmact_t4 vsc_smod_t2 changed to vsc_smod_t4 * In section 2 : Added a note that in concatenated, left aligned, DISCRW=1 writing 16 bit (high-byte-data, low-byte-data) to the counter causes the high byte of the counter to start counting from (high-byte-data) and the low byte of the counter to start counting from (low-byte-data + 1).

Version Number	Revision Date	Effective Date	Author	Description of Changes
01.05		1-6-99		Includes spec tagging in conditional text. There are 3 conditional text tags : Tested- Functional Test (Blue), STATEMENT- Statement (Green), Test Outside Submodule (red). Summary of changes: * Section 1 : - Added the following sentence in section 1.6.2.4 : When the channel is disabled, writing ‘0’ to the period register will cause the counter to reset on the next selected clock. - Added a caution in section 1.7.15.1 about reading from port register after changing pin to input. - In section 1.9 hanged ‘Reset state : The prescale free running counter begins to increment’ to : "All the channels are disabled and all the counters don’t count". * Section 2 : - Section 2.2.3 Design for Test : Scan is not implemented on the PWM module. During scan mode (vsc_scanmod = 1) the module is not selected (vsc_pwmsel_t3 = 0) and the module’s internal clocks stop. - Table 2-1 PWM Module I/O Signals: Removed scane, changed pwm_purst_plug to pwm_puerst_plug, changed vsc_wait_t3 to vsc_wait_t2, changed pwm_outdata_t4[7:0] to pwm_do_t4[7:0], changed pad_indata[7:0] to pwmp_ind[7:0], removed iam8bit, added vsc_en2drv, changed vsc_bdmact_t4 to vsc_bdmact_t2, changed vsc_smod_t4 to vsc_smod_t2. - Added section 2.3.3 and table 2-2 Port Pin Connections. - Section 2.4 table 2-3 : Changed Vdd Value to 3 to 3.6v (it was 2.7 to 3.6). Changed System Clock Value to dc to 16MHz (it was 20). - Section 2.4 : added Figure 2-7 : PWM Timing Diagram. It’s like A.6 in the appendix, but includes more details. * Appendix A : Table A-1 : System Clock dc to 16MHz (it was 20). Table A-2 : E-clock 16MHz (it was 20). A.6 added PWM Timing Diagram.

Version Number	Revision Date	Effective Date	Author	Description of Changes
01.06		09-02-99		* In section 1, Added section 1.7.15 shutdown register (PWMSDN), changed the sec # for the subsequent sections for the registers. * Added emergency shutdown feature in the feature list (sec 1.3) * Added the PWMSDN in the register map (sec 1.5) * Removed $_24 from resvd. reg list in sec 1.7.17 * Modified sec 1.12, Interrupt Op., to support the intr. for emergency shutdown, to The PWM module has only one interrupt which is generated at the time of emergency shutdown, if the corresponding enable bit (PWMSDN[6]) is set. * Section 2.2.3 : Design for Test : The PWM module will be fully scannable as per the project DFT guidelines * removed GPIO note from section 1.2, 1.3, 1.6.2.1, 1.6.2.7, 1.7.1 * renamed PSBCK to PFRZ in PWMCTL, and removed RDPPWM & PUPPWME bits * changed the interface signals for IP bus. * removed electrical spec details.
01.07		10-25-99		updated the specs after feedback from Munich (removed redundant port signals : ibe, offval, obe[6:0])
01.08		11-25-99		added ipb_clk signal to the I/O list. the same is used for reg. writes wherever possible.
01.09		12-08-99		Restart (from shutdown) functionality clarified
01.10		17-01-00		Tagging Done for Barracuda
01.11		15-01-00		Tagging for wait mode and freeze mode.
01.12		7-01-00		Took the reference spec for Barracuda and made changes to the spec for Mako according to the requirement of stripping the number of PWM channels from 8 to 6.
01.13		03-29-2001		Updated according to feedback concerning SRS v2 and additional rules compliance
01.14		03-14-2002		Syntax corrections, document formal updates

Table of Contents

Section 1 Introduction

Section 2 Signal Description

Section 3 Memory Map and Register Definition

Section 4 Functional Description

Section 5 Resets

Section 6 Interrupts

List of Figures

List of Tables

Section 1 Introduction

1.1 Overview

The Pulse Width Modulation (PWM) definition is based on the HC12 PWM definitions. The PWM_8B6C module contains the basic features from the HC11 with some of the enhancements incorporated on the HC12, that is center aligned output mode and four available clock sources. The PWM_8B6C module has six channels with independent control of left and center aligned outputs on each channel.

Each of the six PWM channels has a programmable period and duty cycle as well as a dedicated counter. A flexible clock select scheme allows a total of four different clock sources to be used with the counters. Each of the modulators can create independent continuous waveforms with software-selectable duty rates from 0% to 100%. The PWM outputs can be programmed as left aligned outputs or center aligned outputs

1.2 Features

- Six independent PWM channels with programmable period and duty cycle
- Dedicated counter for each PWM channel
- Programmable PWM enable/disable for each channel
- Software selection of PWM duty pulse polarity for each channel
- Period and duty cycle are double buffered. Change takes effect when the end of the effective period is reached (PWM counter reaches zero) or when the channel is disabled.
- Programmable center or left aligned outputs on individual channels
- Six 8-bit channel or three 16-bit channel PWM resolution
- Four clock sources (A, B, SA and SB) provide for a wide range of frequencies.
- Programmable Clock Select Logic
- Emergency shutdown

1.3 Modes of Operation

There is a software programmable option for low power consumption in Wait mode that disables the input clock to the prescaler.

In freeze mode there is a software programmable option to disable the input clock to the prescaler. This is useful for emulation.

1.4 Block Diagram

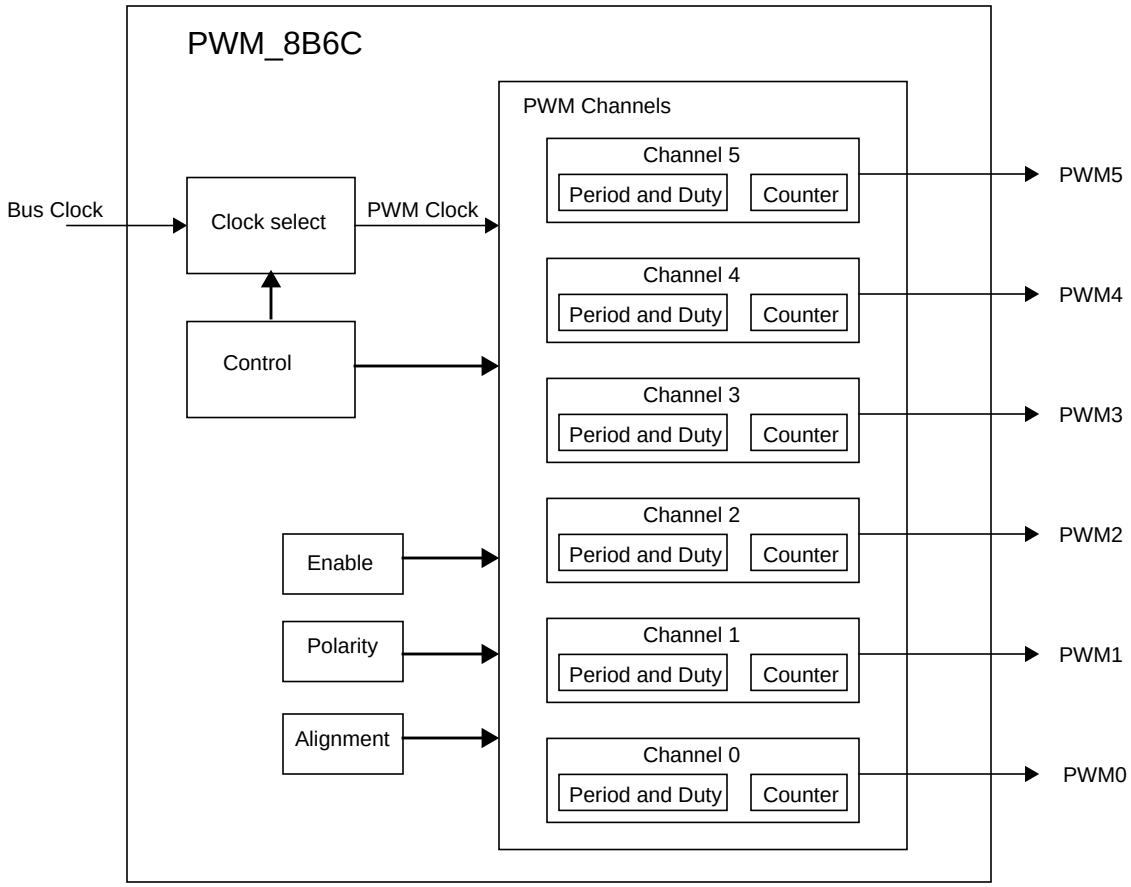

Figure 1-1 PWM_8B6C Block Diagram

Section 2 Signal Description

2.1 Overview

The PWM_8B6C module has a total of 6 external pins.

