Configuration of 8-bit Timer/Counter0

- Timer/Counter0 is a general purpose, single channel, 8-bit Timer/Counter module. The main features are:
  - **Single Channel Counter**
  - **Frequency Generator**
  - **External Event Counter**
  - **10-bit Clock Prescaler**

Configuration of 8-bit/16-bit Timer/Counter0

- AVR has powerful and multifunctional timers.
- Timer is simply a register of 8 or 16 bit size. This is known as the resolution of the timer (ie 8 bit timer, 16 bit timer).
- In an 8 bit timer, the register is 8 bits long and thus can store a number from 0 to 255. Likewise, a 16 bit timer can hold a value between 0 to 65535.
- This register has a property of increasing or decreasing its value without any intervention by CPU at a rate frequently defined by the user. This frequency (at which the timer register increases/decreases) is known as the Timer Frequency.
- Timer Frequency can be less than or equal to the CPU clock frequency

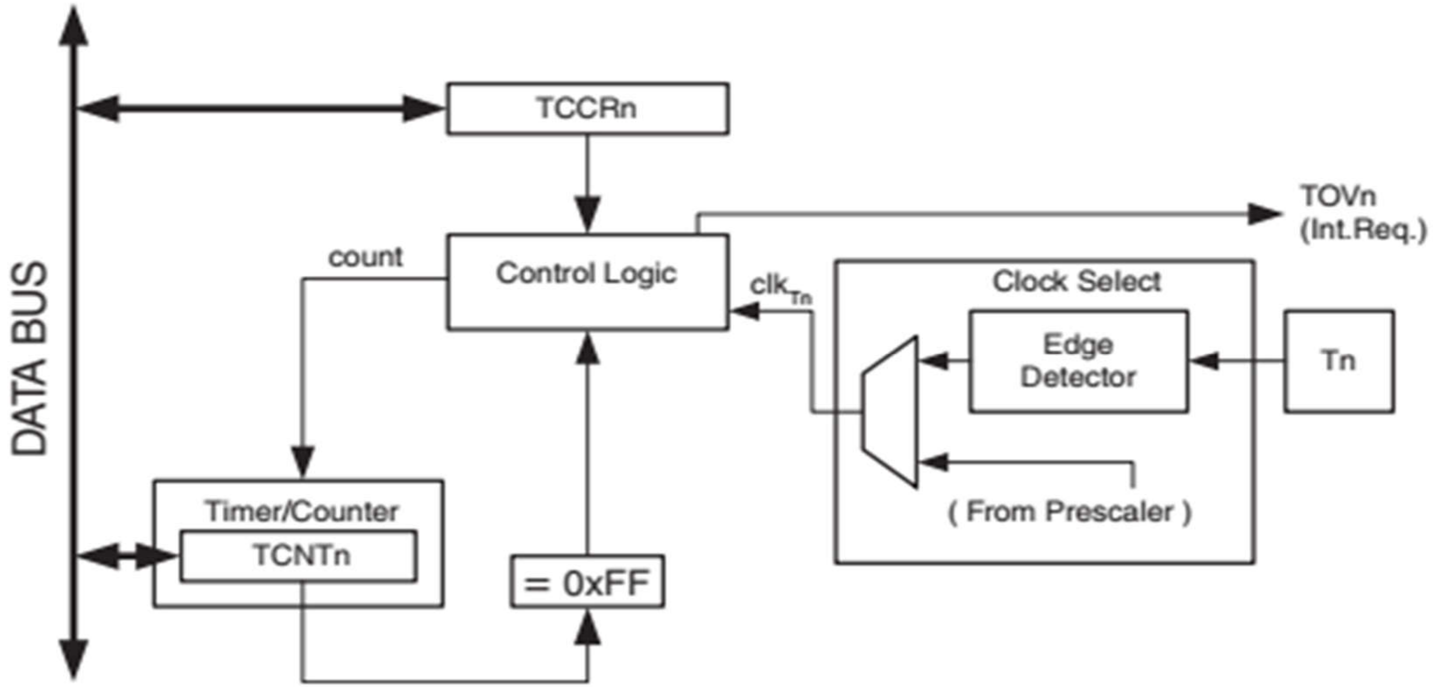
Configuration of 8-bit/16-bit Timer/Counter0

