



---

**eSi-PWM**

---

---

# 1 Contents

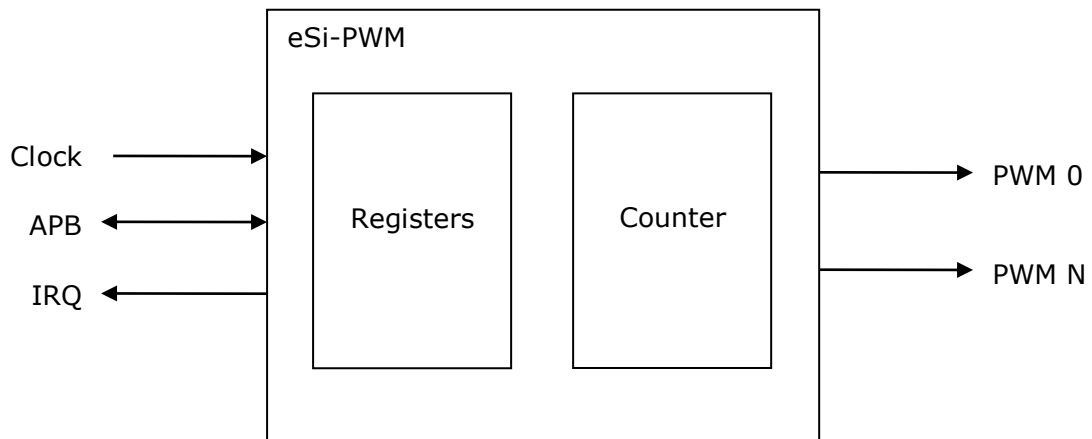
---

1	Contents _____	2
2	Overview _____	3
3	Hardware Interface _____	4
4	Software Interface _____	5
4.1	Register Map _____	5
4.2	Interrupts _____	6

## 2 Overview

The eSi-PWM is a pulse width modulation waveform generator. It has the following features:

- Configurable number of PWM channels.
- Runtime programmable duty cycle from 0% to 100% and period, with configurable precision.
- AMBA 3 APB slave interface.



**Figure 1: eSi-PWM**

### 3 Hardware Interface

<b>Module Name</b>	cpu_apb_pwm
<b>HDL</b>	Verilog
<b>Technology</b>	Generic
<b>Source Files</b>	cpu_apb_pwm.v

Port	Type	Description
bits	Integer	Specifies the number of bits in the counter
channels	Integer	Specifies the number of PWM channels

**Table 1: Parameters**

Port	Direction	Width	Description
clk	Input	1	Clock used for PWM counter. This must be active when <code>clk_cactive</code> is asserted. It must be synchronous to <code>pclk</code> , although can be at a lower frequency.
pclk	Input	1	APB clock. This must be active when <code>pclk_cactive</code> is asserted.
presetn	Input	1	APB reset, active-low
paddr	Input	8	APB address
psel	Input	1	APB slave select
penable	Input	1	APB enable
pwrite	Input	1	APB write
pwdata	Input	16	APB write data
clk_cactive	Output	1	Clock active (for <code>clk</code> )
pclk_cactive	Output	1	Clock active (for <code>pclk</code> )
pready	Output	1	APB ready
prdata	Output	16	APB read data
pslverr	Output	1	APB slave error
interrupt_n	Output	1	Interrupt request, active-low
pwm	Output	channels	PWM outputs

**Table 2: I/O Ports**

For complete details of the APB signals, please refer to the AMBA 3 APB Protocol v1.0 Specification available at <http://www.arm.com/products/solutions/AMBAHomePage.html>

## 4 Software Interface

### 4.1 Register Map

Register	Address offset	Access	Description
status	0x00	R/W	Status register
control	0x04	R/W	Control register
wrap_comparator	0x08	R/W	Wrap comparator
comparator[N]	0x10+4*N	R/W	Comparator register for channel N

**Table 3: Register Map**

#### 4.1.1 Status Register

The status register contains a selection of flags that indicate the current status of the PWM. To clear a bit in the status register, write a 1 to it. Writing 0 will leave it unchanged.

	1	0
	WO	W

**Figure 2: Format of the status register**

Register	Values	Description
W	0 - No wrap 1 - Wrapped	Wrapped flag. Indicates whether the counter has wrapped
WO	0 - No wrap overflow 1 - Wrap overflow	Wrapped overflow flag. Indicates if the W flag was set when the counter wrapped

**Table 4: Fields of the status register**

#### 4.1.2 Control Register

The control register contains a selection of flags that control the operation of the PWM.

	1	0
	WIE	E

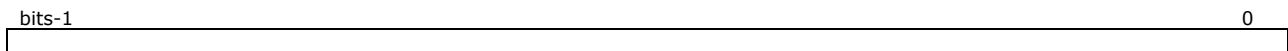
**Figure 3: Format of the control register**

Register	Values	Description
E	0 - Disabled 1 - Enabled	Enables the counter
WIE	0 - Disabled 1 - Enabled	Wrap interrupt enable

**Table 5: Fields of the control register**

#### 4.1.3 Wrap Comparator

The wrap comparator contains the value that the counter will count up to, before wrapping to 0.

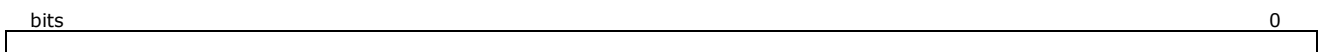


**Figure 4: Format of the `wrap_comparator` register**

#### 4.1.4 Comparator

A comparator register is implemented for each PWM channel. When the value in a comparator register equals the value in the counter, the corresponding PWM output changes from high to low. The PWM output will change from low to high when the counter wraps. The counter counts from 0 to the value held in the `wrap_comparator` register.

In order to achieve a duty cycle of 0%, the comparator should be initialised to 0. To allow a duty cycle of 100% to be achieved, the comparator registers are 1 bit wider than the counter register. Therefore, to implement a duty cycle of 100%, set the comparator to a value 1 higher ( $2^{\text{bits}}$ ) than the maximum value that can be represented in the counter ( $2^{\text{bits}}-1$ ).



**Figure 5: Format of the `comparator` register**

## 4.2 Interrupts

The PWM supports a wrap interrupt. The wrap interrupt will be raised when the counter wraps to 0 and the `WIE` flag in the `control` register is set to 1. The wrap interrupt can be acknowledged by writing a 1 to the `status.W` flag.