PWM programming

Pulse Width Modulation Basics

- Modulation is the process of varying one or more properties of a periodic waveform, called the *carrier signal*, w.r.t the *modulation signal* that typically contains information to be transmitted.
- Pulse Width Modulation or PWM is a way to encode data such that it corresponds to the width of the pulse given a fixed frequency.
- It is also a way to control motors , power circuits , etc., using the ‘width’ of the pulse.
- PWM has numerous applications like Motion Control, Dimming, Encoding Analog Signal into its Digital form, in Power Regulation , etc.
- PWM is a simplest way of producing analog values using digital system

The PWM is based on the standard Timer block and inherits all of its features, although only the PWM function is pinned out on the LPC1768.

The Timer is designed to count cycles of the peripheral clock (PCLK) and optionally generate interrupts or perform other actions when specified timer values occur, based on seven match registers. The PWM function is in addition to these features, and is based on match register events.

PWM can be used for more applications. For instance, multi-phase motor control typically requires three non-overlapping PWM outputs with individual control of all three pulse widths and positions.

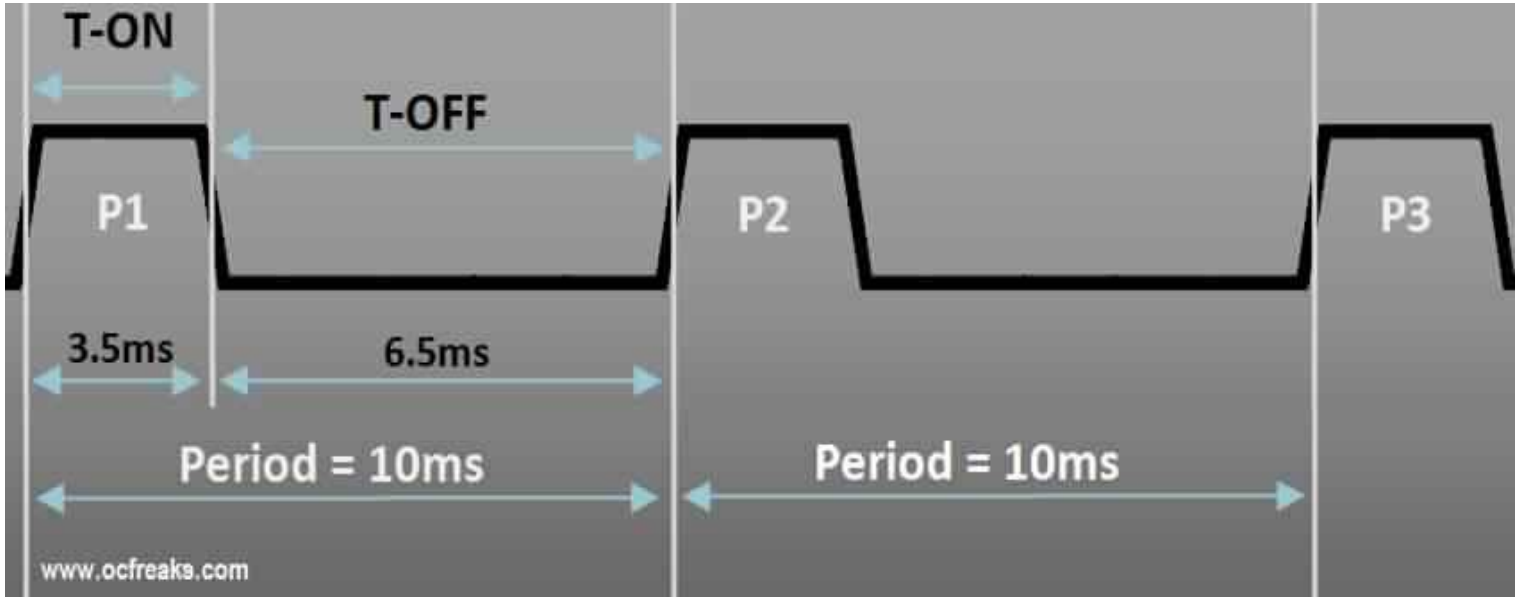
T-ON , T-OFF & DutyCycle :

$$f = \frac{1}{T}$$

Here,
f= Frequency
T= Time period

$$\text{Duty Cycle} = \frac{\text{T-ON}}{\text{T-ON + T-OFF}}$$

$$\text{Duty Cycle \%} = \frac{\text{T-ON}}{\text{T-ON + T-OFF}} \text{ X100}$$



- **Period i.e T = 10ms**
- **Frequency = 100hz**
- **T-ON = 3.5ms**
- **T-OFF = 6.5ms**
- **DutyCyle = 35%**

PWM Edges

- A PWM signal contains 2 types of Edges
- Leading Edge
- Trailing Edge

Types of PWM

PWM Signal can be Classified in Different ways.

