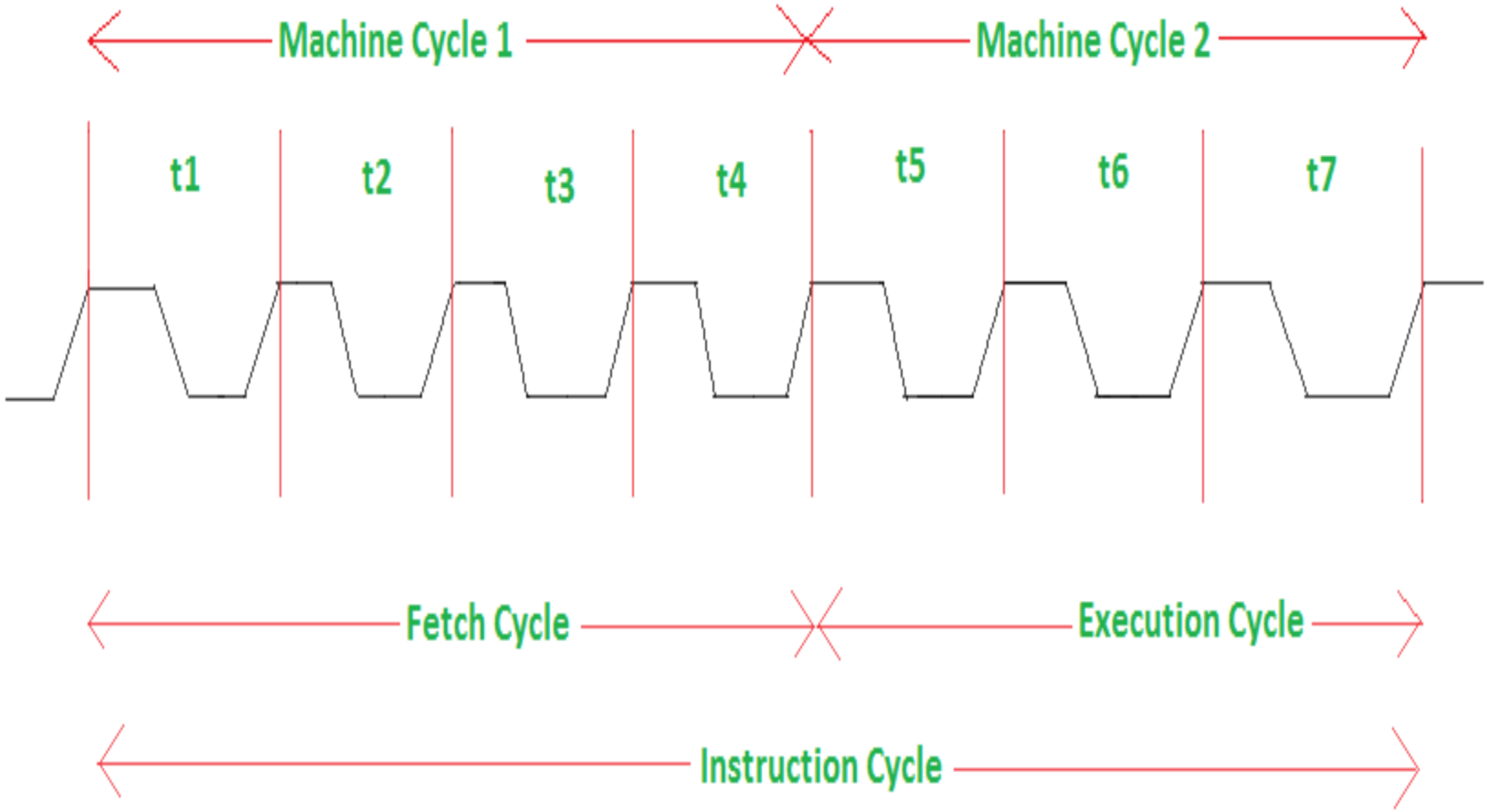
Clock Circuitry

- The processor has to be given the clock input to one of the pins.
- To generate the clock signal, a **crystal oscillator** are required.
- For some processors, the oscillator circuitry is in-built. And for some external circuit is added.
- All processor events are related to the clock. The higher the clock frequency, the higher the speed of the processor.