2.2 Detailed Signal Descriptions

2.2.1 PWM5 — Pulse Width Modulator Channel 5

This pin serves as waveform output of PWM channel 5 and as an input for the emergency shutdown feature.

2.2.2 PWM4 — Pulse Width Modulator Channel 4

This pin serves as waveform output of PWM channel 4.

2.2.3 PWM3 — Pulse Width Modulator Channel 3

This pin serves as waveform output of PWM channel 3.

2.2.4 PWM2 — Pulse Width Modulator Channel 2

This pin serves as waveform output of PWM channel 2.

2.2.5 PWM1 — Pulse Width Modulator Channel 1

This pin serves as waveform output of PWM channel 1.

2.2.6 PWM0 — Pulse Width Modulator Channel 0

This pin serves as waveform output of PWM channel 0.

Section 3 Memory Map and Register Definition

3.1 Overview

This section describes in detail all the registers and register bits in the PWM_8B6C module.

The special-purpose registers and register bit functions that would not normally made available to device end users, such as factory test control registers and reserved registers are clearly identified by means of shading the appropriate portions of address maps and register diagrams. Notes explaining the reasons for restricting access to the registers and functions are also explained in the individual register descriptions.

3.2 Module Memory Map

This section describes the content of the registers in the PWM_8B6C module. The base address of the PWM_8B6C module is determined at the MCU level when the MCU is defined. The register decode map is fixed and begins at the first address of the module address offset. The figure below shows the registers associated with the PWM and their relative offset from the base address. The register detail description follows the order they appear in the register map.

Reserved bits within a register will always read as 0 and the write will be unimplemented. Unimplemented functions are indicated by shading the bit.

Table 3-1 shows the memory map for the PWM_8B6C module

Table 3-1 PWM_8B6C Memory Map

Address	Use	Access
$_00	PWM Enable Register (PWME)	R/W
$_01	PWM Polarity Register (PWMPOL)	R/W
$_02	PWM Clock Select Register (PWMCLK)	R/W
$_03	PWM Prescale Clock Select Register (PWMPRCLK)	R/W
$_04	PWM Center Align Enable Register (PWMCAE)	R/W
$_05	PWM Control Register (PWMCTL)	R/W
$_06	PWM Test Register (PWMTST)[1]	R/W
$_07	PWM Prescale Counter Register (PWMPRSC)[2]	R/W
$_08	PWM Scale A Register (PWMSCLA)	R/W
$_09	PWM Scale B Register (PWMSCLB)	R/W
$_0A	PWM Scale A Counter Register (PWMSCNTA)[3]	R/W
$_0B	PWM Scale B Counter Register (PWMSCNTB)[4]	R/W
$_0C	PWM Channel 0 Counter Register (PWMCNT0)	R/W
$_0D	PWM Channel 1 Counter Register (PWMCNT1)	R/W
$_0E	PWM Channel 2 Counter Register (PWMCNT2)	R/W
$_0F	PWM Channel 3 Counter Register (PWMCNT3)	R/W
$_10	PWM Channel 4 Counter Register (PWMCNT4)	R/W
$_11	PWM Channel 5 Counter Register (PWMCNT5)	R/W
$_12	PWM Channel 0 Period Register (PWMPER0)	R/W

Table 3-1 PWM_8B6C Memory Map

$_13	PWM Channel 1 Period Register (PWMPER1)	R/W
$_14	PWM Channel 2 Period Register (PWMPER2)	R/W
$_15	PWM Channel 3 Period Register (PWMPER3)	R/W
$_16	PWM Channel 4 Period Register (PWMPER4)	R/W
$_17	PWM Channel 5 Period Register (PWMPER5)	R/W
$_18	PWM Channel 0 Duty Register (PWMDTY0)	R/W
$_19	PWM Channel 1 Duty Register (PWMDTY1)	R/W
$_1A	PWM Channel 2 Duty Register (PWMDTY2)	R/W
$_1B	PWM Channel 3 Duty Register (PWMDTY3)	R/W
$_1C	PWM Channel 4 Duty Register (PWMDTY4)	R/W
$_1D	PWM Channel 5 Duty Register (PWMDTY5)	R/W
$_1E	PWM Shutdown Register (PWMSDN)	R/W

NOTES:
1. PWMTST is intended for factory test purposes only.
2. PWMPRSC is intended for factory test purposes only.
3. PWMSCNTA is intended for factory test purposes only.
4. PWMSCNTB is intended for factory test purposes only.

NOTE: *Register Address = Base Address + Address Offset, where the Base Address is defined at the MCU level and the Address Offset is defined at the module level.*

3.3 Register Descriptions

This section describes in detail all the registers and register bits in the PWM_8B6C module.

3.3.1 PWM Enable Register (PWME)

Each PWM channel has an enable bit (PWMEx) to start its waveform output. When any of the PWMEx bits are set (PWMEx=1), the associated PWM output is enabled immediately. However, the actual PWM waveform is not available on the associated PWM output until its clock source begins its next cycle due to the synchronization of PWMEx and the clock source.

NOTE: *The first PWM cycle after enabling the channel can be irregular.*

An exception to this is when channels are concatenated. Once concatenated mode is enabled (CONxx bits set in PWMCTL register) then enabling/disabling the corresponding 16-bit PWM channel is controlled by the low order PWMEx bit.In this case, the high order bytes PWMEx bits have no effect and their corresponding PWM output lines are disabled.

While in run mode, if all six PWM channels are disabled (PWME5-0=0), the prescaler counter shuts off for power savings.

$_00

	7	6	5	4	3	2	1	0
R	0	0	PWME5	PWME4	PWME3	PWME2	PWME1	PWME0
W								
RESET:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 3-1 PWM Enable Register (PWME)

Read: anytime

Write: anytime

PWME5 — Pulse Width Channel 5 Enable

1 = Pulse Width channel 5 is enabled. The pulse modulated signal becomes available at PWM, output bit 5 when its clock source begins its next cycle.

0 = Pulse Width channel 5 is disabled.

PWME4 — Pulse Width Channel 4 Enable

1 = Pulse Width channel 4 is enabled. The pulse modulated signal becomes available at PWM, output bit 4 when its clock source begins its next cycle. If CON45=1, then bit has no effect and PWM output line4 is disabled.

0 = Pulse Width channel 4 is disabled.

PWME3 — Pulse Width Channel 3 Enable

1 = Pulse Width channel 3 is enabled. The pulse modulated signal becomes available at PWM, output bit 3 when its clock source begins its next cycle.

0 = Pulse Width channel 3 is disabled.

PWME2 — Pulse Width Channel 2 Enable

1 = Pulse Width channel 2 is enabled. The pulse modulated signal becomes available at PWM, output bit 2 when its clock source begins its next cycle. If CON23=1, then bit has no effect and PWM output line2 is disabled.

0 = Pulse Width channel 2 is disabled.

PWME1 — Pulse Width Channel 1 Enable

1 = Pulse Width channel 1 is enabled. The pulse modulated signal becomes available at PWM, output bit 1 when its clock source begins its next cycle.

0 = Pulse Width channel 1 is disabled.

PWME0 — Pulse Width Channel 0 Enable

1 = Pulse Width channel 0 is enabled. The pulse modulated signal becomes available at PWM, output bit 0 when its clock source begins its next cycle. If CON01=1, then bit has no effect and PWM output line0 is disabled.

0 = Pulse Width channel 0 is disabled.

3.3.2 PWM Polarity Register (PWMPOL)

The starting polarity of each PWM channel waveform is determined by the associated PPOLx bit in the PWMPOL register. If the polarity bit is one, the PWM channel output is high at the beginning of the cycle and then goes low when the duty count is reached. Conversely, if the polarity bit is zero, the output starts low and then goes high when the duty count is reached.

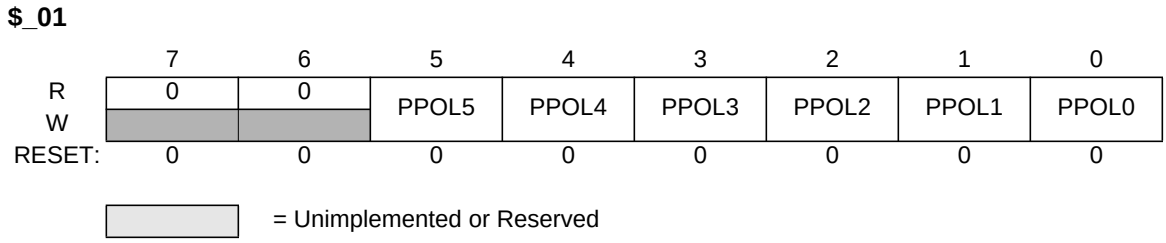

Figure 3-2 PWM Polarity Register (PWMPOL)

Read: anytime

Write: anytime

NOTE: *PPOLx register bits can be written anytime. If the polarity is changed while a PWM signal is being generated, a truncated or stretched pulse can occur during the transition*

PPOL5 — Pulse Width Channel 5 Polarity

1 = PWM channel 5 output is high at the beginning of the period, then goes low when the duty count is reached.