1) Single Edge PWM

2) Double Edge PWM

Single Edge PWM

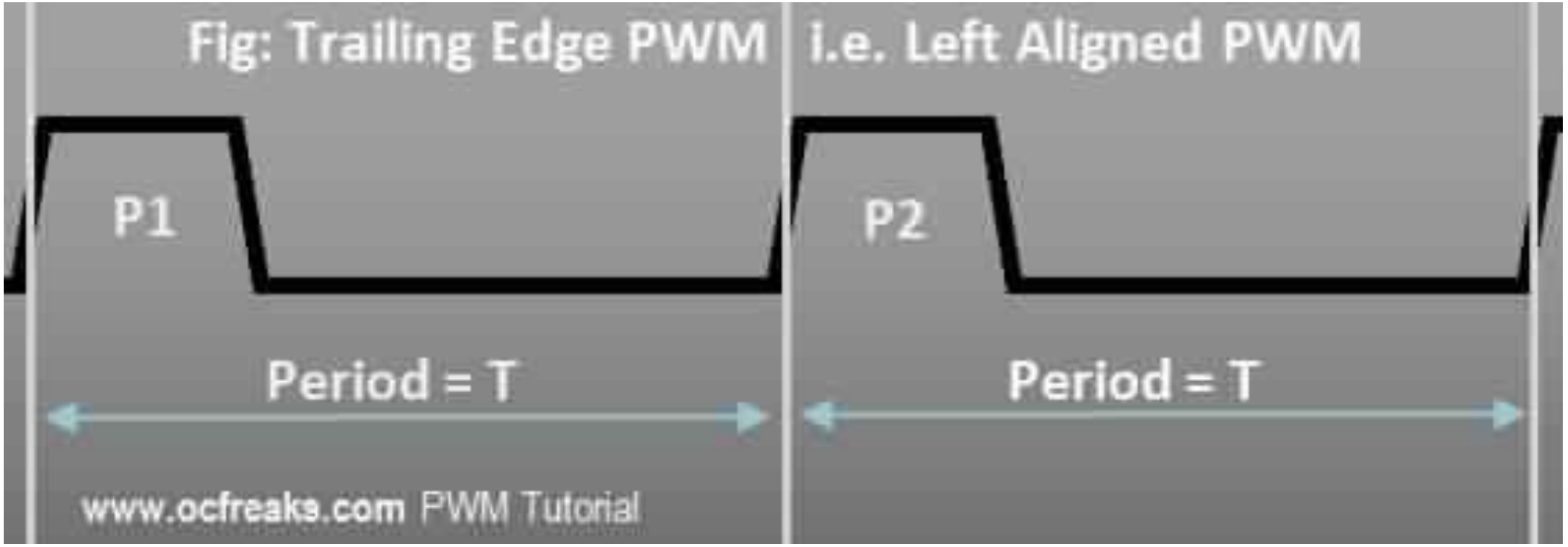
Pulse can either at the Beginning or the End of the Period

➔ **Single Edge PWM can be** further Classified

- **Leading Edge(Right Aligned) PWM**
- **Trailing Edge(Left Aligned) PWM**.