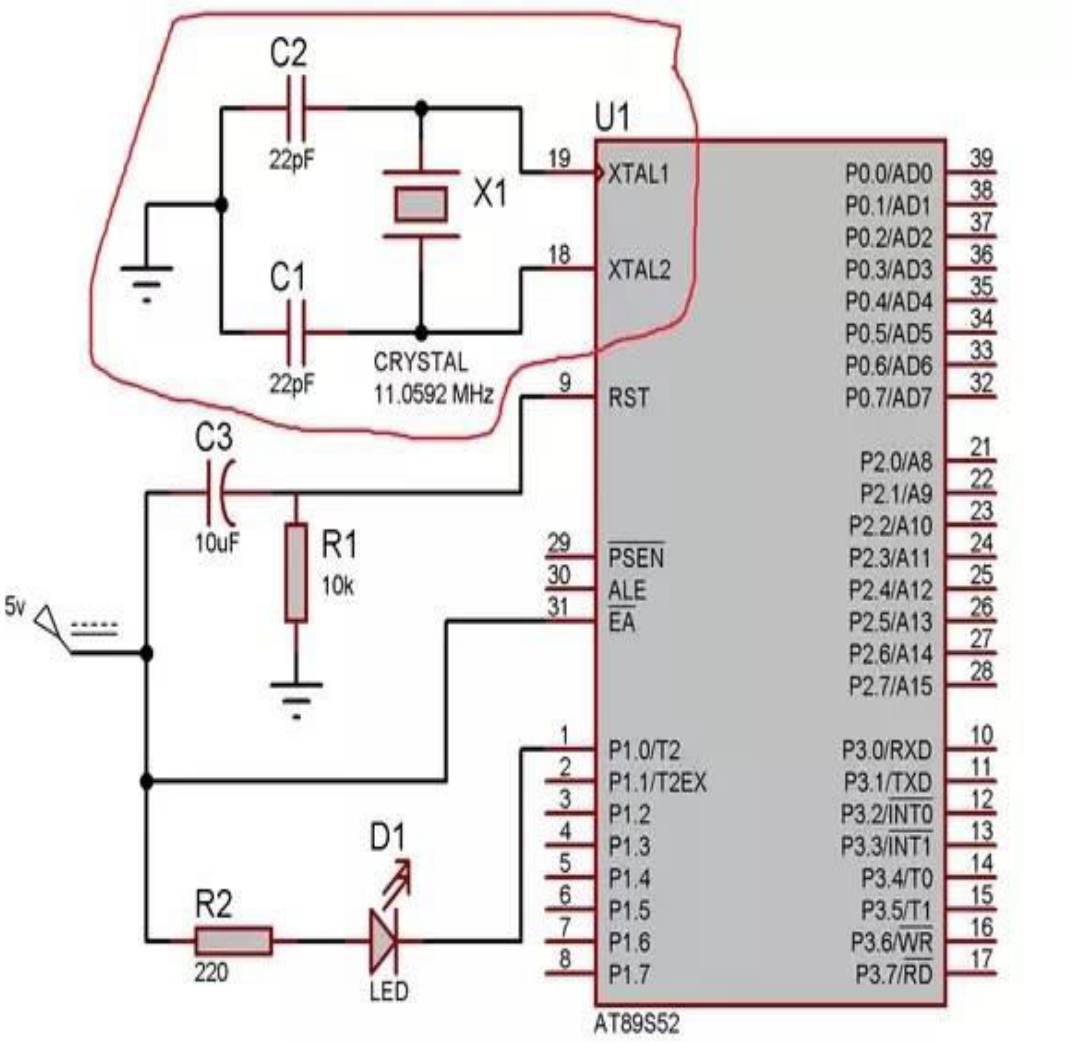
Cycle

- Instruction Cycle
- Machine Cycle
- Clock Period

Cycle


Machine Cycle 1
Machine Cycle 2
t1
t2
t3
t4
t5
t6
t7
Fetch Cycle
Execution Cycle
Instruction Cycle

8051 microcontroller connected with crystal oscillator



Clock Circuitry

- The CPU takes a certain number of clock cycles to execute an instruction. In the 8051 family, these clock cycles are referred to as *machine cycles.*
- In the 8051 family, the length of the machine cycle depends on the frequency of the crystal oscillator connected to the 8051 system.

