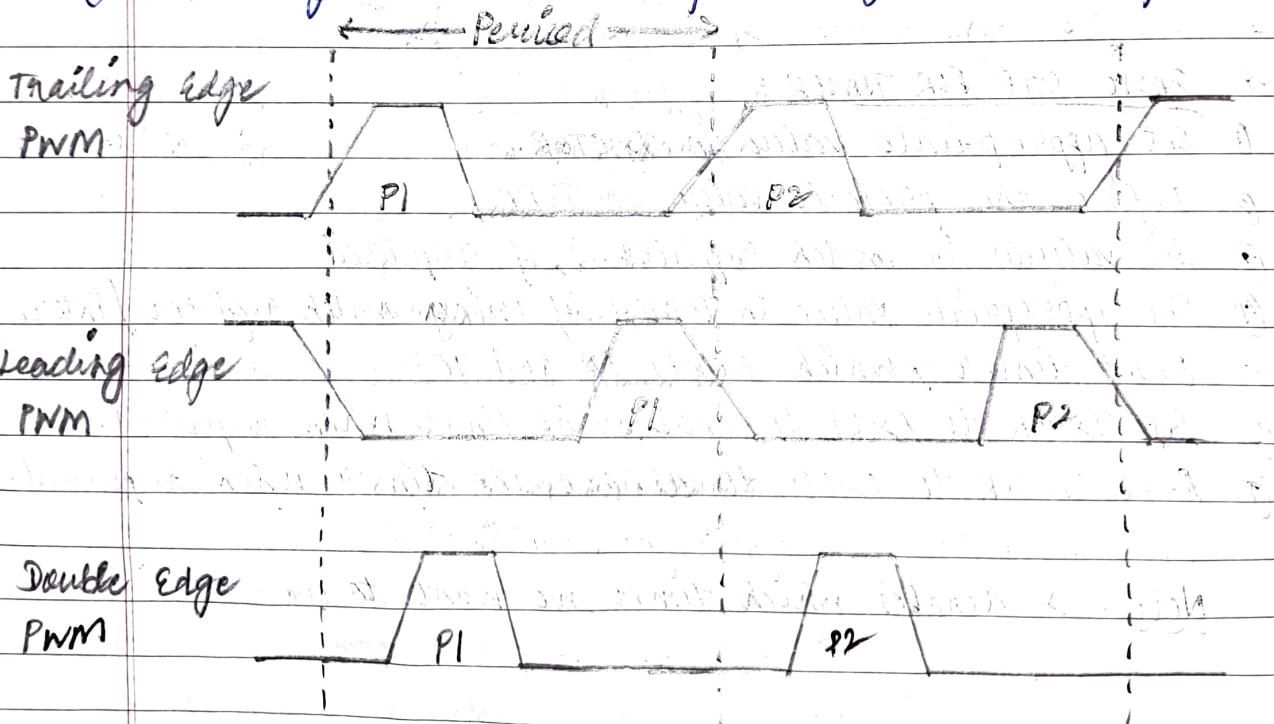


LECTURE 28 - NXP's LPC2148 PWM

PWM is used for generating pulses, which in turn is used for motor control. There are 6 PWM channels available in LPC2148 namely, P0.0 - PWM1, P0.1 - PWM3, P0.7 - PWM2, P0.8 - PWM4, P0.9 - PWM6 and P0.21 - PWM5.