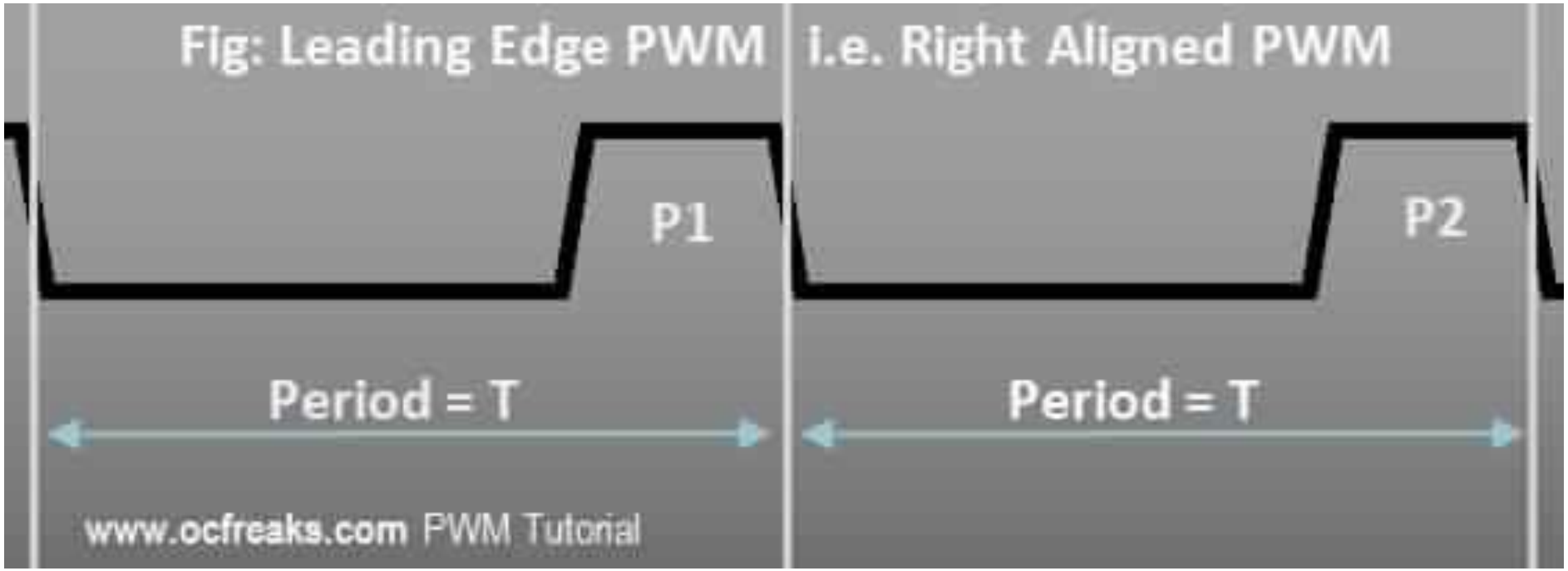
Trailing Edge PWM

Leading Edge is fixed at the Beginning of a Period and the **Trailing Edge is Modulated i.e. Varied**.



Leading Edge PWM

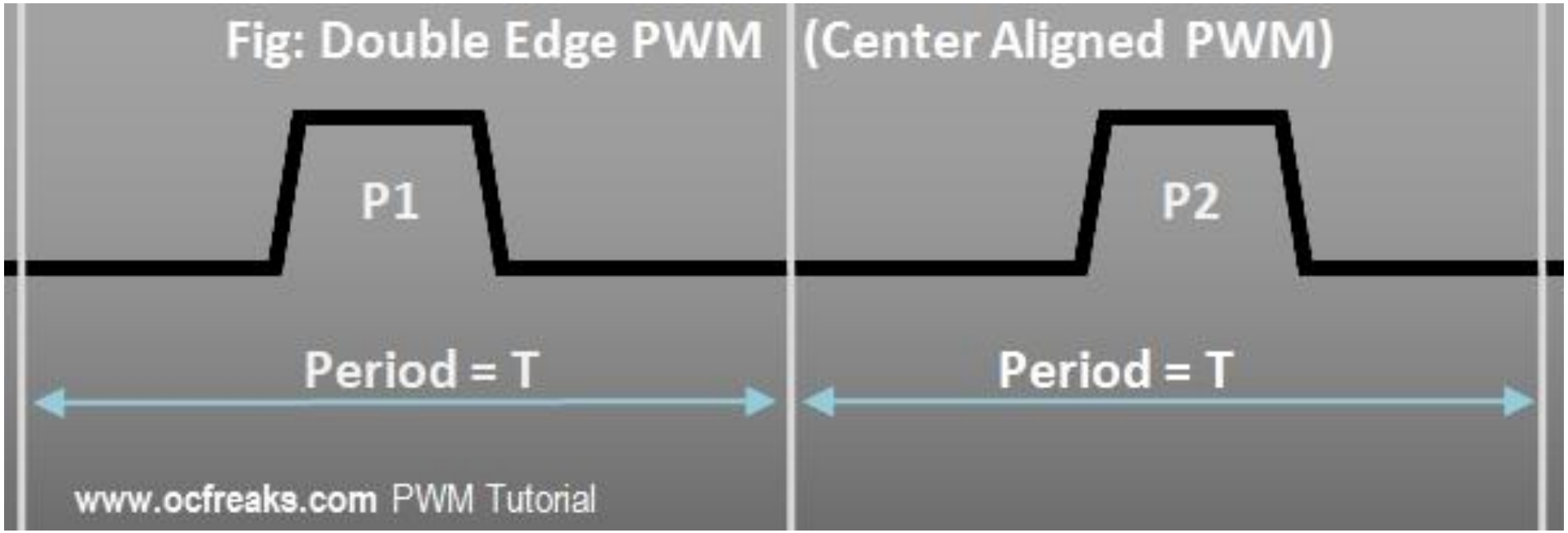
Trailing Edge is fixed at the End of a Period and the **Leading Edge is Modulated i.e. Varied**.