- **8-bit Timer** :
- Simplest timer is (**TIMER0) T0.**
- The Atmega controllers provide hardware counters.
- Those counters are registers that are incremented normally by a signal from the oscillator which also drives the Atmega.
- The oscillator is not connected directly but via a variable prescaler to run slower.
- To every counter there is at least one compare register . When the counter value equals the value of the compare register, a certain action can be triggered.
- Actions can also be triggered when the timer value reaches an overflow.

Configuration of 8-bit/16-bit Timer/Counter0

- **Prescaler:**
- Used to set the Timer frequency
- It is the method of generating the clock frequency for the TIMER from the CPU clock by dividing it with a suitable number.
- **Overflow:**
- The timer in some conditions automatically take an action or informs the CPU to suspend the current execution, by an Interrupt signal.
- Example of this is the Timer Overflow, i.e an 8 bit timer has counted upto its maximum value (255) and revert to 0.Here, the timer sends a signal or interrupt to the CPU to break its current execution and execute the ISR (interrupt service routine).
- ISR is a function that the CPU should execute whenever an interrupt occurs. The programmer must write into the ISR to handle the interrupt.

Configuration of 8-bit Timer/Counter0



8-bit Timer/Counter Block Diagram

Configuration of 8-bit Timer/Counter0

- Timer/Counter can be clocked by an internal or an external clock source. The clock source is selected by the clock select logic which is controlled by the clock select bits located in the Timer/Counter Control Register.
- The Timer0 in ATmega8 has some registers, and few of them are introduced here :
- **TCCR0:** This Timer Counter Control Register is used to configure the timer.

Configuration of 8-bit Timer/Counter0

- **TCNT0:** Timer Counter0 register is the “real” counter in the TIMER0 counter
- **TIMSK:** Timer Interrupt Mask Register, used to activate/deactivate interrupts related to timer.
- **TOIE0:** This bit when sets to “1” enables the OVERFLOW interrupt.

The output from the clock select logic is referred to as the timer clock(**clk**$_{T0}$).

Timer/Counter Clock Sources

- The Timer/Counter can be clocked by an internal or an external clock source.

- The clock source is selected by the clock select logic which is controlled by the clock select (CS02:0) bits located in the Timer/Counter Control Register (TCCR0)

Timer/Counter Clock Sources

- **Counter Unit:**

  The main part of the 8-bit Timer/Counter is the programmable counter unit.

- **Operation:**

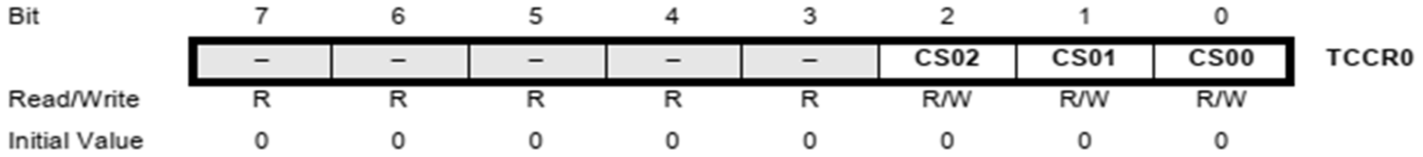
  The counting direction is always up (incrementing), and no counter clear is performed. The counter simply overruns when it passes its maximum 8-bit value (MAX = 0xFF) and then restarts from the bottom (0x00).

  In normal operation the Timer/Counter Overflow Flag (TOV0) will be set in the same timer clock cycle as the TCNT0 becomes zero. The TOV0 Flag in this case behaves like a ninth bit, except that it is only set, not cleared.