0 = PWM channel 5 output is low at the beginning of the period, then goes high when the duty count is reached.

PPOL4 — Pulse Width Channel 4 Polarity

1 = PWM channel 4 output is high at the beginning of the period, then goes low when the duty count is reached.

0 = PWM channel 4 output is low at the beginning of the period, then goes high when the duty count is reached.

PPOL3 — Pulse Width Channel 3 Polarity

1 = PWM channel 3 output is high at the beginning of the period, then goes low when the duty count is reached.

0 = PWM channel 3 output is low at the beginning of the period, then goes high when the duty count is reached.

PPOL2 — Pulse Width Channel 2 Polarity

1 = PWM channel 2 output is high at the beginning of the period, then goes low when the duty count is reached.

0 = PWM channel 2 output is low at the beginning of the period, then goes high when the duty count is reached.

PPOL1 — Pulse Width Channel 1 Polarity

1 = PWM channel 1 output is high at the beginning of the period, then goes low when the duty count is reached.

0 = PWM channel 1 output is low at the beginning of the period, then goes high when the duty count is reached.

PPOL0 — Pulse Width Channel 0 Polarity

1 = PWM channel 0 output is high at the beginning of the period, then goes low when the duty count is reached.

0 = PWM channel 0 output is low at the beginning of the period, then goes high when the duty count is reached

3.3.3 PWM Clock Select Register (PWMCLK)

Each PWM channel has a choice of two clocks to use as the clock source for that channel as described below.

$_02

	7	6	5	4	3	2	1	0
R	0	0	PCLK5	PCLK4	PCLK3	PCLK2	PCLK1	PCLK0
W								
RESET:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 3-3 PWM Clock Select Register (PWMCLK)

Read: anytime

Write: anytime

NOTE: *Register bits PCLK0 to PCLK5 can be written anytime. If a clock select is changed while a PWM signal is being generated, a truncated or stretched pulse can occur during the transition.*

PCLK5 — Pulse Width Channel 5 Clock Select

1 = Clock SA is the clock source for PWM channel 5.

0 = Clock A is the clock source for PWM channel 5.

PCLK4 — Pulse Width Channel 4 Clock Select

1 = Clock SA is the clock source for PWM channel 4.

0 = Clock A is the clock source for PWM channel 4.

PCLK3 — Pulse Width Channel 3 Clock Select

1 = Clock SB is the clock source for PWM channel 3.

0 = Clock B is the clock source for PWM channel 3.

PCLK2 — Pulse Width Channel 2 Clock Select
1 = Clock SB is the clock source for PWM channel 2.
0 = Clock B is the clock source for PWM channel 2.

PCLK1 — Pulse Width Channel 1 Clock Select
1 = Clock SA is the clock source for PWM channel 1.
0 = Clock A is the clock source for PWM channel 1.

PCLK0 — Pulse Width Channel 0 Clock Select
1 = Clock SA is the clock source for PWM channel 0.
0 = Clock A is the clock source for PWM channel 0.

3.3.4 PWM Prescale Clock Select Register (PWMPRCLK)

This register selects the prescale clock source for clocks A and B independently.

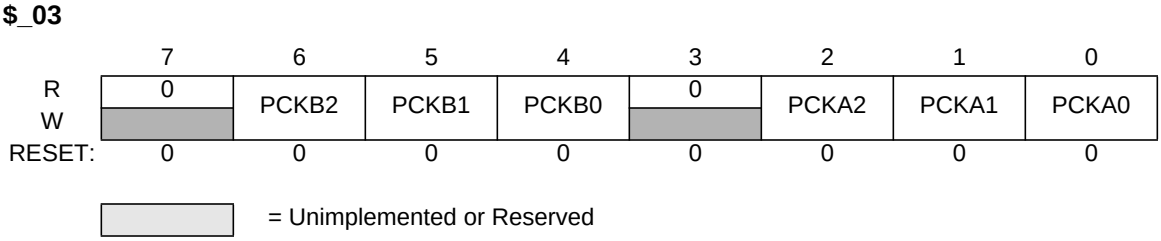

Figure 3-4 PWM Prescale Clock Select Register (PWMPRCLK)

Read: anytime

Write: anytime

NOTE: *PCKB2-0 and PCKA2-0 register bits can be written anytime. If the clock pre-scale is changed while a PWM signal is being generated, a truncated or stretched pulse can occur during the transition.*

PCKB2 – PCKB0 — Prescaler Select for Clock B

Clock B is one of two clock sources which can be used for channels 2 or 3. These three bits determine the rate of clock B, as shown in the following table.

Table 3-2 Clock B Prescaler Selects

PCKB2	PCKB1	PCKB0	Value of Clock B
0	0	0	bus clock
0	0	1	bus clock / 2
0	1	0	bus clock / 4

PCKB2	PCKB1	PCKB0	Value of Clock B
0	1	1	bus clock / 8
1	0	0	bus clock / 16
1	0	1	bus clock / 32
1	1	0	bus clock / 64
1	1	1	bus clock / 128

PCKA2 – PCKA0 — Prescaler Select for Clock A

Clock A is one of two clock sources which can be used for channels 0, 1, 4 or 5. These three bits determine the rate of clock A, as shown in the following table.

Table 3-3 Clock A Prescaler Selects

PCKA2	PCKA1	PCKA0	Value of Clock A
0	0	0	bus clock
0	0	1	bus clock / 2
0	1	0	bus clock / 4
0	1	1	bus clock / 8
1	0	0	bus clock / 16
1	0	1	bus clock / 32
1	1	0	bus clock / 64
1	1	1	bus clock / 128

3.3.5 PWM Center Align Enable Register (PWMCAE)

The PWMCAE register contains six control bits for the selection of center aligned outputs or left aligned outputs for each PWM channel. If the CAEx bit is set to a one, the corresponding PWM output will be center aligned. If the CAEx bit is cleared, the corresponding PWM output will be left aligned. Reference **4.2.5 Left Aligned Outputs** and **4.2.6 Center Aligned Outputs** for a more detailed description of the PWM output modes.

$_04

	7	6	5	4	3	2	1	0
R	0	0	CAE5	CAE4	CAE3	CAE2	CAE1	CAE0
W								
RESET:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 3-5 PWM Center Align Enable Register (PWMCAE)

Read: anytime

Write: anytime

NOTE: *Write these bits only when the corresponding channel is disabled.*

CAE5 — Center Aligned Output Mode on channel 5
1 = Channel 5 operates in Center Aligned Output Mode.
0 = Channel 5 operates in Left Aligned Output Mode.

CAE4 — Center Aligned Output Mode on channel 4
1 = Channel 4 operates in Center Aligned Output Mode.
0 = Channel 4 operates in Left Aligned Output Mode.

CAE3 — Center Aligned Output Mode on channel 3
1 = Channel 3 operates in Center Aligned Output Mode.
0 = Channel 3 operates in Left Aligned Output Mode.

CAE2 — Center Aligned Output Mode on channel 2
1 = Channel 2 operates in Center Aligned Output Mode.
0 = Channel 2 operates in Left Aligned Output Mode.

CAE1 — Center Aligned Output Mode on channel 1
1 = Channel 1 operates in Center Aligned Output Mode.
0 = Channel 1 operates in Left Aligned Output Mode.

CAE0 — Center Aligned Output Mode on channel 0
1 = Channel 0 operates in Center Aligned Output Mode.
0 = Channel 0 operates in Left Aligned Output Mode.

3.3.6 PWM Control Register (PWMCTL)

The PWMCTL register provides for various control of the PWM module.

$_05

	7	6	5	4	3	2	1	0
R	0	CON45	CON23	CON01	PSWAI	PFRZ	0	0
W								
RESET:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 3-6 PWM Control Register (PWMCTL)

Read: anytime

Write: anytime

There are three control bits for concatenation, each of which is used to concatenate a pair of PWM channels into one 16-bit channel. When channels 4 and 5 are concatenated, channel 4 registers become the high order bytes of the double byte channel. When channels 2 and 3 are concatenated, channel 2 registers become the high order bytes of the double byte channel. When channels 0 and 1 are concatenated, channel 0 registers become the high order bytes of the double byte channel.

Reference **4.2.7 PWM 16-Bit Functions** for a more detailed description of the concatenation PWM Function.

NOTE: *Change these bits only when both corresponding channels are disabled.*

CON45 — Concatenate channels 4 and 5

1 = Channels 4 and 5 are concatenated to create one 16-bit PWM channel. Channel 4 becomes the high order byte and channel 5 becomes the low order byte. Channel 5 output pin is used as the output for this 16-bit PWM (bit 5 of port PWMP). Channel 5 clock select control-bit determines the clock source, channel 5 polarity bit determines the polarity, channel 5 enable bit enables the output and channel 5 center aligned enable bit determines the output mode.