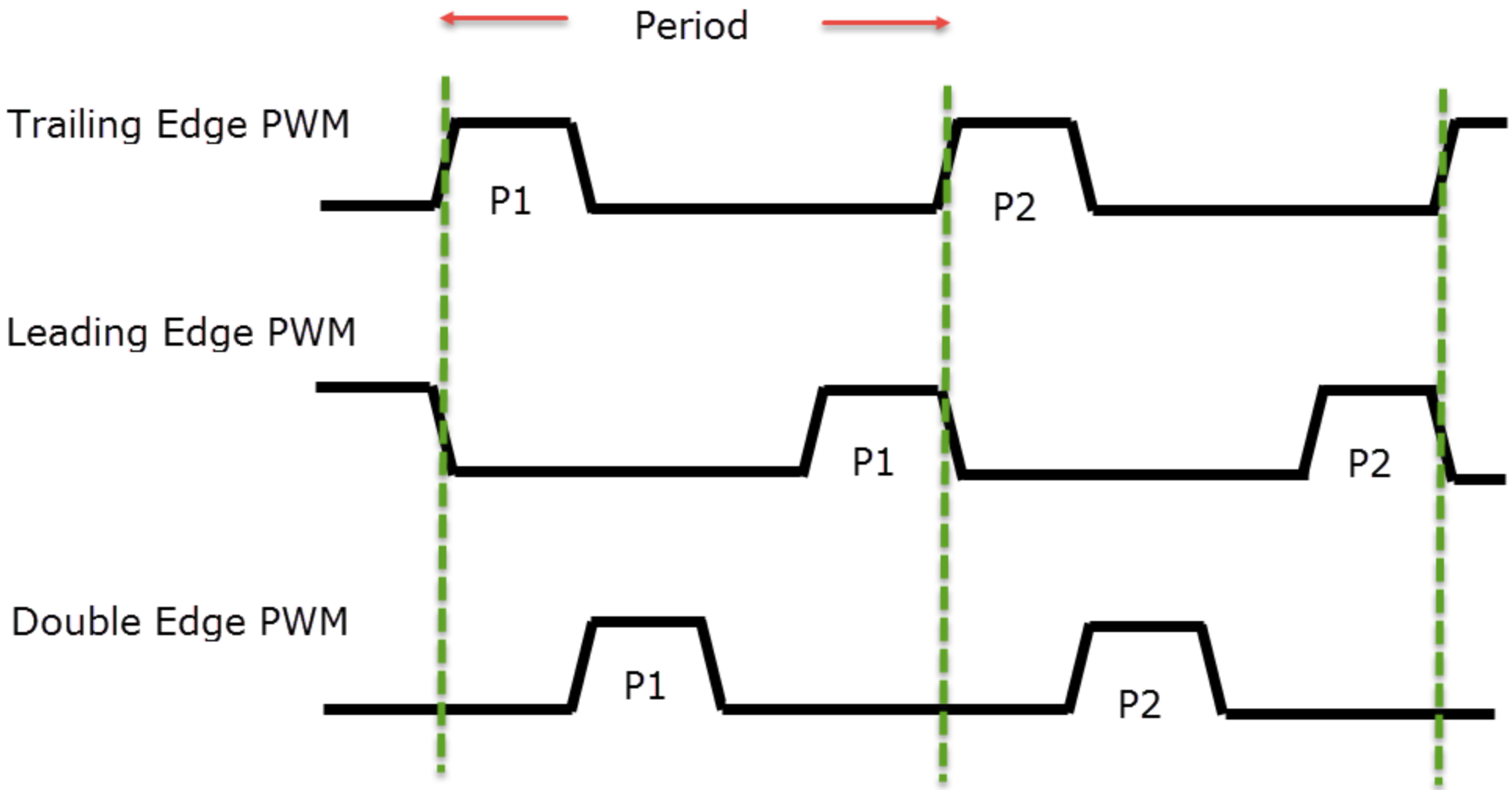


Double Edge PWM

Pulse can be Positioned anywhere within the Period. Here the Edges are Modulated or Varied.