  However, combined with the timer overflow interrupt that automatically clears the TOV0 Flag, the timer resolution can be increased by software. A new counter value can be written anytime by CPU.

8-bit Timer/Counter Register Description

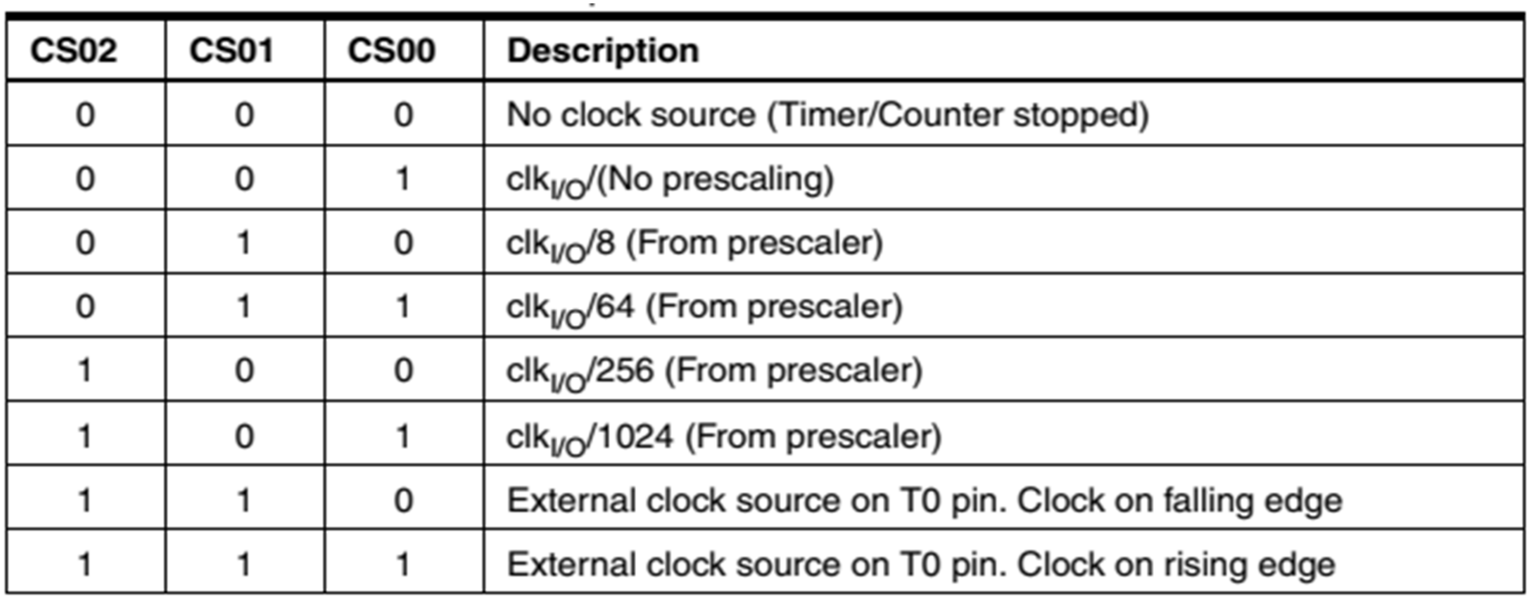
- **Timer/Counter Control Register –TCCR0:**



Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	–	CS02	CS01	CS00	TCCR0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