0 = Channels 4 and 5 are separate 8-bit PWMs.

CON23 — Concatenate channels 2 and 3

1 = Channels 2 and 3 are concatenated to create one 16-bit PWM channel. Channel 2 becomes the high order byte and channel 3 becomes the low order byte. Channel 3 output pin is used as the output for this 16-bit PWM (bit 3 of port PWMP). Channel 3 clock select control-bit determines the clock source, channel 3 polarity bit determines the polarity, channel 3 enable bit enables the output and channel 3 center aligned enable bit determines the output mode.

0 = Channels 2 and 3 are separate 8-bit PWMs.

CON01 — Concatenate channels 0 and 1

1 = Channels 0 and 1 are concatenated to create one 16-bit PWM channel. Channel 0 becomes the high order byte and channel 1 becomes the low order byte. Channel 1 output pin is used as the output for this 16-bit PWM (bit 1 of port PWMP). Channel 1 clock select control-bit determines the clock source, channel 1 polarity bit determines the polarity, channel 1 enable bit enables the output and channel 1 center aligned enable bit determines the output mode.

0 = Channels 0 and 1 are separate 8-bit PWMs.

PSWAI — PWM Stops in Wait Mode

Enabling this bit allows for lower power consumption in Wait Mode by disabling the input clock to the prescaler.

1 = Stop the input clock to the prescaler whenever the MCU is in Wait Mode.

0 = Allow the clock to the prescaler to continue while in wait mode.

PFRZ — PWM Counters Stop in Freeze Mode

In Freeze Mode, there is an option to disable the input clock to the prescaler by setting the PFRZ bit in the PWMCTL register. If this bit is set, whenever the MCU is in freeze mode the input clock to the prescaler is disabled. This feature is useful during emulation as it allows the PWM function to be suspended. In this way, the counters of the PWM can be stopped while in freeze mode so that once

normal program flow is continued, the counters are re-enabled to simulate real-time operations. Since the registers can still be accessed in this mode, to re-enable the prescaler clock, either disable the PFRZ bit or exit freeze mode.

1 = Disable PWM input clock to the prescaler whenever the part is in freeze mode. This is useful for emulation.

0 = Allow PWM to continue while in freeze mode.

3.3.7 Reserved Register (PWMTST)

This register is reserved for factory testing of the PWM module and is not available in normal modes.

$_06

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W								
RESET:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 3-7 Reserved Register (PWMTST)

Read: always read $00 in normal modes

Write: unimplemented in normal modes

NOTE: *Writing to this register when in special modes can alter the PWM functionality.*

3.3.8 Reserved Register (PWMPRSC)

This register is reserved for factory testing of the PWM module and is not available in normal modes.

$_07

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W								
RESET:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 3-8 Reserved Register (PWMPRSC)

Read: always read $00 in normal modes

Write: unimplemented in normal modes

NOTE: *Writing to this register when in special modes can alter the PWM functionality.*

3.3.9 PWM Scale A Register (PWMSCLA)

PWMSCLA is the programmable scale value used in scaling clock A to generate clock SA. Clock SA is generated by taking clock A, dividing it by the value in the PWMSCLA register and dividing that by two.

Clock SA = Clock A / (2 * PWMSCLA)

NOTE: *When PWMSCLA = $00, PWMSCLA value is considered a full scale value of 256. Clock A is thus divided by 512.*

Any value written to this register will cause the scale counter to load the new scale value (PWMSCLA).

$_08

	7	6	5	4	3	2	1	0
R W	Bit 7	6	5	4	3	2	1	Bit 0
RESET:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 3-9 PWM Scale A Register (PWMSCLA)

Read: anytime

Write: anytime (causes the scale counter to load the PWMSCLA value)

3.3.10 PWM Scale B Register (PWMSCLB)

PWMSCLB is the programmable scale value used in scaling clock B to generate clock SB. Clock SB is generated by taking clock B, dividing it by the value in the PWMSCLB register and dividing that by two.

Clock SB = Clock B / (2 * PWMSCLB)

NOTE: *When PWMSCLB = $00, PWMSCLB value is considered a full scale value of 256. Clock B is thus divided by 512.*

Any value written to this register will cause the scale counter to load the new scale value (PWMSCLB).

$_09

	7	6	5	4	3	2	1	0
R W	Bit 7	6	5	4	3	2	1	Bit 0
RESET:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 3-10 PWM Scale B Register (PWMSCLB)

Read: anytime

Write: anytime (causes the scale counter to load the PWMSCLB value).

3.3.11 Reserved Registers (PWMSCNTx)

The registers PWMSCNTA and PWMSCNTB are reserved for factory testing of the PWM module and are not available in normal modes.

$_0A, $_0B

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W								
RESET:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 3-11 Reserved Registers (PWMSCNTx)

Read: always read $00 in normal modes

Write: unimplemented in normal modes

NOTE: *Writing to these registers when in special modes can alter the PWM functionality.*

3.3.12 PWM Channel Counter Registers (PWMCNTx)

Each channel has a dedicated 8-bit up/down counter which runs at the rate of the selected clock source. The counter can be read at any time without affecting the count or the operation of the PWM channel. In left aligned output mode, the counter counts from 0 to the value in the period register – 1. In center aligned output mode, the counter counts from 0 up to the value in the period register and then back down to 0.

Any value written to the counter causes the counter to reset to $00, the counter direction to be set to up, the immediate load of both duty and period registers with values from the buffers, and the output to change according to the polarity bit. The counter is also cleared at the end of the effective period (see Sections **4.2.5 Left Aligned Outputs** and **4.2.6 Center Aligned Outputs** for more details). When the channel is disabled (PWMEx=0), the PWMCNTx register does not count. When a channel becomes enabled (PWMEx=1), the associated PWM counter starts at the count in the PWMCNTx register. For more detailed information on the operation of the counters, reference **4.2.4 PWM Timer Counters**.

In concatenated mode, writes to the 16-bit counter by using a 16-bit access or writes to either the low or high order byte of the counter will reset the 16-bit counter. Reads of the 16-bit counter must be made by 16-bit access to maintain data coherency.

NOTE:*Writing to the counter while the channel is enabled can cause an irregular PWM cycle to occur.*

$_0C = PWMCNT0
$_0D = PWMCNT1
$_0E = PWMCNT2
$_0F = PWMCNT3
$_10 = PWMCNT4
$_11 = PWMCNT5

	7	6	5	4	3	2	1	0
R	Bit 7	6	5	4	3	2	1	Bit 0
W	0	0	0	0	0	0	0	0
RESET:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 3-12 PWM Channel Counter Registers (PWMCNTx)

Read: anytime

Write: anytime (any value written causes PWM counter to be reset to $00).

3.3.13 PWM Channel Period Registers (PWMPERx)

There is a dedicated period register for each channel. The value in this register determines the period of the associated PWM channel.

The period registers for each channel are double buffered so that if they change while the channel is enabled, the change will NOT take effect until one of the following occurs:

- The effective period ends
- The counter is written (counter resets to $00)
- The channel is disabled

In this way, the output of the PWM will always be either the old waveform or the new waveform, not some variation in between. If the channel is not enabled, then writes to the period register will go directly to the latches as well as the buffer.

NOTE: *Reads of this register return the most recent value written. Reads do not necessarily return the value of the currently active period due to the double buffering scheme.*

Reference **4.2.3 PWM Period and Duty** for more information.

To calculate the output period, take the selected clock source period for the channel of interest (A, B, SA, or SB) and multiply it by the value in the period register for that channel:

- Left Aligned Output (CAEx=0)
- PWMx Period = Channel Clock Period * PWMPERx Center Aligned Output (CAEx=1)

  PWMx Period = Channel Clock Period * (2 * PWMPERx)

  For Boundary Case programming values, please refer to Section **4.2.8 PWM Boundary Cases**

$_12 = PWMPER0
$_13 = PWMPER1
$_14 = PWMPER2
$_15 = PWMPER3
$_16 = PWMPER4
$_17 = PWMPER5

	7	6	5	4	3	2	1	0
R W	Bit 7	6	5	4	3	2	1	Bit 0
RESET:	1	1	1	1	1	1	1	1

= Unimplemented or Reserved

Figure 3-13 PWM Channel Period Registers (PWMPERx)

Read: anytime

Write: anytime

3.3.14 PWM Channel Duty Registers (PWMDTYx)

There is a dedicated duty register for each channel. The value in this register determines the duty of the associated PWM channel. The duty value is compared to the counter and if it is equal to the counter value a match occurs and the output changes state.