In applications like Multi-Phase Motor control Double Edge PWM is used where the Pulse is Center Aligned to reduce Harmonics.




Period
Trailing Edge PWM
P1
P2
Leading Edge PWM
P1
P2
Double Edge PWM
P1
P2

PWM Voltage

The **Average Voltage** of a PWM Output depends on its Duty Cycle. Lower the Duty Cycle lower will be the Voltage and Vice-Versa.

$$\mathbf{V_{average} = DutyCycle \times V_H}$$

Where , V_H = Voltage for High State

In case the Low State Represents a Negative Voltage then the above equation can be generalized as follows:

$$\mathbf{V_{average} = (DutyCycle \times V_H) + ((1-D) \times V_L)}$$

Where , V_H = Voltage for High State & V_L = Voltage for Low State

ARM Cortex-M3 LPC1768 PWM Module

The LPC1768 Contains a single PWM Module called PWM1 and supports 2 types of PWM:

1) Single Edge PWM

2) Double Edge PWM

A PWM block, similar to a Timer block. It has a Timer Counter and an associated Prescale Register along with Match Registers. These work exactly the same way as in the case of Timers.

There are 7 match registers inside the PWM1 block.

- The first Match register PWM1MR0 is used to generate PWM period/frequency
- So we are left with 6 Match Registers PWM1MR1 to PWM1MR6
- The PWM1MR1 to PWM1MR6 generate 6 different Single Edge PWM signals or 3 different Double Edge PWM signals.
- Double edge PWM uses 2 match registers hence we can get only 3 different double edge outputs.

- Match Registers 1 to 6 (except 0) are pinned on LPC1768 i.e. the corresponding outputs are given to actual Pins on LPC1768 MCU.
- The PWM function must be selected for these Pins, using PINSELx Register, to get the PWM output. These pins are:

Output	PWM1.1	PWM1.2	PWM1.3	PWM1.4	PWM1.5	PWM1.6
Pin Name	P1.18 P2.0	P1.20 P2.1 P3.25	P1.21 P2.2 P3.26	P1.23 P2.3	P1.24 P2.4	P1.26 P2.5

- PWM1.1 output corresponds to PWM Match Register 1 i.e., PWM1MR1
- PWM1.2 output corresponds to PWM1MR2 , and so on.
- Also Match Register 0 i.e., PWM1MR0 is NOT pinned because it is used to generate the PWM Period/frequency.

PWM Channel	Single Edge Controlled		Double Edge Controlled	
	Set by	Reset by	Set by	Reset by
1	Match 0	Match 1	Match 0	Match 1
2	Match 0	Match 2	Match 1	Match 2
3	Match 0	Match 3	Match 2	Match 3
4	Match 0	Match 4	Match 3	Match 4
5	Match 0	Match 5	Match 4	Match 5
6	Match 0	Match 6	Match 5	Match 6

Table shows when the PWM is Set (Rising Edge) and Reset (Falling Edge) for different PWM channels using 7 Match Register.

PWM Rules –

The Rules for Single Edged PWM are:
1) All single edged PWM outputs will go high at the beginning of a PWM cycle unless their match value is 0.
2) Each PWM output will go low when its match value is reached.
Note: If no match occurs i.e Match value is greater than Period then the output will remain high!

Registers used in LPC176x PWM Programming

1. **PWM1TCR – PWM Timer Control Register**
2. **PWM1PR – PWM Prescale Register**
3. **PWM1MR0 – PWM1MR6 (Match Registers)**
4. **PWM1MCR – PWM Match Control Registers**
5. **PWM1IR – PWM Interrupt Register**
6. **PWMLER – Load Enable Register**
7. **PWM1PCR – PWM Control Register**

1. PWM1TCR – PWM Timer Control Register: Used to control the operations of Timer Counter inside the PWM block. Only Bits: 0, 1 & 3 are used rest are reserved.

- **Bit 0 :** Counter Enable
  - When 1 both PWM Timer counter and PWM Prescale counter are enabled.
  - When 0 both are disabled.
- **Bit 1 :** Counter Reset
  - When 1, both PWM Timer Counter and Prescale Counter are reset on next positive edge of PCLK.
- **Bit 3 :** PWM Enable
  - Enables PWM mode i.e., the PWM outputs.
- Other Bits : Reserved.

2) PWM1PR – PWM Prescale Register:

- PWMPR is used to control the resolution of the PWM outputs.
- The Timer Counter (TC) will increment every PWMPR+1 Peripheral Clock Cycles (PCLK).