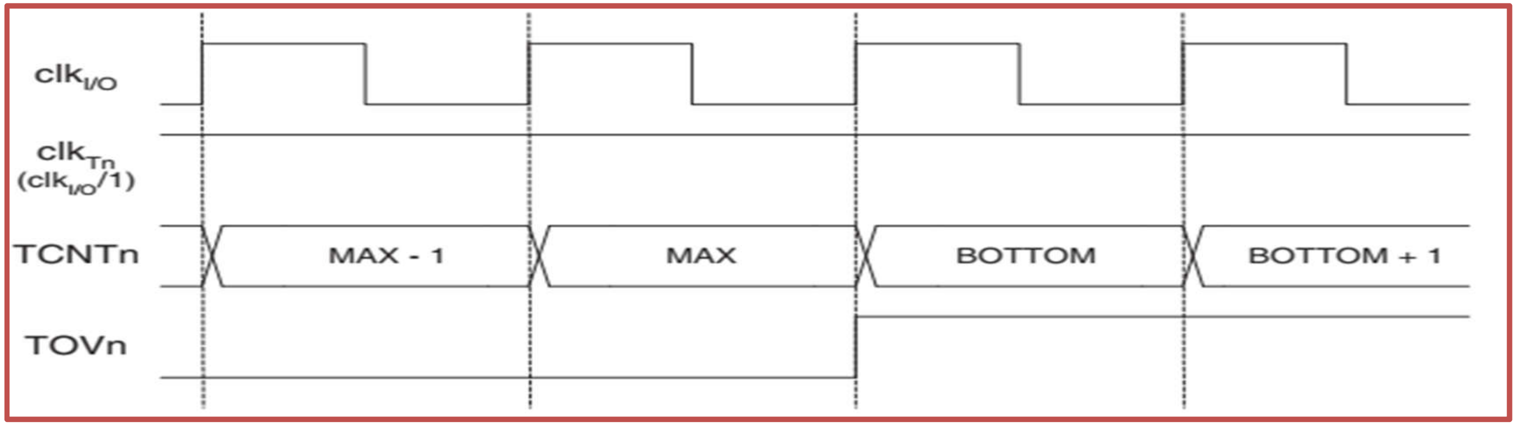
• Bit 2:0 – CS02:0: Clock Select

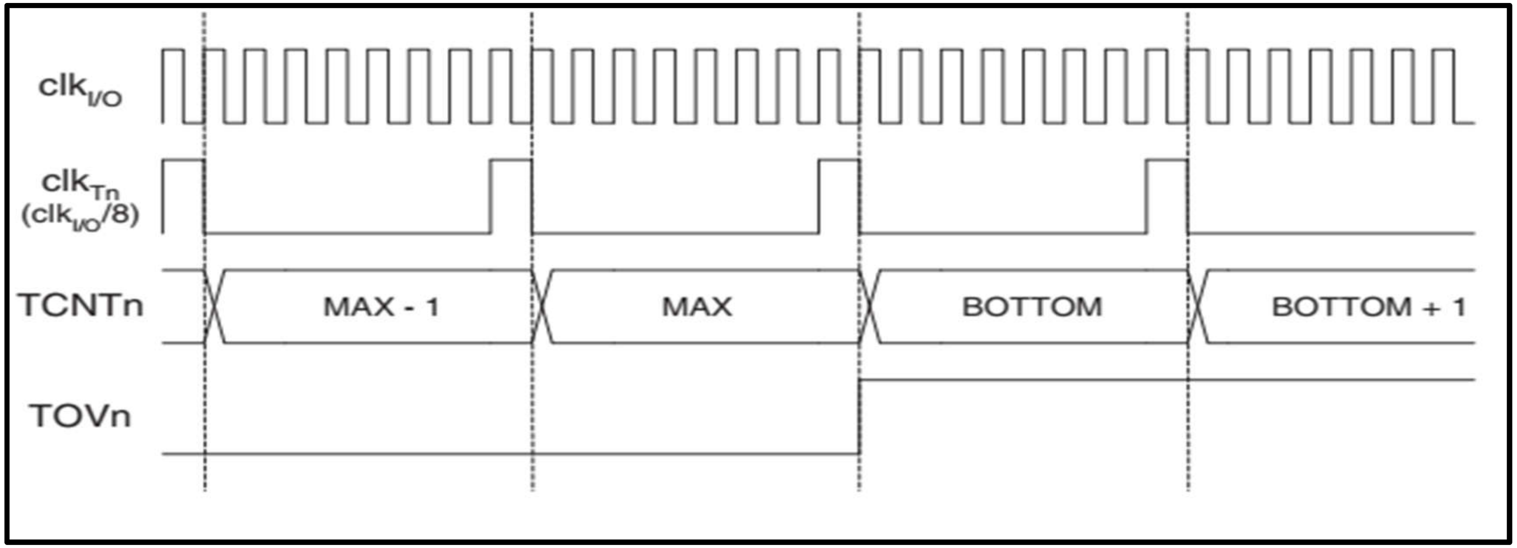
The three clock select bits select the clock source to be used by theTimer/Counter.



CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	$clk_{I/O}$/(No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge
1	1	1	External clock source on T0 pin. Clock on rising edge

Timer/Counter Timing Diagram with/ without Prescaler

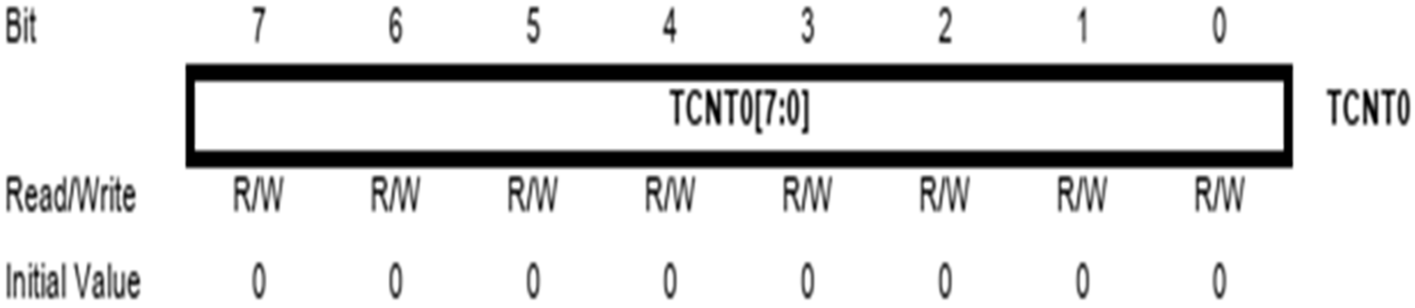




8-bit Timer/Counter Register Description

- **Timer/Counter Register – TCNT0 :**

The Timer/Counter Register gives direct access, both for read and write operations, to the Timer/Counter unit 8-bit counter.



Bit	7	6	5	4	3	2	1	0	
	TCNT0[7:0]								TCNT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

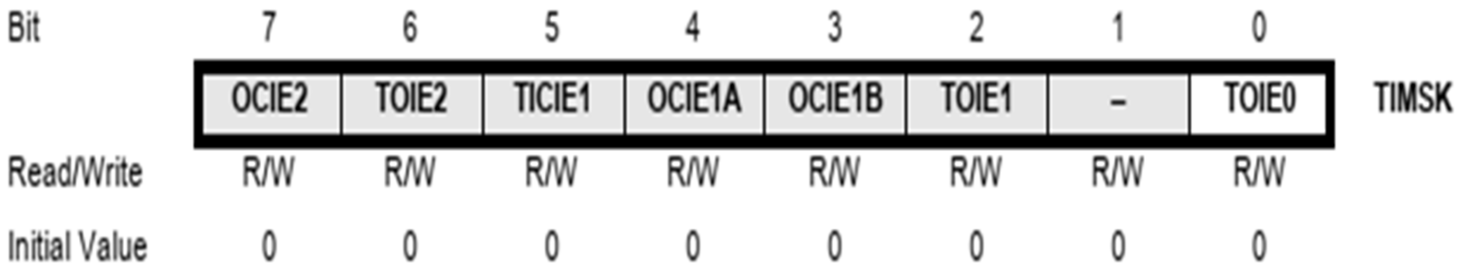
.