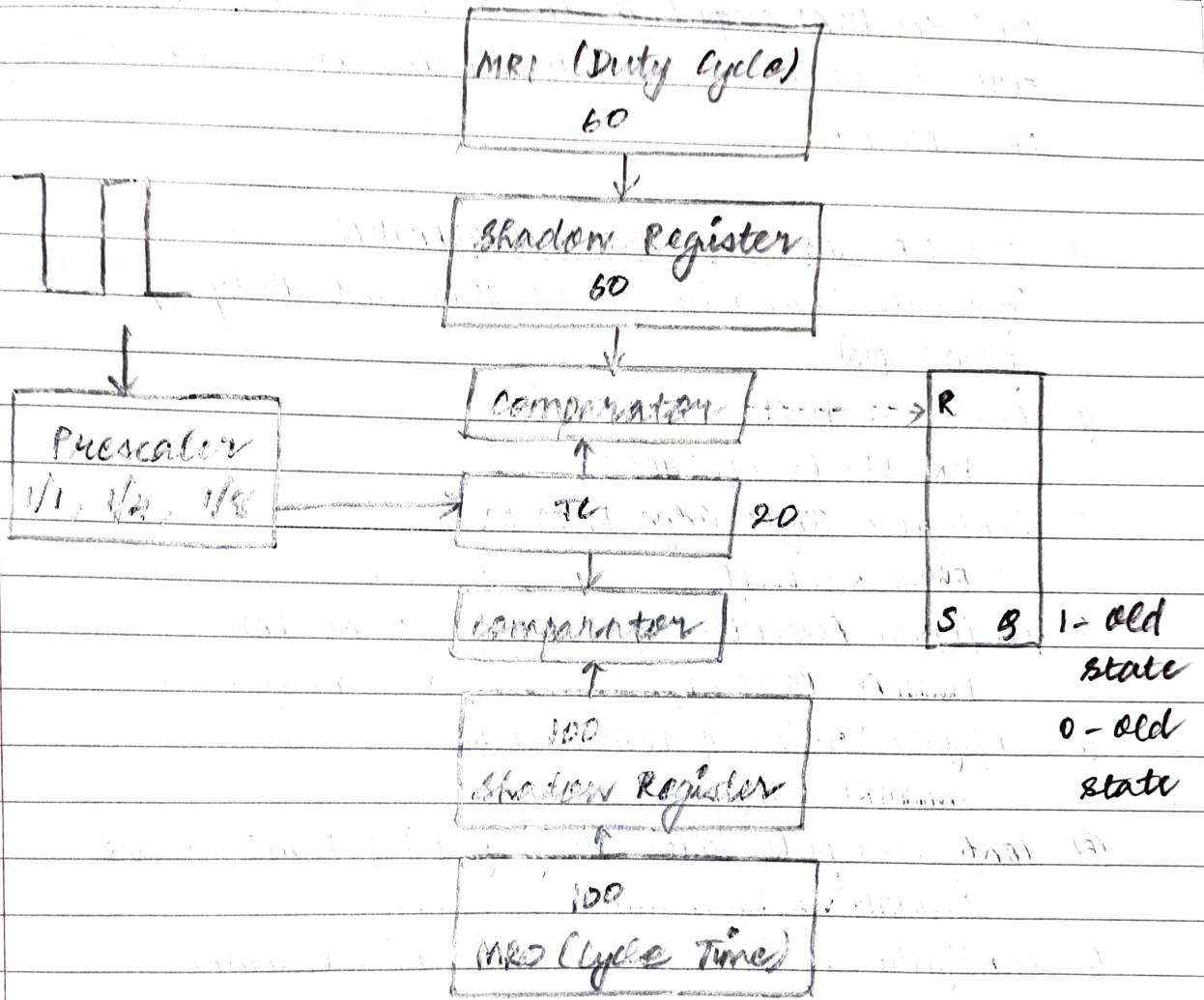
★ PWM TYPES:

- ① Single Edge PWM - Pulse at the beginning or the ending of the period, trailing edge PWM or leading edge PWM respectively.
- ② Double Edge PWM - Pulse can be placed anywhere in the period.

★ PWM OPERATION:

- IC matches MRL-6 (Duty Cycle)
 - PWM pin pulled Low.
- IC matches MRD (PWM Time Period)
 - Reset the IC value.
 - Pull the PWM pin High.

- Latches new Match register values
(Both Time Period - MRD and Duty cycle - MRI-6).



* PWM OPERATION - SUMMARY :

- ① The TC is being incremented as per the pre-scalar configuration. The PWM output is high as the TC is still less than duty cycle.
- ② TC is incremented to 40 and still the PWM pin is HIGH.
- ③ TC is incremented to 60 and it matches the Duty cycle (MRI=60).
- ④ Now the comparator (green) will trigger the R of SR latch and pulls the PWM output to zero ($q=0$). TC still continues to increment.
- ⑤ TC is incremented to 80 and PWM pin is low as $TC > \text{Duty cycle}$.