The duty registers for each channel are double buffered so that if they change while the channel is enabled, the change will NOT take effect until one of the following occurs:

- The effective period ends
- The counter is written (counter resets to $00)
- The channel is disabled

In this way, the output of the PWM will always be either the old duty waveform or the new duty waveform, not some variation in between. If the channel is not enabled, then writes to the duty register will go directly to the latches as well as the buffer.

NOTE: *Reads of this register return the most recent value written. Reads do not necessarily return the value of the currently active duty due to the double buffering scheme.*

Reference **4.2.3 PWM Period and Duty** for more information.

NOTE: *Depending on the polarity bit, the duty registers will contain the count of either the high time or the low time. If the polarity bit is one, the output starts high and then goes low when the duty count is reached, so the duty registers contain a count of the high time. If the polarity bit is zero, the output starts low and then goes high when the duty count is reached, so the duty registers contain a count of the low time.*

To calculate the output duty cycle (high time as a% of period) for a particular channel:

- Polarity = 0 (PPOLx=0)

  Duty Cycle = [(PWMPERx-PWMDTYx)/PWMPERx] * 100%

- Polarity = 1 (PPOLx=1)

  Duty Cycle = [PWMDTYx / PWMPERx] * 100%

- For Boundary Case programming values, please refer to Section **4.2.8 PWM Boundary Cases**.

$_18 = PWMDTY0
$_19 = PWMDTY1
$_1A = PWMDTY2
$_1B = PWMDTY3
$_1C = PWMDTY4
$_1D = PWMDTY5

	7	6	5	4	3	2	1	0
R W	Bit 7	6	5	4	3	2	1	Bit 0
RESET:	1	1	1	1	1	1	1	1

= Unimplemented or Reserved

Figure 3-14 PWM Channel Duty Registers (PWMDTYx)

Read: anytime

Write: anytime

3.3.15 PWM Shutdown Register (PWMSDN)

The PWMSDN register provides for the shutdown functionality of the PWM module in the emergency cases.

$_1E

	7	6	5	4	3	2	1	0
R	PWMIF	PWMIE	0	PWMLVL	0	PWM5IN	PWM5INL	PWM5ENA
W			PWMR-STRT					
RESET:	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 3-15 PWM Shutdown Register (PWMSDN)

Read: anytime

Write: anytime

PWM5ENA — PWM emergency shutdown Enable

If this bit is logic 1 the pin associated with channel 5 is forced to input and the emergency shutdown feature is enabled.All the other bits in this register are meaningful only if PWM5ENA = 1.

1 = PWM emergency feature is enabled.
0 = PWM emergency feature disabled.

PWM5INL — PWM shutdown active input level for channel 5.

If the emergency shutdown feature is enabled (PWM5ENA = 1), this bit determines the active level of the PWM5channel.

1 = Active level is high
0 = Active level is low

PWM5IN — PWM channel 5 input status.

This reflects the current status of the PWM5 pin.

PWMLVL — PWM shutdown output Level.

If active level as defined by the PWM5IN input, gets asserted all enabled PWM channels are immediately driven to the level defined by PWMLVL.

1 = PWM outputs are forced to 1.
0 = PWM outputs are forced to 0

PWMRSTRT — PWM Restart.

The PWM can only be restarted if the PWM channel input 5 is de-asserted. After writing a logic 1 to the PWMRSTRT bit (trigger event) the PWM channels start running after the corresponding counter passes next “counter == 0” phase.

Also if the PWM5ENA bit is reset to 0, the PWM do not start before the counter passes $00.
The bit is always read as “0”.

PWMIE — PWM Interrupt Enable

If interrupt is enabled an interrupt to the CPU is asserted.

1 = PWM interrupt is enabled.
0 = PWM interrupt is disabled.

PWMIF — PWM Interrupt Flag

Any change from passive to asserted (active) state or from active to passive state will be flagged by setting the PWMIF flag = 1. The flag is cleared by writing a logic 1 to it. Writing a 0 has no effect.

1 = change on PWM5IN input
0 = No change on PWM5IN input.

Section 4 Functional Description

4.1 PWM Clock Select

There are four available clocks called clock A, clock B, clock SA (Scaled A), and clock SB (Scaled B). These four clocks are based on the bus clock**.**

Clock A and B can be software selected to be 1, 1/2, 1/4, 1/8,..., 1/64, 1/128 times the bus clock. Clock SA uses clock A as an input and divides it further with a reloadable counter. Similarly, Clock SB uses clock B as an input and divides it further with a reloadable counter. The rates available for clock SA are software selectable to be clock A divided by 2, 4, 6, 8, ..., or 512 in increments of divide by 2. Similar rates are available for clock SB. Each PWM channel has the capability of selecting one of two clocks, either the pre-scaled clock (clock A or B) or the scaled clock (clock SA or SB).

The block diagram in **Figure 4-1 PWM Clock Select Block Diagram** shows the four different clocks and how the scaled clocks are created.

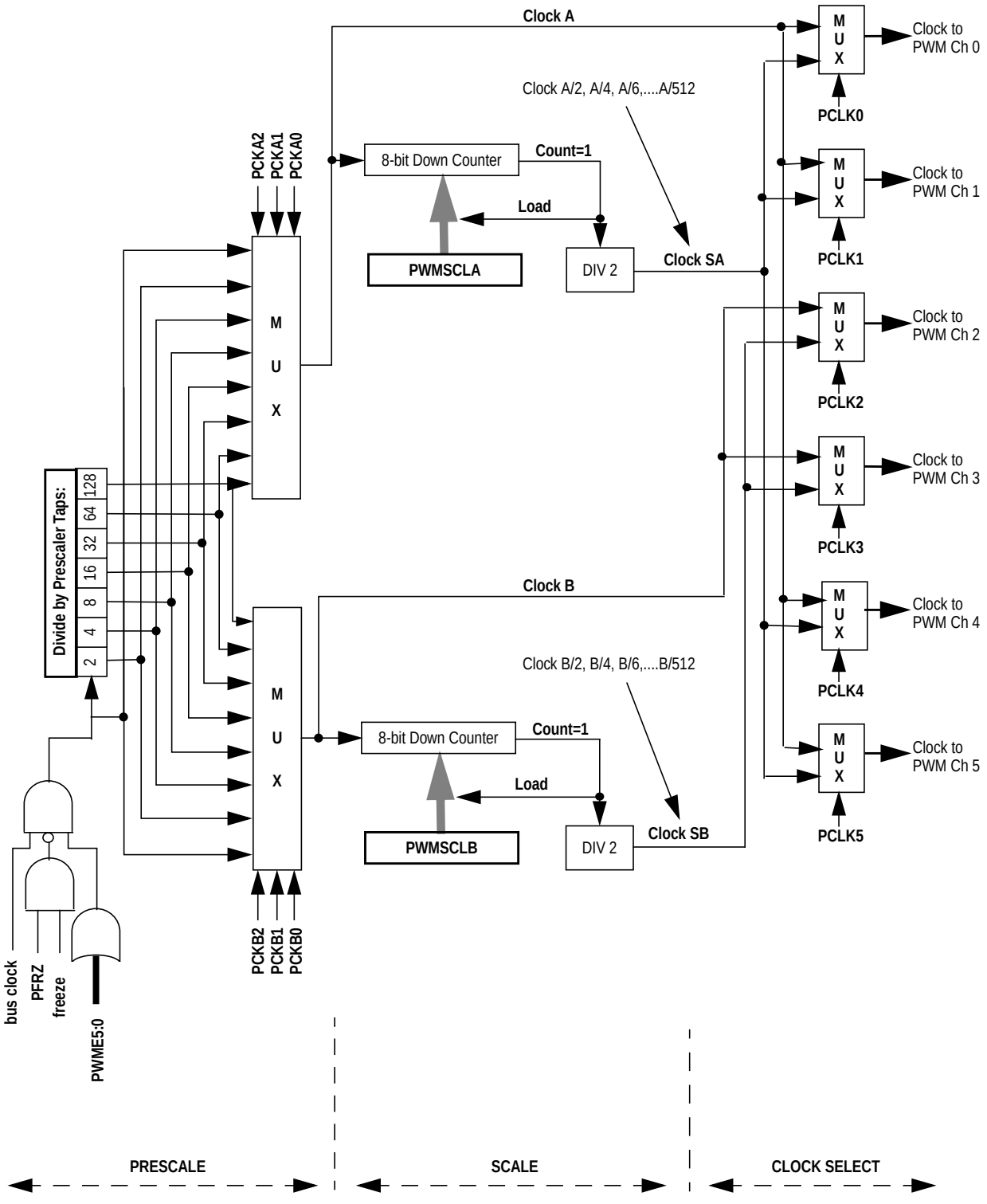

Figure 4-1 PWM Clock Select Block Diagram

4.1.1 Prescale

The input clock to the PWM prescaler is the bus clock. It can be disabled whenever the part is in freeze mode by setting the PFRZ bit in the PWMCTL register. If this bit is set, whenever the MCU is in freeze mode the input clock to the prescaler is disabled. This is useful for emulation in order to freeze the PWM. The input clock can also be disabled when all six PWM channels are disabled (PWME5-0=0) This is useful for reducing power by disabling the prescale counter.

