LPC2148 Timer/Counter

Introduction

Timer is a specific type of clock which is used to measure the time intervals. It provides/measures the time interval by counting the input clocks. Every timer needs a clock to work. We can provide/measure any time interval if we know the time of one clock period.

e.g. Let’s say we have 1 kHz input clock frequency for the timer unit, then,

We can calculate time of one clock period as,

Time of one clock period = 1 / clock frequency

= 1 / 1000

= 1milliSecond

i.e. 1000 clock counts provide a time interval of 1 second, and hence we can provide 1 second delay with these 1000 clock counts.

Now, once we have the time period of one clock, we can use this time period to generate delays that are integer multiples of it. We can also use the time period to measure the time interval between specific events of a received signal.

Counter is the unit which is similar to Timers but works in a reverse manner to the timers. It counts the external events or we can say external clock ticks. It is mostly used to measure frequency from the counts of clock ticks.

e.g. Let’s say Counter is measuring counts of external clock ticks, and frequently its count reaches 2000 in one second i.e. 2000 clock ticks/second.

Then, we can calculate external clock frequency as,

External clock frequency = count of clocks / one second

= 2000 / 1

= 2 kHz

Hence, we can measure such external clock/event frequencies using counter.

There are many applications for which we can use these timers and counters in real world.

LPC2148 Timer& Counter

LPC2148 has two 32-bit timers/counters:Timer0/Counter0 & Timer1/Counter1.

- LPC2148 Timer has input of peripheral clock (PCLK) or an external clock. It counts the clock from either of these clock sources for its operation.
- LPC2148 Timer/Counter can generate an interrupt signal at specified time value.
- LPC2148 has match registers that contain count value which is continuously compared with the value of the Timer register. When the value in the Timer register matches the value in the match register, specific action (timer reset, or timer stop, or generate an interrupt) is taken.
- For information about LPC2148 compare and match operation refer Timer Compare Mode**.**
- Also, LPC2148 has capture registers which can be used to capture the timer value on a specific external event on capture pins. For information about LPC2148 capture operation refer Input Capture Mode**.**

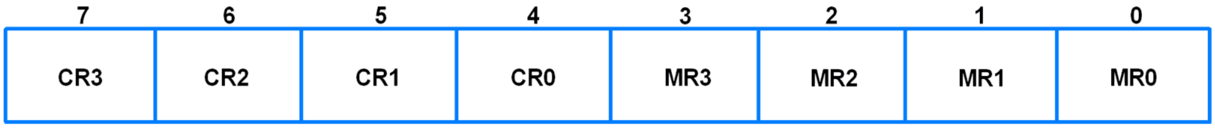
Here, we will see how to use timer 0 registers to configure the timer 0. Timer 1 can be used in a similar manner by using its corresponding registers.

Let’s see the various registers of timer 0.

Timer0 Registers

1. **T0IR (Timer0 Interrupt Register)**

- It is an 8-bit read-write register.
- Consists of 4 bits for match register interrupts and 4 bits for compare register interrupts.
- If interrupt is generated, then the corresponding bit in this register will be high, otherwise it will be low.
- Writing a 1 to any bit of this register will reset that interrupt. Writing a 0 has no effect.

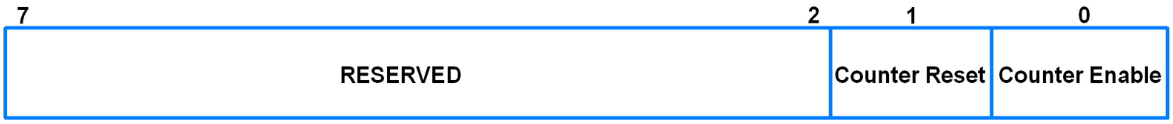


7	6	5	4	3	2	1	0
CR3	CR2	CR1	CR0	MR3	MR2	MR1	MR0

T0IR (Timer0 Interrupt Register)

2. **T0TCR (Timer0 Timer Control Register)**

- It is an 8-bit read-write register.
- It is used to control the operation of the timer counter.