- ⑥ Now TC is 100 and it matches the cycle time ($MRD = 100$).
 ⑦ Now, the comparator 2 (red) will trigger the S of SR latch and pulls the PWM output to ONE ($g=1$). It also resets the TC to zero. It updates shadow buffers with new match values from MRD and MRI.

* STEPS FOR SINGLE EDGE PWM CONFIGURATION:

(Generate PWM with period 10ms and duty cycle 1ms in PWM5 pin)

- ① Configure P0.21 as PWM5 pin
 $PINSEL1 = 0x000000400;$
- ② Configure single edge PWM mode
 $PWMPCR = 0x00;$
- ③ Configure Resolution of PWM is set at 1ms
 $PWMPR = 60000 - 1;$
- ④ Configure period of PWM is 10ms
 $PWMMRD = 10;$
- ⑤ Configure pulse width (duty cycle) of PWM5 is 1ms
 $PWMMR5 = 1;$
- ⑥ Configure TC to reset on match with PWMMRD
 $PWMMCR = (1 \ll 1);$
- ⑦ Update match registers PWMMRD and PWMMR5
 $PWMCR = (1 \ll 5) | (1 \ll 0);$
- ⑧ Enable PWM5 output
 $PWMPCR = (1 \ll 3);$
- ⑨ Reset PWM TC and PWM PR
 $PWMTCR = (1 \ll 1);$
- ⑩ Enable PWM timer counters and PWM mode (also Counter Reset = 0)
 $PWMJCR = (1 \ll 0) | (1 \ll 3);$

* STEP 1 - PINSEL1 = 0x00000400;

Bit	Symbol	Value	Function
11:10	P0.21	00	GPIO Port 0.21
		01	PWM5
		10	Reserved or AD1.6
		11	Capture 1.3 (Timer 1)

* STEP 2 - PWMPCR = 0x00;

Table - PWM control Register (PWMPCR - address 0xED01 404C)

bit description

Bit	Symbol	Value	Description	Reset Value
1:0	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
2	PWMSEL2	1	selects double edge controlled mode for the PWM2 output.	0
		0	selects single edge controlled mode for PWM2.	
3	PWMSEL3	1	selects double edge controlled mode for the PWM3 output.	0
		0	selects single edge controlled mode for PWM3.	
4	PWMSEL4	1	selects double edge controlled mode for the PWM4 output.	0
		0	selects single edge controlled mode for PWM4.	
5	PWMSEL5	1	selects double edge controlled mode for the PWM5 output.	0

		o selects single edge controlled mode for PWM0.	
6	PWMSEL6	1	o selects double edge controlled mode for the PWM6 output.
		o	selects single edge controlled mode for PWM6.

* STEP 3 - PWMPR = 60000 - 1;

PWM block derives its clock from the peripheral clock (PCLK) - 60 MHz. Time taken for one PCLK cycle is $\frac{1}{60 \times 10^6}$.

So, time taken for 'x' PCLK cycles at 60MHz is $\frac{x}{60 \times 10^6}$.

If the prescale is considered, then $x = PC = PR + 1$.

If prescale register value is 59,

$$\text{Delay} = \frac{(59+1)}{60 \times 10^6} = 1 \mu\text{s.}$$

If prescale register value is 59999,

$$\text{Delay} = \frac{(59999+1)}{60 \times 10^6} = 1 \text{ms.}$$

* STEP 4 - PWMMRD = 10;

PWMMRD - PWM time period for all PWM channels

* STEP 5 - PWMMRS = 1;

PWMMRS - 6 - PWM duty cycle for each PWM channels respectively.