Clock A and clock B are scaled values of the input clock. The value is software selectable for both clock A and clock B and has options of 1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, or 1/128 times the bus clock. The value selected for clock A is determined by the PCKA2, PCKA1, PCKA0 bits in the PWMPRCLK register. The value selected for clock B is determined by the PCKB2, PCKB1, PCKB0 bits also in the PWMPRCLK register.

4.1.2 Clock Scale

The scaled A clock uses clock A as an input and divides it further with a user programmable value and then divides this by 2. The scaled B clock uses clock B as an input and divides it further with a user programmable value and then divides this by 2. The rates available for clock SA are software selectable to be clock A divided by 2, 4, 6, 8, ..., or 512 in increments of divide by 2. Similar rates are available for clock SB.

Clock A is used as an input to an 8-bit down counter. This down counter loads a user programmable scale value from the scale register (PWMSCLA). When the down counter reaches one, two things happen; a pulse is output and the 8-bit counter is re-loaded. The output signal from this circuit is further divided by two.

This gives a greater range with only a slight reduction in granularity. Clock SA equals Clock A divided by two times the value in the PWMSCLA register.

NOTE: *Clock SA = Clock A / (2 * PWMSCLA)*

When PWMSCLA = $00, PWMSCLA value is considered a full scale value of 256. Clock A is thus divided by 512.

Similarly, Clock B is used as an input to an 8-bit down counter followed by a divide by two producing clock SB. Thus, clock SB equals Clock B divided by two times the value in the PWMSCLB register.

NOTE: *Clock SB = Clock B / (2 * PWMSCLB)*

When PWMSCLB = $00, PWMSCLB value is considered a full scale value of 256. Clock B is thus divided by 512.

As an example, consider the case in which the user writes $FF into the PWMSCLA register. Clock A for this case will be bus clock divided by 4. A pulse will occur at a rate of once every 255x4 bus cycles. Passing this through the divide by two circuit produces a clock signal at a bus clock divided by 2040 rate. Similarly, a value of $01 in the PWMSCLA register when clock A is bus clock divided by 4 will produce a bus clock divided by 8 rate.

Writing to PWMSCLA or PWMSCLB causes the associated 8-bit down counter to be re-loaded. Otherwise, when changing rates the counter would have to count down to $01 before counting at the proper rate. Forcing the associated counter to re-load the scale register value every time PWMSCLA or PWMSCLB is written prevents this.

NOTE: *Writing to the scale registers while channels are operating can cause irregularities in the PWM outputs.*

4.1.3 Clock Select

Each PWM channel has the capability of selecting one of two clocks. For channels 0, 1, 4 and 5 the clock choices are clock A or clock SA. For channels 2 and 3 the choices are clock B or clock SB. The clock selection is done with the PCLKx control bits in the PWMCLK register.

NOTE: *Changing clock control bits while channels are operating can cause irregularities in the PWM outputs.*

4.2 PWM Channel Timers

The main part of the PWM module are the actual timers. Each of the timer channels has a counter, a period register and a duty register (each are 8-bit). The waveform output period is controlled by a match between the period register and the value in the counter. The duty is controlled by a match between the duty register and the counter value and causes the state of the output to change during the period. The starting polarity of the output is also selectable on a per channel basis. **Figure 4-2** shows a block diagram for PWM timer.

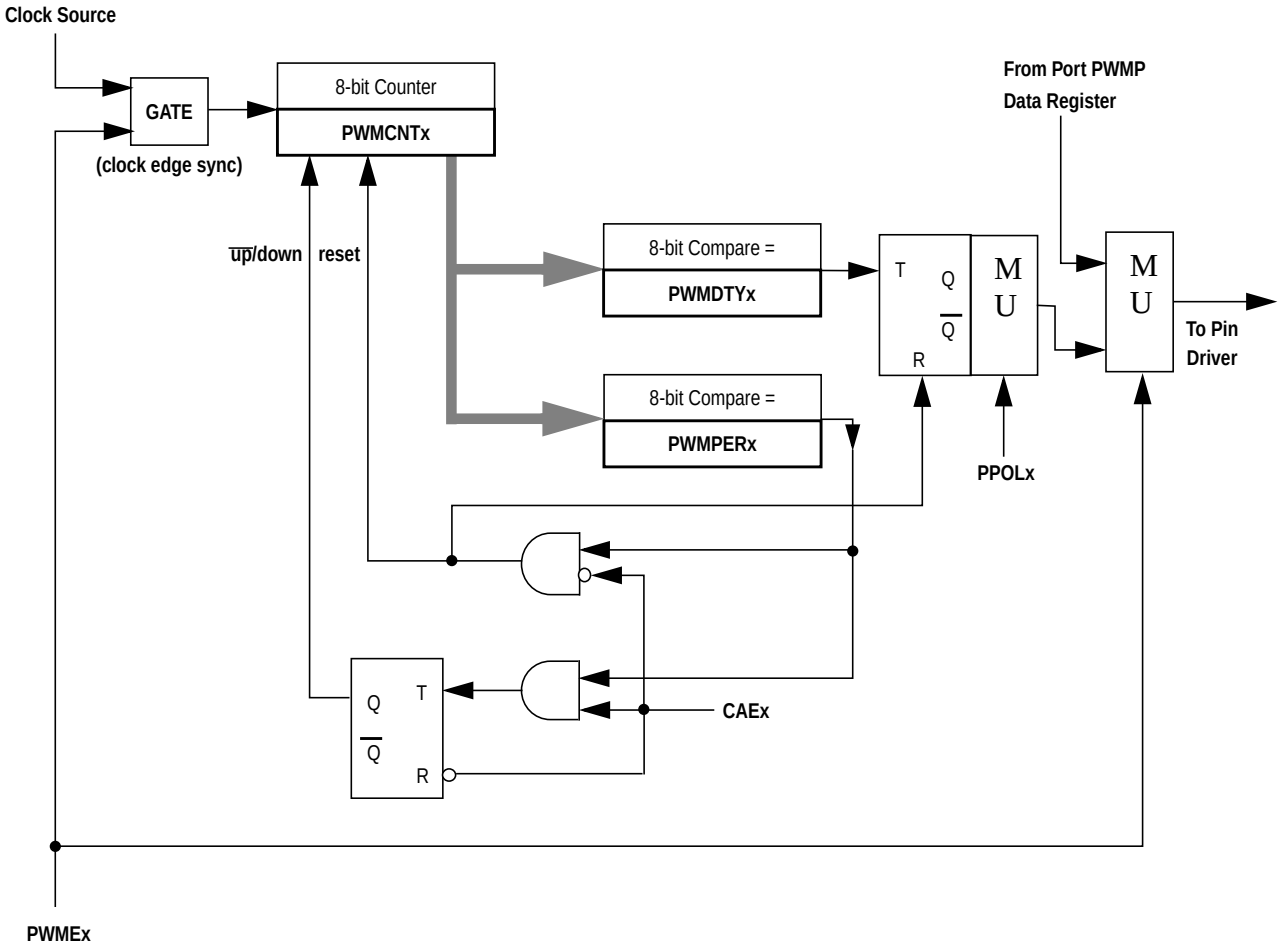

Figure 4-2 PWM Timer Channel Block Diagram

4.2.1 PWM Enable

Each PWM channel has an enable bit (PWMEx) to start its waveform output. When any of the PWMEx bits are set (PWMEx=1), the associated PWM output signal is enabled immediately. However, the actual PWM waveform is not available on the associated PWM output until its clock source begins its next cycle due to the synchronization of PWMEx and the clock source. An exception to this is when channels are concatenated. Refer to **4.2.7 PWM 16-Bit Functions** for more detail.

NOTE: *The first PWM cycle after enabling the channel can be irregular.*

On the front end of the PWM timer, the clock is enabled to the PWM circuit by the PWMEx bit being high. There is an edge-synchronizing circuit to guarantee that the clock will only be enabled or disabled at an edge. When the channel is disabled (PWMEx=0), the counter for the channel does not count.

4.2.2 PWM Polarity

Each channel has a polarity bit to allow starting a waveform cycle with a high or low signal. This is shown on the block diagram as a mux select of either the Q output or the $\overline{Q}$ output of the PWM output flip flop. When one of the bits in the PWMPOL register is set, the associated PWM channel output is high at the beginning of the waveform, then goes low when the duty count is reached. Conversely, if the polarity bit is zero, the output starts low and then goes high when the duty count is reached.

4.2.3 PWM Period and Duty

Dedicated period and duty registers exist for each channel and are double buffered so that if they change while the channel is enabled, the change will NOT take effect until one of the following occurs:

- The effective period ends
- The counter is written (counter resets to $00)
- The channel is disabled

In this way, the output of the PWM will always be either the old waveform or the new waveform, not some variation in between. If the channel is not enabled, then writes to the period and duty registers will go directly to the latches as well as the buffer.