3) PWM1MR0 – PWM1MR6 (Match Registers):

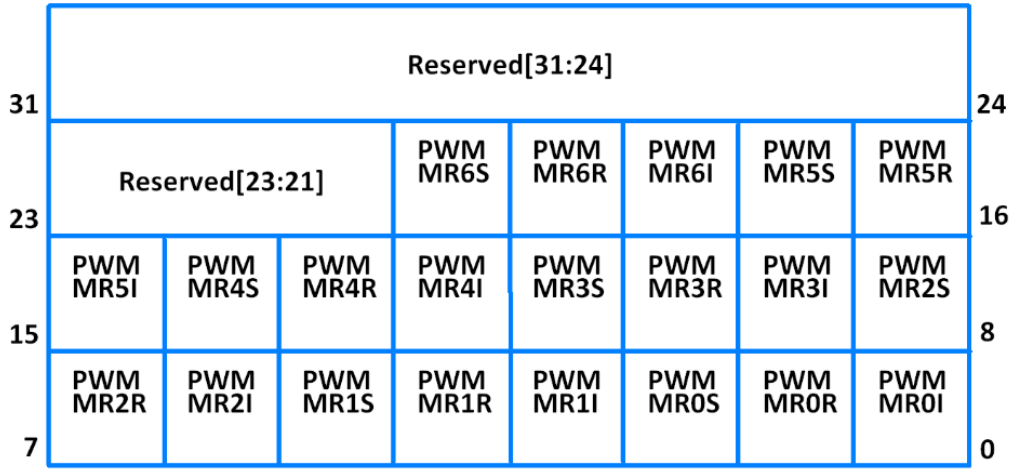
- The seven Match registers contain **Pulse Width Values i.e the Number of PWM1TC Ticks**.
- **PWM1MR0** holds the value of 1 complete cycle or the period.

•The values stored in these registers are continuously compared with the PWM Timer Counter value.

•When the two values are equal, the timer can be reset or stop or an interrupt may be generated depending on values set in PWMMCR.

4) PWM1MCR – PWM Match Control Registers:

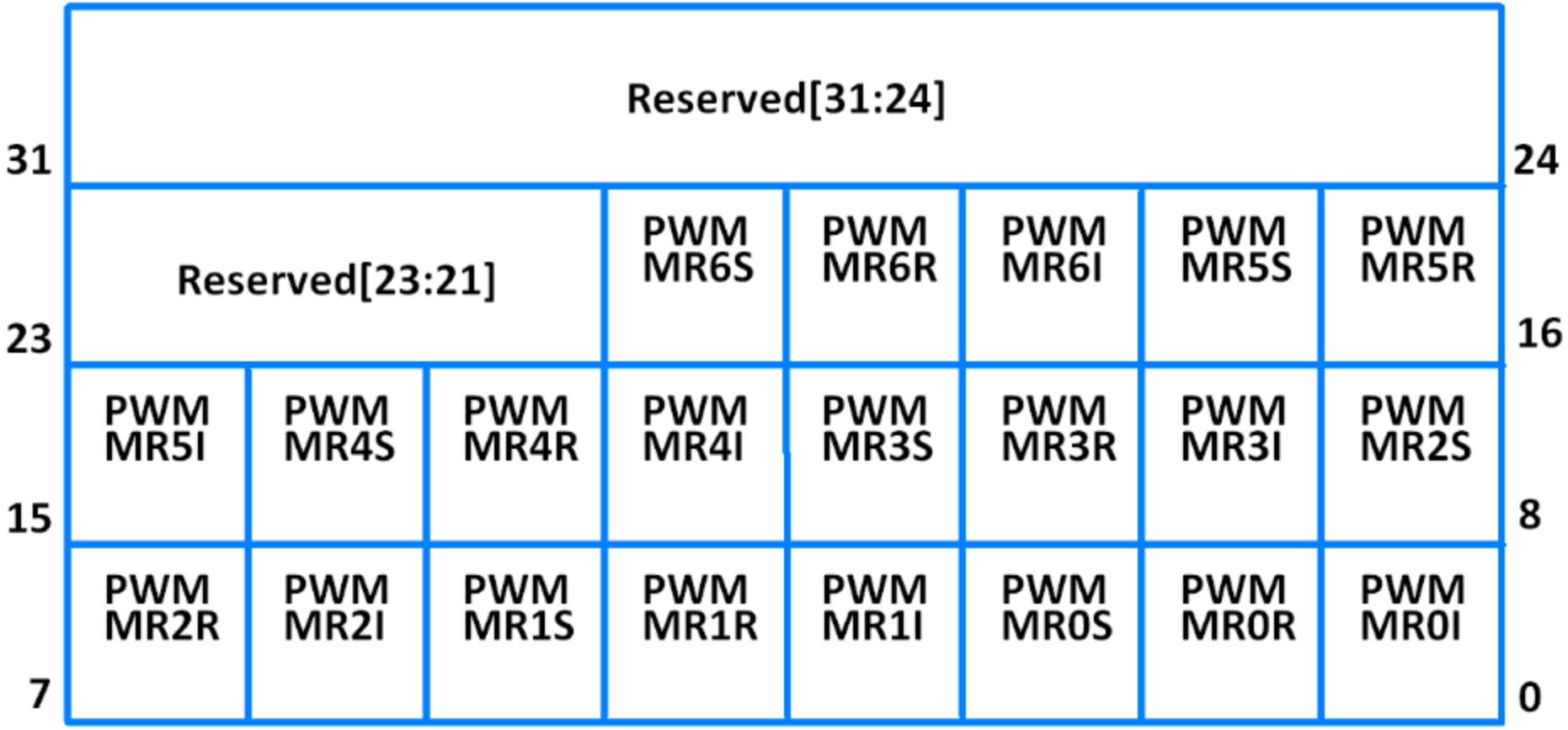
- The PWMMCR is used to specify what operations can be done when the value in a particular MR equals the value in TC.
- For each MR we have 3 options : Either
  - Either generate an Interrupt ,
  - Or Reset the TC,
  - Or Stop, which stops the counters and disables PWM
- Hence this register is divided into group of 3 bits.
  - The first 3 bits are for Match Register 0 i.e PWMMR0
  - Next 3 for PWMMR1 , and so on.



