## Counter Mode in PIC

Counter mode is selected by setting the TOCS bit in the OPTION register. In this mode, the Timer0 module counts the external clock pulses applied to its RA2/T0CKI pin. The counter will increment either on rising or falling edge of the clock pulses, which is software selectable by the T0SE (Timer0 Source Edge) bit of the OPTION register. If you set the T0SE bit, the timer will increment on the falling edge of every clock pulse arriving at the RA2/T0CKI pin. Again, the range of the counter can be extended by the use of the prescaler.

The maximum clock frequency at the T0CKI input is limited by the synchronization requirement with the internal clock. Each machine cycle (or instruction cycle) for a PIC microcontroller consists of four clock cycles, which are named as Q1, Q2, Q3, and Q4. The synchronization of T0CKI with the internal clock is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of each machine cycle. Therefore, the external clock signal at T0CKI should remain high or low for at least half of the duration of the machine cycle (which is 2Tosc, Tosc is the period for the main oscillator), plus an additional resistor-capacitor delay of 20 ns. This determines the minimum value of the pulse width that enters through the T0CKI pin. The minimum time period of the input clock pulse is, therefore, 4Tosc + 40 ns, and the maximum frequency will be the reciprocal of this.

For example, if the main oscillator frequency is 4 MHz (Tosc 0.25 ?s), the machine cycle will be 4 x Tosc = 1 ?s long. An external clock signal going directly into the counter (without the prescaler) should be high for at least 2Tosc+20 ns = 520 ns and low for at least the same time, giving the total time period of 520 x 2 = 1040 ns. Therefore, the limit for maximum input frequeny would be 1/1040 ns = 961.5 KHz. If the prescaler is used, the electrical specification of PIC16F688 says that the external clock input must be high and low for at least 10 ns, which gives the maximum countable frequency through T0CKI pin equal to 50 MHz. Read *Microchip's tutorials on Timers:* ***Part 1*** *and* ***Part 2*** for further details.

## Experimental Setup and Software

After reviewing the theory, lets think about doing some experiments with the Timer0 module. First, we will create an approximate 1 sec delay using the Timer0 module as a timer. The PIC16F688 runs at 4 MHz clock frequency, so the duration of a machine cycle is 1 ?s. To generate 1 sec delay interval, the timer should count 1000000 machine cycles. If we use the prescaler value of 256, the required count will be reduced to 3906. Now, if we preload the TMR0 with 39, it will overflow after 256-39 = 217 counts. This gives the required number of overflows to make 3906 counts = 3906/217 = 18. With this setting, after every 18 overflows of TMR0 register (preloaded with 39), an approximate 1 sec interval is elapsed. The software below implements this to flash an LED with an approximate 1 sec interval.