A change in duty or period can be forced into effect “immediately” by writing the new value to the duty and/or period registers and then writing to the counter. This forces the counter to reset and the new duty and/or period values to be latched. In addition, since the counter is readable it is possible to know where the count is with respect to the duty value and software can be used to make adjustments.

NOTE: *When forcing a new period or duty into effect immediately, an irregular PWM cycle can occur.*

NOTE: *Depending on the polarity bit, the duty registers will contain the count of either the high time or the low time.*

4.2.4 PWM Timer Counters

Each channel has a dedicated 8-bit up/down counter which runs at the rate of the selected clock source (reference **Figure 4-1** for the available clock sources and rates). The counter compares to two registers, a duty register and a period register as shown in **Figure 4-2 PWM Timer Channel Block Diagram**. When the PWM counter matches the duty register the output flip-flop changes state causing the PWM waveform to also change state. A match between the PWM counter and the period register behaves differently depending on what output mode is selected as shown in **Figure 4-2** and described in **4.2.5 Left Aligned Outputs** and **4.2.6 Center Aligned Outputs**.

Each channel counter can be read at anytime without affecting the count or the operation of the PWM channel.

Any value written to the counter causes the counter to reset to $00, the counter direction to be set to up, the immediate load of both duty and period registers with values from the buffers, and the output to change

according to the polarity bit. When the channel is disabled (PWMEx=0), the counter stops. When a channel becomes enabled (PWMEx=1), the associated PWM counter continues from the count in the PWMCNTx register. This allows the waveform to continue where it left off when the channel is re-enabled. When the channel is disabled, writing “0” to the period register will cause the counter to reset on the next selected clock.

NOTE: *If the user wants to start a new “clean” PWM waveform without any “history” from the old waveform, the user must write to channel counter (PWMCNTx) prior to enabling the PWM channel (PWMEx=1).*

Generally, writes to the counter are done prior to enabling a channel in order to start from a known state. However, writing a counter can also be done while the PWM channel is enabled (counting). The effect is similar to writing the counter when the channel is disabled except that the new period is started immediately with the output set according to the polarity bit.

NOTE: *Writing to the counter while the channel is enabled can cause an irregular PWM cycle to occur.*

The counter is cleared at the end of the effective period (see **4.2.5 Left Aligned Outputs** and **4.2.6 Center Aligned Outputs** for more details).

Table 4-1 PWM Timer Counter Conditions

Counter Clears ($00)	Counter Counts	Counter Stops
When PWMCNTx register written to any value	When PWM channel is enabled (PWMEx=1). Counts from last value in PWMCNTx.	When PWM channel is disabled (PWMEx=0)
Effective period ends		

4.2.5 Left Aligned Outputs

The PWM timer provides the choice of two types of outputs, Left Aligned or Center Aligned outputs. They are selected with the CAEx bits in the PWMCAE register. If the CAEx bit is cleared (CAEx=0), the corresponding PWM output will be left aligned.

In left aligned output mode, the 8-bit counter is configured as an up counter only. It compares to two registers, a duty register and a period register as shown in the block diagram in **Figure 4-2**. When the PWM counter matches the duty register the output flip-flop changes state causing the PWM waveform to also change state. A match between the PWM counter and the period register resets the counter and the output flip-flop as shown in **Figure 4-2** as well as performing a load from the double buffer period and duty register to the associated registers as described in **4.2.3 PWM Period and Duty**. The counter counts from 0 to the value in the period register – 1.

NOTE: *Changing the PWM output mode from Left Aligned Output to Center Aligned Output (or vice versa) while channels are operating can cause irregularities in the*

PWM output. It is recommended to program the output mode before enabling the PWM channel.

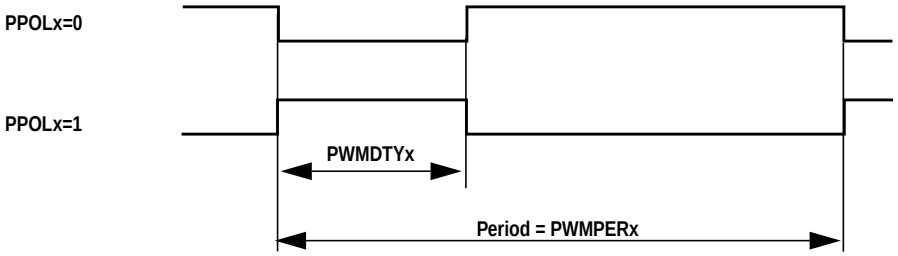

Figure 4-3 PWM Left Aligned Output Waveform

To calculate the output frequency in left aligned output mode for a particular channel, take the selected clock source frequency for the channel (A, B, SA, or SB) and divide it by the value in the period register for that channel.

- PWMx Frequency = Clock(A, B, SA, or SB) / PWMPERx
- PWMx Duty Cycle (high time as a% of period):
  - Polarity = 0 (PPOLx=0)
    Duty Cycle = [(PWMPERx-PWMDTYx)/PWMPERx] * 100%
  - Polarity = 1 (PPOLx=1)
    Duty Cycle = [PWMDTYx / PWMPERx] * 100%

As an example of a left aligned output, consider the following case:

Clock Source = bus clock, where bus clock=10MHz (100ns period)

PPOLx = 0

PWMPERx = 4

PWMDTYx = 1

PWMx Frequency = 10MHz/4 = 2.5MHz

PWMx Period = 400ns

PWMx Duty Cycle = 3/4 *100% = 75%

Shown below is the output waveform generated.

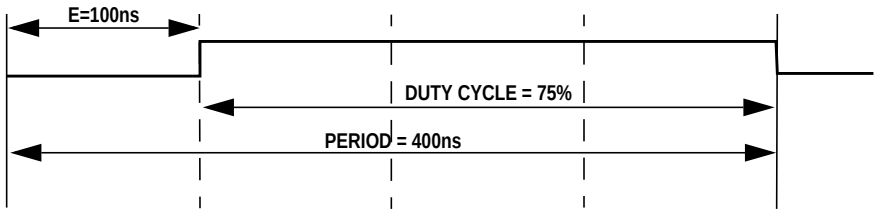

Figure 4-4 PWM Left Aligned Output Example Waveform

4.2.6 Center Aligned Outputs

For Center Aligned Output Mode selection, set the CAEx bit (CAEx=1) in the PWMCAE register and the corresponding PWM output will be center aligned.

The 8-bit counter operates as an up/down counter in this mode and is set to up whenever the counter is equal to $00. The counter compares to two registers, a duty register and a period register as shown in the block diagram in **Figure 4-2**. When the PWM counter matches the duty register the output flip-flop changes state causing the PWM waveform to also change state. A match between the PWM counter and the period register changes the counter direction from an up-count to a down-count. When the PWM counter decrements and matches the duty register again, the output flip-flop changes state causing the PWM output to also change state. When the PWM counter decrements and reaches zero, the counter direction changes from a down-count back to an up-count and a load from the double buffer period and duty registers to the associated registers is performed as described in **4.2.3 PWM Period and Duty**. The counter counts from 0 up to the value in the period register and then back down to 0. Thus the effective period is PWMPERx*2.

NOTE: *Changing the PWM output mode from Left Aligned Output to Center Aligned Output (or vice versa) while channels are operating can cause irregularities in the PWM output. It is recommended to program the output mode before enabling the PWM channel.*

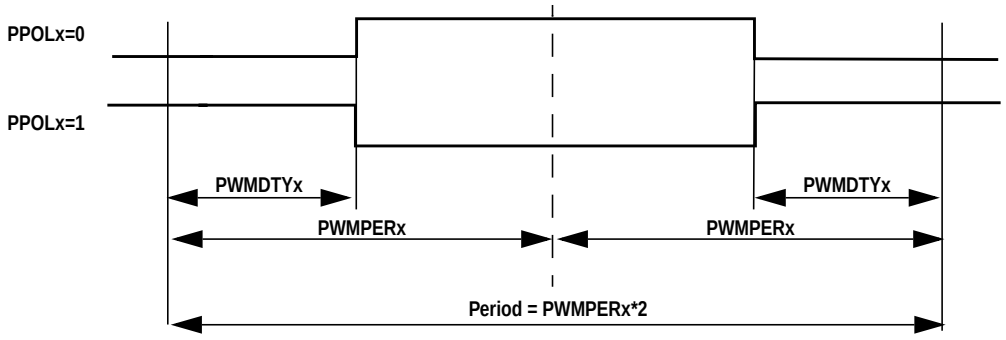

Figure 4-5 PWM Center Aligned Output Waveform

To calculate the output frequency in center aligned output mode for a particular channel, take the selected clock source frequency for the channel (A, B, SA, or SB) and divide it by twice the value in the period register for that channel.