Let us see one such set of bits for PWM1MR0

Bit 0	PWMMR0I	If this bit is set to 1, then an interrupt is generated when PWMMR0 matches with PWMTC.
Bit 1	PWMMR0R	If this bit is set to 1, then PWMTC is reset when PWMMR0 matches with PWMTC.
Bit 2	PWMMR0S	If this bit is set to 1, then PWMTC and PWMPC is stopped and PWMTCR[0] is set to 0, when PWMMR0 matches with PWMTC.

Similarly, Bits [5:3] for PWMMR1, Bits [8:6] for PWMMR2 etc.



5) PWM1IR – PWM Interrupt Register:

- If an interrupt is generated by any of the Match Register then the corresponding bit in PWM1IR becomes high.
- Otherwise, the bit will be LOW.
- Writing a 1 to a bit in this register clears that interrupt.

1) Bits 0,1,2,3 are for PWM1MR0, PWM1MR1, PWM1MR2, PWM1MR3 respectively and
2) Bits 8,9,10 are for PWM1MR4 , PWM1MR5 , PWM1MR6 respectively.
Other bits are reserved.

6) PWMLER – Load Enable Register:

- Used to control the way by which Match Registers are updated when PWM generation is active.
- When PWM mode is active and we apply new values to the Match Registers the new values won't get applied immediately.
- Instead the new value is written to a "Shadow Register".
  - Shadow Register can be thought of as a duplicate Match Register.
  - Each Match Register has a corresponding Shadow Register.
- The value in this Shadow Register is transferred to the actual MR:
  - when: PWM1TC is reset (i.e., at the beginning of the next period) and the corresponding Bit in PWM1LER is 1.

7	6	5	4	3	2	1	0
RESERVED	Enable PWM Match6 Latch	Enable PWM Match5 Latch	Enable PWM Match4 Latch	Enable PWM Match3 Latch	Enable PWM Match2 Latch	Enable PWM Match1 Latch	Enable PWM Match0 Latch

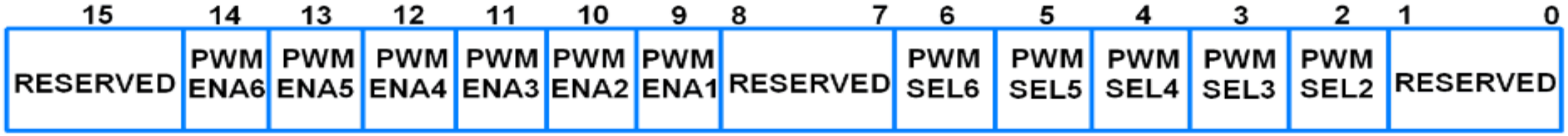
7) PWM1PCR – PWM Control Register: Used to enable a PWM Channel and also select the type of PWM i.e. either a Single Edge PWM or a Double Edge PWM.