8-bit Timer/Counter Register Description

- **Timer/Counter Interrupt Mask Register – TIMSK:**

Bit 0 – TOIE0: Timer/Counter0 Overflow Interrupt Enable

When the TOIE0 bit is written to one, and the I-bit in the Status Register is set (one), the Timer/Counter0 Overflow interrupt is enabled. The corresponding interrupt is executed if an overflow in Timer/Counter0 occurs, that is, when the TOV0 bit is set in the Timer/Counter Interrupt Flag Register – TIFR.

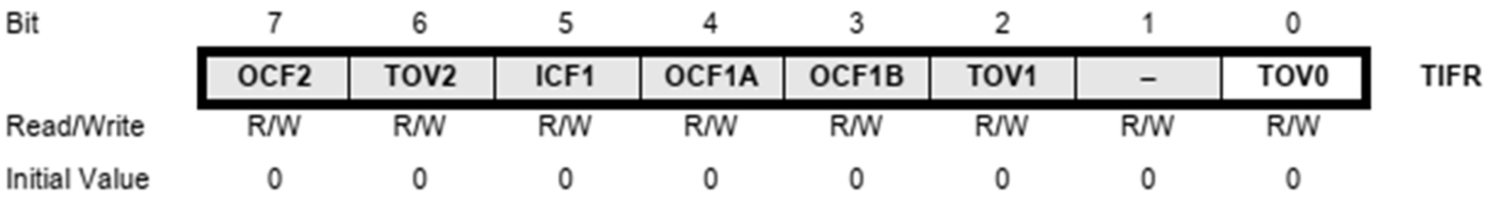


Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	–	TOIE0	TIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

8-bit Timer/Counter Register Description

- **Timer/Counter Interrupt Flag Register – TIFR:**
- **Bit 0 – TOV0: Timer/Counter0 Overflow Flag**

The bit TOV0 is set (one) when an overflow occurs in Timer/Counter0. TOV0 is cleared by hardware when executing the corresponding interrupt Handling Vector. Alternatively, TOV0 is cleared by writing a logic one to the flag. When the SREG I-bit, TOIE0 (Timer/Counter0 Overflow Interrupt Enable), and TOV0 are set (one), the Timer/Counter0 Overflow interrupt is executed.