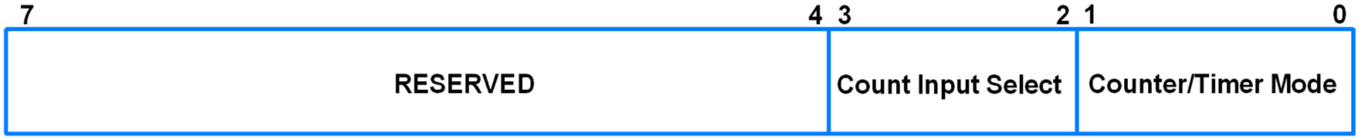
7 2	1	0
RESERVED	Counter Reset	Counter Enable

T0TCR (Timer0 Timer Control Register)

- **Bit 0 – Counter Enable**
  0 = Counters are disabled
  1 = Timer counter and Prescale counter are enabled for counting
- **Bit 1 – Counter Reset**
  0 = Counter not reset
  1 = Timer counter and Prescale counter are synchronously reset on next positive edge of PCLK

3. **T0CTCR (Timer0 Counter Control Register)**

- It is an 8-bit read-write register.



7 4	3 2	1 0
RESERVED	Count Input Select	Counter/Timer Mode

T0CTCR (Timer0 Counter Control Register)

- Used to select between timer counter mode.
- When in counter mode, it is used to select the pin and edges for counting.
- **Bits 1:0 – Counter/Timer Mode**
  This field selects which rising edges of PCLK can increment Timer's Prescale Counter (PC), or clear PC and increment Timer Counter (TC).
  00 = Timer Mode: Every rising edge of PCLK
  01 = Counter Mode: TC is incremented on rising edge on the capture input selected by Bits 3:2.
  10 = Counter Mode: TC is incremented on falling edge on the capture input selected by Bits 3:2
  01 = Counter Mode: TC is incremented on both edges on the capture input selected by Bits 3:2
- **Bits 3:2 – Count Input Select**
  When bits 1:0 in this register are not 00, these bits select which capture pin is sampled for clocking.
  00 = CAP0.0
  01 = CAP0.1
  10 = CAP0.2
  11 = CAP0.3
- **Note** : If counter mode is selected for a certain capture (CAP) input, then the corresponding 3 bits in the T0CCR register must be programmed as 000. Capture and/or interrupt can be selected for other CAP inputs.

1. **T0TC (Timer0 Timer Counter)**

· It is a 32-bit timer counter.

· It is incremented when the Prescale Counter (PC) reaches its maximum value held by Prescaler Register (PR).

Note : When TC overflow occurs, it does not generate any overflow interrupt. Alternatively, we can use match register to detect overflow event if needed.

4. **T0PR (Timer0 Prescale Register)**

- It is a 32-bit register.
- It holds the maximum value of the Prescale Counter.

5. **T0PC (Timer0 Prescale Counter Register)**

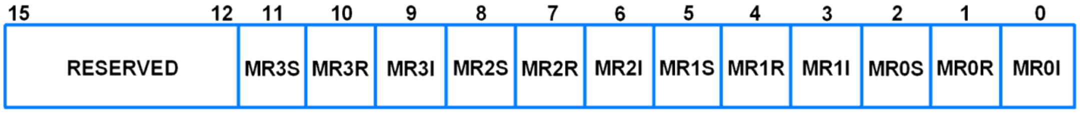
- It is a 32-bit register.
- It controls the division of PCLK by some constant value before it is applied to the Timer Counter.
- It is incremented on every PCLK.
- When it reaches the value in Prescale Register, the Timer Counter is incremented and Prescale Counter is reset on next PCLK.

6. **T0MR0-T0MR3 (Timer0 Match Registers)**

- These are 32-bit registers.
- The values stored in these registers are continuously compared with the Timer Counter value.
- When the two values are equal, the timer can be reset or stop or an interrupt may be generated. The T0MCR controls what action should be taken on a match.

7. **T0MCR (Timer0 Match Control Register)**

- It is a 16-bit register.
- It controls what action is to be taken on a match between the Match Registers and Timer Counter.