1) **Bits 2 to 6**

- Bit 2 is called PWMSEL2 and when this bit 0, PWM2 output is single edged. When this bit is 1, PWM2 output is double edged.
- Similarly Bit 3 for PWM3, Bit 4 for PWM4 until Bit 6 for PWM6.

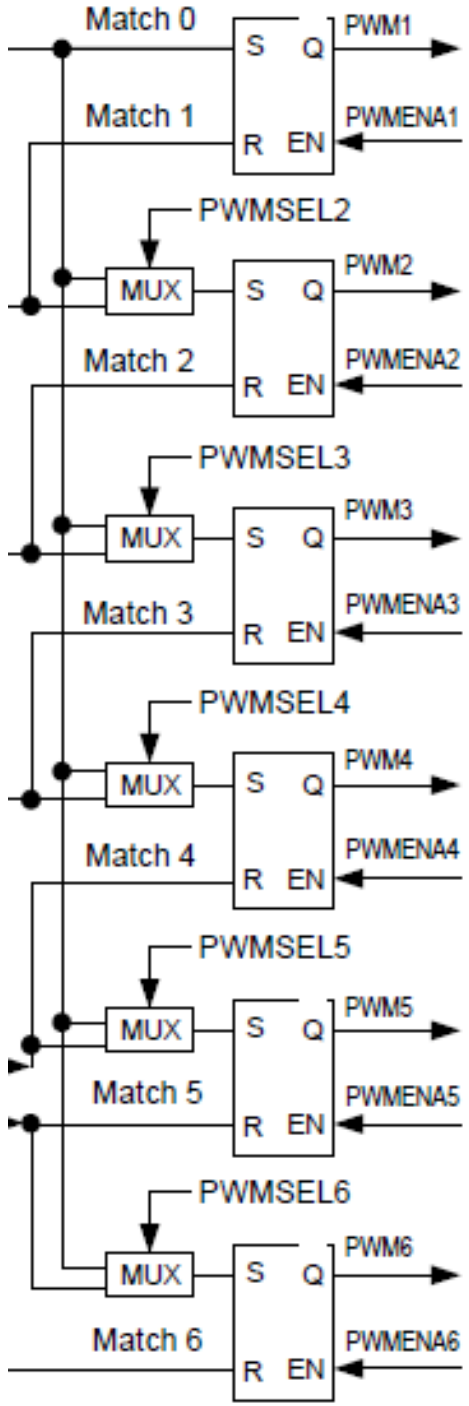
2) **Bits 9 to 14**

- Bits 9 in PWM1PCR is called PWMENA1 and when this bit is 1, PWM1 output is enabled, otherwise disabled.
- Similarly, Bit 10 for PWM2 output, Bit 11 for PWM3 output and so on until Bit 14 for PWM6 output.



How to generate PWM?

When the value for the cycle time or the period is loaded into MR0, The selected PWM output goes high. TC starts counting. When it matches with the value in the corresponding match register, the PWM output goes low and remains low until TC matches with the MR0. Then the next cycle starts. For the next cycle if the match register value corresponding to the PWM output is changed, the pulse width is changed.



Configuring and Initializing PWM Block

1.Select the PWM function for the PIN on which you need the PWM output using applicable `LPC_PINCON->PINSELx` register.
2.Select Single Edge or Double Edge Mode using `LPC_PWM1->PCR`. By default its Single Edge Mode.
3.Load the value to `LPC_PWM1->PR`.
4.Set the Value for PWM Period in `LPC_PWM1->MR0`.
5.Set the Values for other Match Registers i.e the Pulse Widths.
6.Set appropriate bit values in `LPC_PWM1->MCR`, like for e.g. resetting PWM1TC for PWMMR0 match and optionally generate interrupts if required
7.Set Load Enable Bits for the Match Registers that you've used. This is important!
8.Then Enable PWM outputs using `LPC_PWM1->PCR`.
9.Now Reset PWM Timer using `LPC_PWM1->TCR`.
10.Finally, Enable Timer Counter and PWM Mode using `LPC_PWM1->TCR`.

Ref:

http://www.ocfreaks.com/pulse-width-modulation-pwm-tutorial/

http://www.ocfreaks.com/lpc1768-pwm-programming-tutorial/

https://www.exploreembedded.com/wiki/LPC1768:_PWM