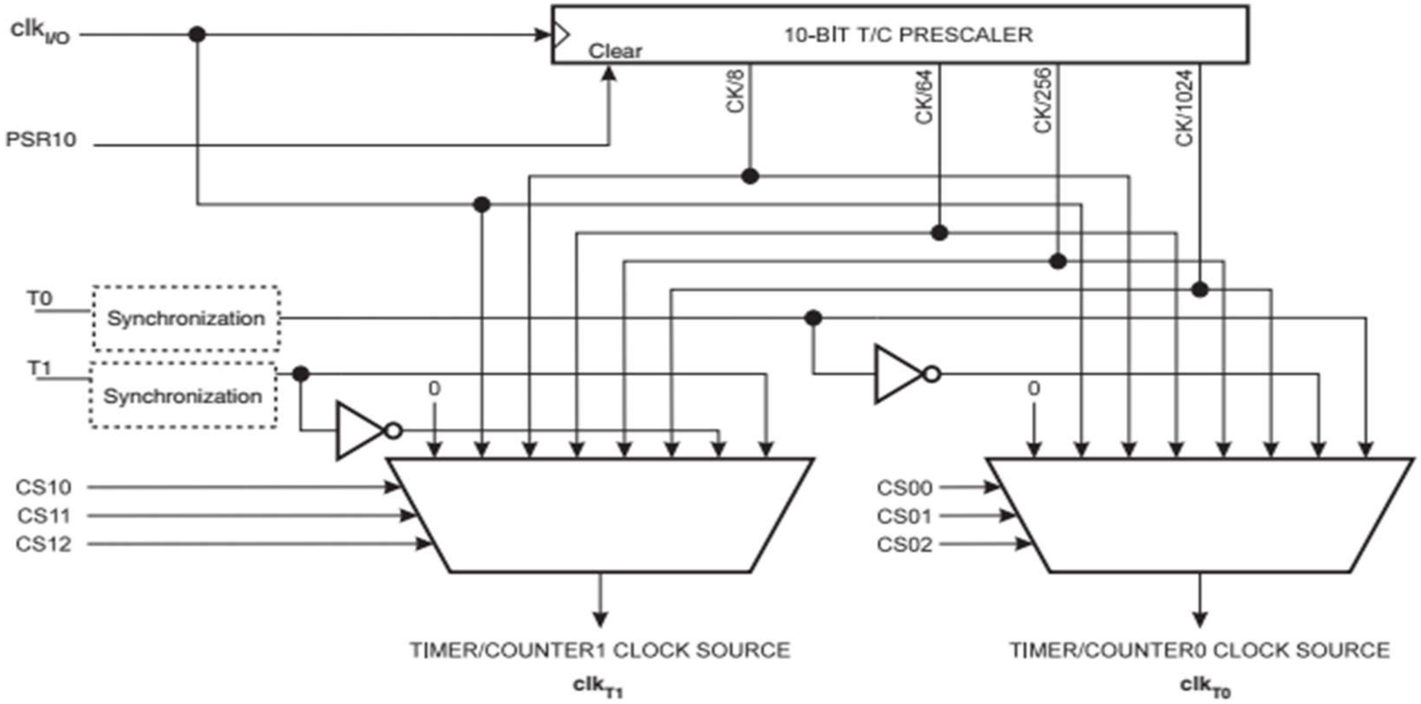


Bit	7	6	5	4	3	2	1	0	
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	–	TOV0	TIFR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Timer/Counter0 and Timer/Counter1 Prescalers

- Timer/Counter1 and Timer/Counter0 share the same prescaler module, but the Timer/Counters can have different prescaler settings.
- An external clock source can not be prescaled.

Figure 31. Prescaler for Timer/Counter0 and Timer/Counter1[(1)]



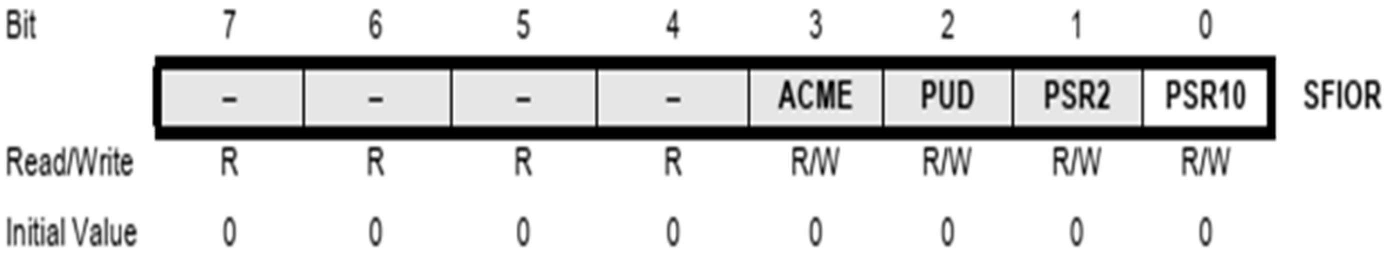
- **Special function IO register (SFIOR):**
- **Bit 0 – PSR10: Prescaler Reset Timer/Counter1 and Timer/Counter0**

✓ When this bit is written to “1”, the Timer/Counter1 and Timer/Counter0 prescaler will be reset.

✓ The bit will be cleared by hardware after the operation is performed.