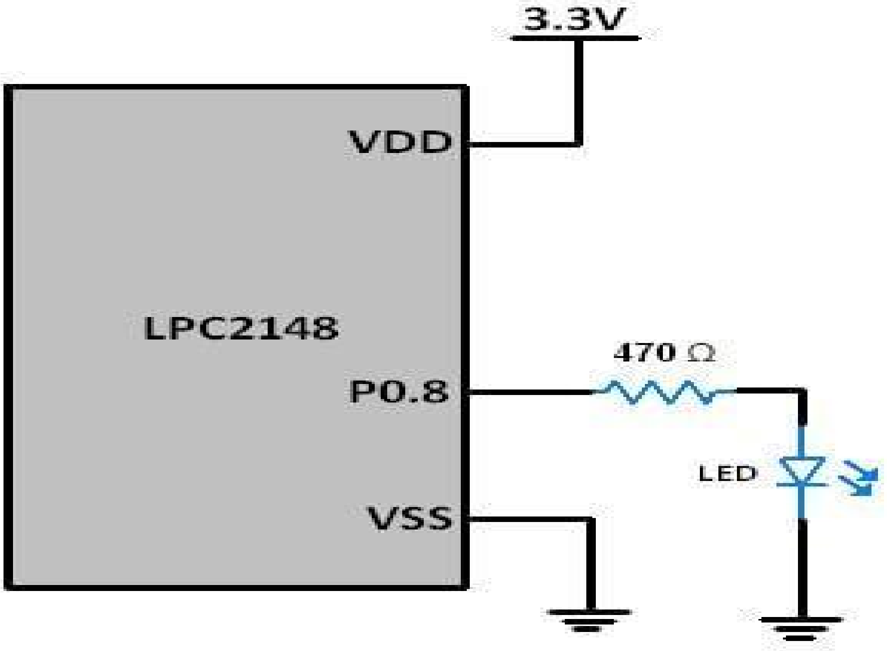
15 12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	MR3S	MR3R	MR3I	MR2S	MR2R	MR2I	MR1S	MR1R	MR1I	MR0S	MR0R	MR0I

T0MCR (Timer0 Match Control Register)

- **Bit 0 – MR0I (Match register 0 interrupt)**
  0 = This interrupt is disabled
  1 = Interrupt on MR0. An interrupt is generated when MR0 matches the value in TC (Timer Counter)
- **Bit 1 – MR0R (Match register 0 reset)**
  0 = This feature is disabled
  1 = Reset on MR0. The TC (Timer Counter) will be reset if MR0 matches it
- **Bit 2 – MR0S (Match register 0 stop)**
  0 = This feature is disabled
  1 = Stop on MR0. The TC (Timer Counter) and PC (Prescale Counter) is stopped and Counter Enable bit in T0TCR is set to 0 if MR0 matches TC
- MR1, MR2 and MR3 bits function in the same manner as MR0 bits.

Example

Let’s write a code for generating a delay of 100msec and blink LEDs using LPC2148 Timer.



PCLK = 30MHz

Hence, we will load T0PR with a value 29, so that the TC will increment after every 30 PCLK rising edges.

30MHz/30 = 1MHz, which gives 1μsec time.

To get 100msec time, we will have to load T0MR with 100000 (1 μsec * 1000 = 1msec, 1msec * 100 = 100msec. Hence, 1000 * 100 = 100000)

This will give us a delay of 100msec.

Here we are using timer interrupt MR0. On each compare match between MR0 and TC, the ISR for Timer0 will be executed where we are toggling the pin connected to the LED.

Program for Timer Delay using LPC2148 Microcontroller

```
/*
  Delay using Timer in LPC2148(ARM7)
  http://www.electronicwings.com/arm7/lpc2148-timercounter
*/

#include <lpc214x.h>

__irq void T0_ISR (void)
{
        IO0PIN = ( IO0PIN ^ (0x00000100) );     /* Toggle P0.8 pin */
        T0IR = ( T0IR | (0x01) );
        VICVectAddr = 0x00;
}


int main (void)
{
        VPBDIV = 0x00000002;  /* For Pclk = 30MHz */
        /* We have configured Cclk=60MHz. Above instruction makes Pclk = Cclk/2 = 30MHz */
        PINSEL0 = PINSEL0 | 0x00000020;  /* Configure P0.2 as Capture 0.0 */
        IO0DIR = ( IO0DIR | (0x00000100) );  /* 8 P0.8-P0.15 as output pins for LED port */
        IO0PIN = IO0PIN | 0x00000100;  /* Writing 1 to LED pin P0.8 */
        VICVectAddr0 = (unsigned) T0_ISR;  /* T0 ISR Address */
        VICVectCntl0 = 0x00000024;  /* Enable T0 IRQ slot */
```