\overrightarrow{PDD}

STEP 6 - PWMMCR = (1<<1);

Table - Match Control Register (MCR, TIMER0 : TMCR - address 0xE000 4014 and TIMER1 : TIMCR - address 0xE000.8014) bit description

Bit	Symbol	Value	Description	Reset Value
0	PWMMR0I	1	Interrupt on PWMMRD: an interrupt is generated when PWMMRD matches the value in the PWMTC.	0
		0	This interrupt is disabled.	
1	PWMMR0R	1	Reset on PWMMRD: the PWMTC will be reset if PWMMRD matches it.	0
		0	This feature is disabled.	
2	PWMMR0S	1	Stop on PWMMRD: the PWMTC and PWMPC will be stopped and PWMTCR[0] will be set to 0 if PWMMRD matches the PWMTC.	0
		0	This feature is disabled.	
3	PWMMR1I	1	Interrupt on PWMMRI: an interrupt is generated when PWMMRI matches the value in the PWMTC.	0
		0	This interrupt is disabled.	
4	PWMMR1R	1	Reset on PWMMRI: the PWMTC will be reset if PWMMRI matches it.	0
		0	This feature is disabled.	
5	PWMMR1S	1	Stop on PWMMRI: the PWMTC and PWMPC will be stopped and PWMTCR[0] will be set to 0 if PWMMRI matches the PWMTC.	0
		0	This feature is disabled.	

* STEP 7 - PWMLER = $(1 \ll 5) | (1 \ll 0)$,

Table - PWM Latch Enable Register (PWMLER - address 0xE001 405D) bit description

bit	symbol	Description	Reset Value
0	Enable	Writing a one to this bit allows the last PWM Match value written to the PWM Match 0	0
0	Latch	register to be become effective when the timer is next reset by a PWM Match event	
1	Enable	Writing a one to this bit allows the last PWM Match value written to the PWM Match 1	0
1	Latch	register to be become effective when the timer is next reset by a PWM Match event	
2	Enable	Writing a one to this bit allows the last PWM Match value written to the PWM Match 2	0
2	Latch	register to be become effective when the timer is next reset by a PWM Match event	
3	Enable	Writing a one to this bit allows the last PWM Match value written to the PWM Match 3	0
3	Latch	register to be become effective when the timer is next reset by a PWM Match event	
4	Enable	Writing a one to this bit allows the last PWM Match value written to the PWM Match 4	0
4	Latch	register to be become effective when the timer is next reset by a PWM Match event	
5	Enable	Writing a one to this bit allows the last PWM Match value written to the PWM Match 5	0
5	Latch	register to be become effective when the timer is next reset by a PWM Match event	
6	Enable	Writing a one to this bit allows the last PWM Match value written to the PWM Match 6	0
6	Latch	register to be become effective when the timer is next reset by a PWM Match event	

* STEP 8 - PWMPCR = $(1 \ll 13)$:

Table - PWM Control Register (PWMPCR - address 0xE001 404C) bit registration

Bit	Symbol	Value	Description	Reset Value
9	PWMENAI	1	The PWM1 output enabled.	0
		0	The PWM1 output disabled.	
10	PWMENAZ	1	The PWM2 output enabled.	0
		0	The PWM2 output disabled.	
11	PWMENAS3	1	The PWM3 output enabled.	0
		0	The PWM3 output disabled.	
12	PWMENAS4	1	The PWM4 output enabled.	0
		0	The PWM4 output disabled.	
13	PWMENAS5	1	The PWM5 output enabled.	0
		0	The PWM5 output disabled.	
14	PWMENAS6	1	The PWM6 output enabled.	0
		0	The PWM6 output disabled.	

* STEP 9 - PWMTCR = (1<<1);

Table - PWM Timer Control Register (PWMTCR - address 0xEDD14004) bit description

Bit	Symbol	Description	Reset Value
0	Counter Enable	When one, the PWM Timer Counter and PWM Prescale Counter are enabled for counting. When zero, the counters are disabled.	0
1	Counter Reset	When one, the PWM Timer Counter and the PWM Prescale Counter are synchronously reset on the next positive edge of PCLK. The counters remain reset until TCR[1] is returned to zero.	0
2	-	Reserved, user software should not write ones to NA reserved bits. The value read from a reserved bit is not defined.	NA

3 PWM
Enable

When one, PWM mode is enabled. PWM mode causes shadow registers to operate in connection with the match registers. A program write to a Match register will not have an effect on the match result until the corresponding bit in PWMLTR has been set, followed by the occurrence of a PWM match 0 event. Note that the PWM match register that determines the PWM rate (PWM match 0) must be set up prior to the PWM being enabled. otherwise a match event will not occur to cause shadow register contents to become effective.

* STEP 10 - PWMTCR = (1<<0) | (K<3);