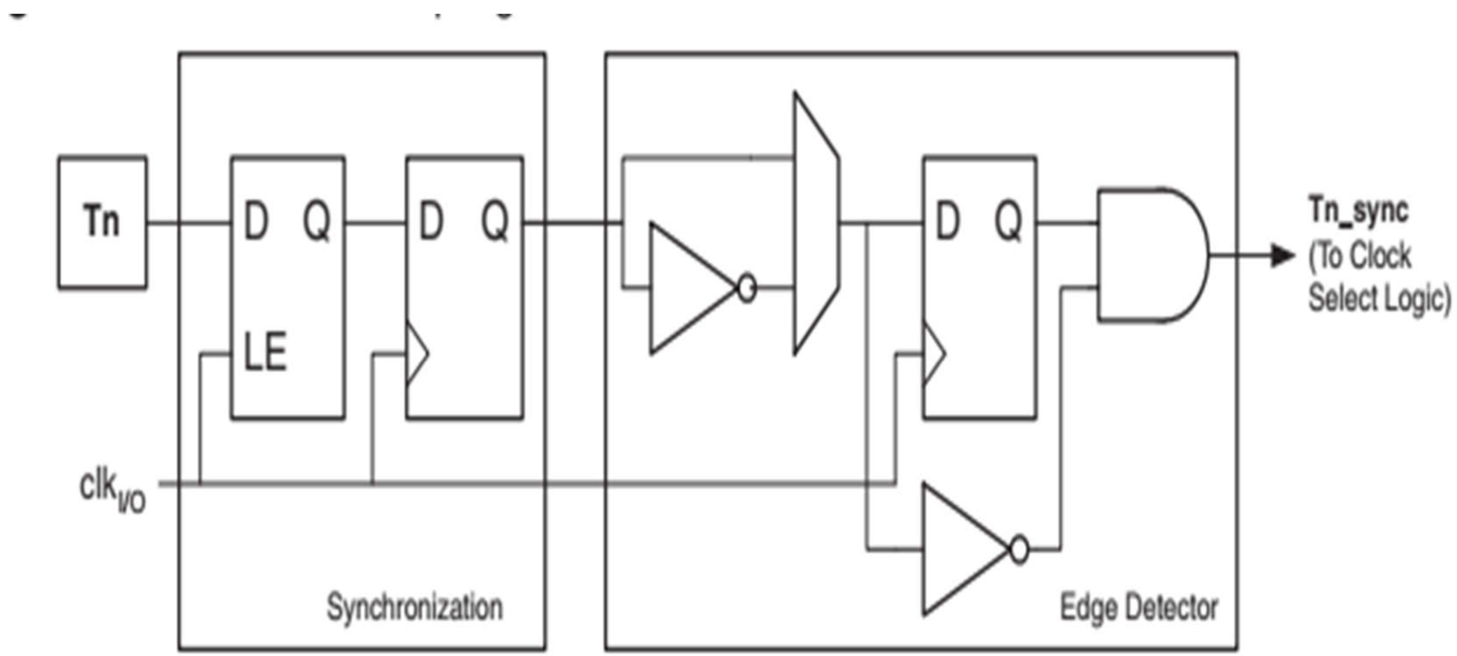
✓ Writing a zero to this bit will have no effect.

✓ This bit will always be read as zero.



Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	ACME	PUD	PSR2	PSR10	SFIOR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Synchronization Logic on input pins T1/T0



T1/T0 Pin Sampling

Synchronization Logic on input pins T1/T0

- An external clock source applied to the T1/T0 pin can be used as Timer/Counter clock (clkT1/clkT0).

- The T1/T0 pin is sampled once every system clock cycle by the pin synchronization logic. The synchronized (sampled) signal is then passed through the edge detector.

- The edge detector generates one clk_{T1}/clk_{T0} pulse for each positive (CSn2:0 = 7) or negative (CSn2:0 = 6) edge it detects.