- PWMx Frequency = Clock(A, B, SA, or SB) / (2*PWMPERx)
- PWMx Duty Cycle (high time as a% of period):
  - Polarity = 0 (PPOLx=0)
    Duty Cycle = [(PWMPERx-PWMDTYx)/PWMPERx] * 100%
  - Polarity = 1 (PPOLx=1)
    Duty Cycle = [PWMDTYx / PWMPERx] * 100%

As an example of a center aligned output, consider the following case:

Clock Source = bus clock, where bus clock=10MHz (100ns period)

PPOLx = 0

PWMPERx = 4

PWMDTYx = 1

PWMx Frequency = 10MHz/8 = 1.25MHz

PWMx Period = 800ns

PWMx Duty Cycle = 3/4 *100% = 75%

Shown below is the output waveform generated.

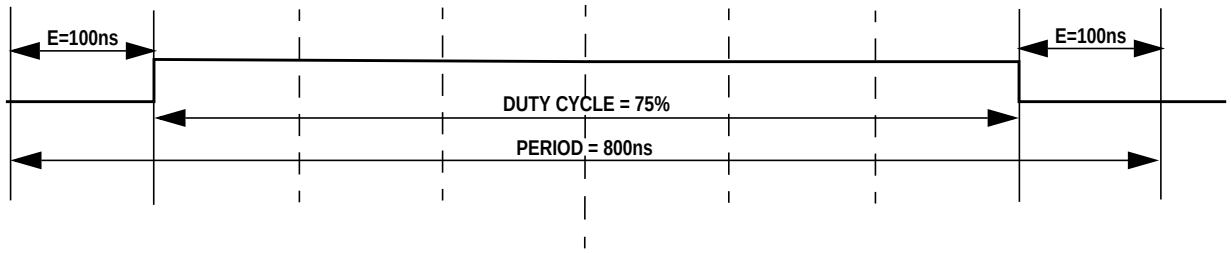

Figure 4-6 PWM Center Aligned Output Example Waveform

4.2.7 PWM 16-Bit Functions

The PWM timer also has the option of generating 6-channels of 8-bits or 3-channels of 16-bits for greater PWM resolution}. This 16-bit channel option is achieved through the concatenation of two 8-bit channels.

The PWMCTL register contains three control bits, each of which is used to concatenate a pair of PWM channels into one 16-bit channel. Channels 4 and 5 are concatenated with the CON45 bit, channels 2 and 3 are concatenated with the CON23 bit, and channels 0 and 1 are concatenated with the CON01 bit.

NOTE: *Change these bits only when both corresponding channels are disabled.*

When channels 4 and 5 are concatenated, channel 4 registers become the high order bytes of the double byte channel as shown in **Figure 4-7**. Similarly, when channels 2 and 3 are concatenated, channel 2 registers become the high order bytes of the double byte channel. When channels 0 and 1 are concatenated, channel 0 registers become the high order bytes of the double byte channel.

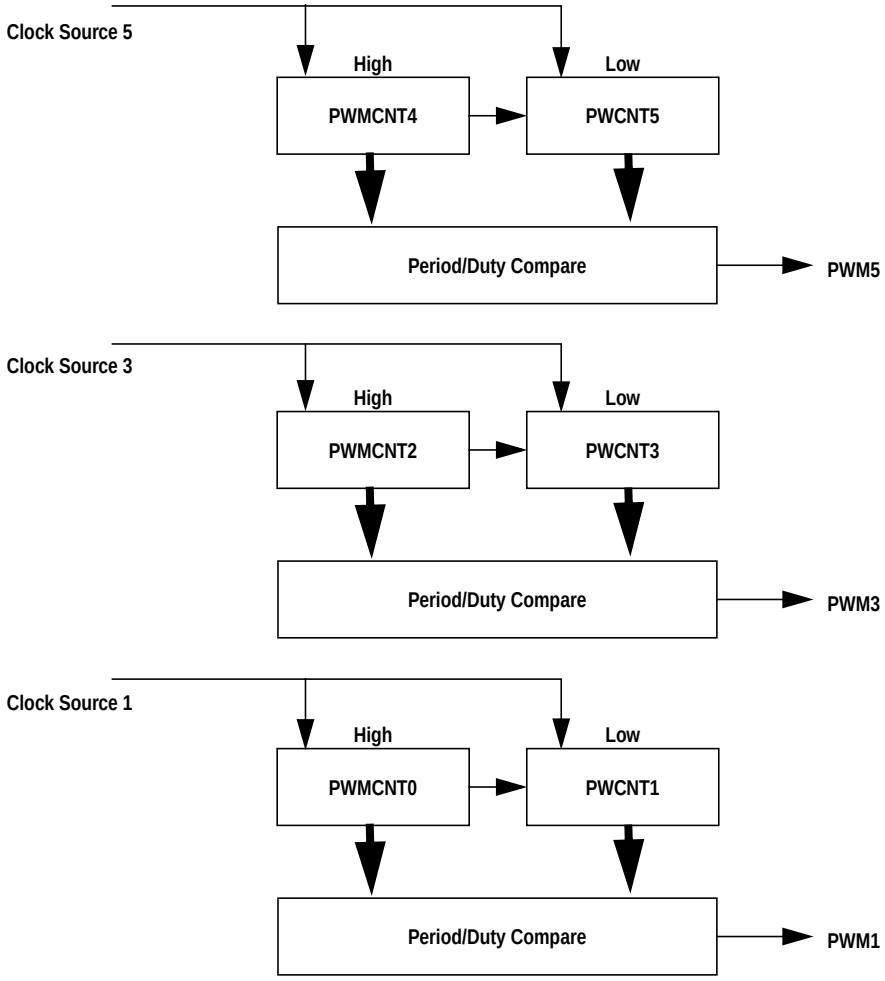

Figure 4-7 PWM 16-Bit Mode

When using the 16-bit concatenated mode, the clock source is determined by the low order 8-bit channel clock select control bits. That is channel 5 when channels 4 and 5 are concatenated, channel 3 when channels 2 and 3 are concatenated, and channel 1 when channels 0 and 1 are concatenated. The resulting PWM is output to the pins of the corresponding low order 8-bit channel as also shown in **Figure 4-7**. The polarity of the resulting PWM output is controlled by the PPOLx bit of the corresponding low order 8-bit channel as well.

Once concatenated mode is enabled (CONxx bits set in PWMCTL register) then enabling/disabling the corresponding 16-bit PWM channel is controlled by the low order PWMEx bit. In this case, the high order bytes PWMEx bits have no effect and their corresponding PWM output is disabled.

In concatenated mode, writes to the 16-bit counter by using a 16-bit access or writes to either the low or high order byte of the counter will reset the 16-bit counter. Reads of the 16-bit counter must be made by 16-bit access to maintain data coherency.

Either left aligned or center aligned output mode can be used in concatenated mode and is controlled by the low order CAEx bit. The high order CAEx bit has no effect.

The table shown below is used to summarize which channels are used to set the various control bits when in 16-bit mode.

Table 4-2 16-bit Concatenation Mode Summary

CONxx	PWMEx	PPOLx	PCLKx	CAEx	PWMx OUTPUT
CON45	PWME5	PPOL5	PCLK5	CAE5	PWM5
CON23	PWME3	PPOL3	PCLK3	CAE3	PWM3
CON01	PWME1	PPOL1	PCLK1	CAE1	PWM1

4.2.8 PWM Boundary Cases

The following table summarizes the boundary conditions for the PWM regardless of the output mode (Left Aligned or Center Aligned) and 8-bit (normal) or 16-bit (concatenation):

Table 4-3 PWM Boundary Cases

PWMDTYx	PWMPERx	PPOLx	PWMx Output
$00 (indicates no duty)	>$00	1	Always Low
$00 (indicates no duty)	>$00	0	Always High
XX	$00[1] (indicates no period)	1	Always High
XX	$00[1] (indicates no period)	0	Always Low
>= PWMPERx	XX	1	Always High
>= PWMPERx	XX	0	Always Low

NOTES:
1. Counter=$00 and does not count.

Section 5 Resets

5.1 General

The reset state of each individual bit is listed within the Register Description section (see **Section 3 Memory Map and Register Definition**) which details the registers and their bit-fields. All special functions or modes which are initialized during or just following reset are described within this section.

- The 8-bit up/down counter is configured as an up counter out of reset.
- All the channels are disabled and all the counters don’t count.

Section 6 Interrupts

6.1 Interrupt Operation

The PWM_8B6C module has only one interrupt which is generated at the time of emergency shutdown, if the corresponding enable bit (PWMIE) is set. This bit is the enable for the interrupt. The interrupt flag PWMIF is set whenever the input level of the PWM5 channel changes while PWM5ENA=1 or when PWMENA is being asserted while the level at PWM5 is active.

A description of the registers involved and affected due to this interrupt is explained in **Section 3.3.15 PWM Shutdown Register (PWMSDN)**.

User Guide End Sheet

FINAL PAGE OF
52
PAGES