Clock Circuitry

- The frequency of the crystal connected to the 8051 family can vary from 4 MHz to 30 MHz, depending on the chip rating and manufacturer.
- Very often the 11.0592 MHz crystal oscillator is used to make the 8051 -based system compatible with the serial port of the IBM PC

Clock Circuitry

- In the original 8051, one machine cycle lasts 12 oscillator periods. Therefore, to calculate the **period of machine cycle for the 8051**, we take 1/12 of the crystal frequency, then take its inverse.

Clock Circuitry

Example 3-13

The following shows crystal frequency for three different 8051-based systems. Find the period of the machine cycle in each case.
(a) 11.0592 MHz (b) 16 MHz (c) 20 MHz

Solution:

(a) 11.0592 MHz/12 = 921.6 kHz; machine cycle is 1/921.6 kHz = 1.085 µs (microsec-ond)
(b) 16 MHz/12 = 1.333 MHz; machine cycle (MC) = 1/1.333 MHz = 0.75 µs
(c) 20 MHz/12 = 1.66 MHz; MC = 1/1.66 MHz = 0.60 µs

Clock Circuitry

- Advances in CPU design in recent years have increase the 8051 performance.
- The number of machine cycles and the number of clock periods per machine cycle varies among the different versions of the 8051 microcontrollers.
- While the original 8051 design used 12 clock periods per machine cycle, many of the newer generations of the 8051 use much fewer clocks per machine cycle.

Clock Circuitry

Various 8051 Versions

Chip/Maker	Clocks per Machine Cycle
AT89C51 Atmel	12
P89C54X2 Philips	6
DS5000 Dallas Semi	4
DS89C420/30/40/50 Dallas Semi	1

Clock Circuitry

Example 3-18

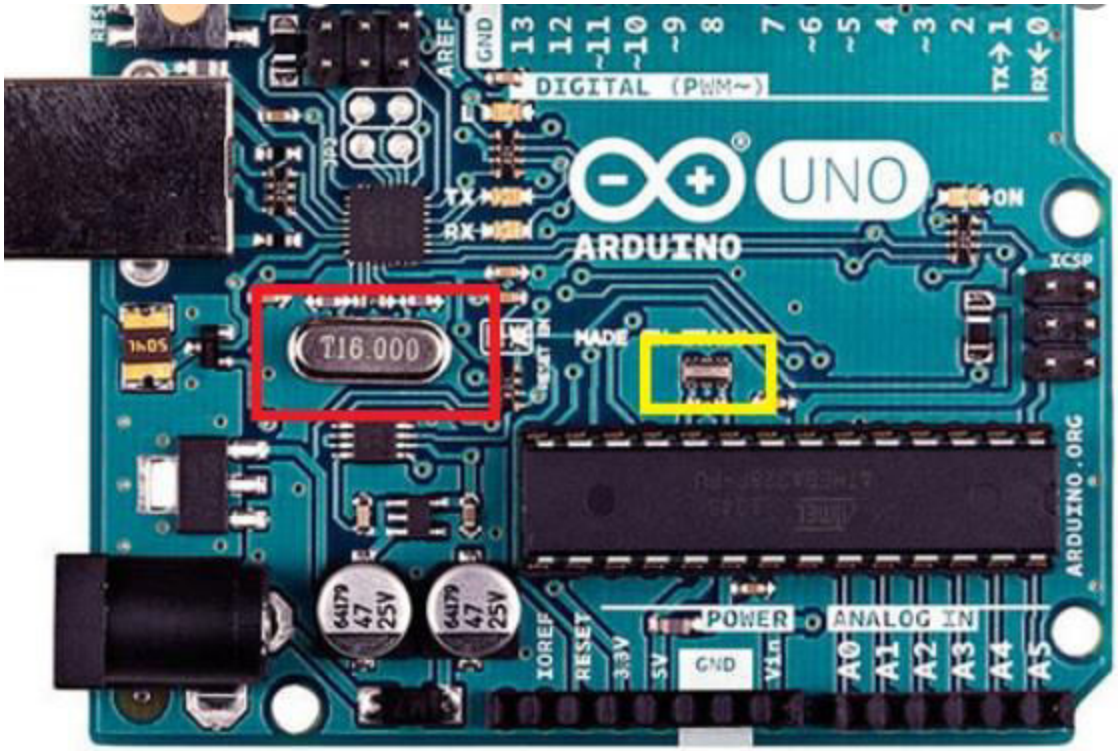
From Table 3-2, find the period of the machine cycle (MC) in each case if XTAL = 11.0592 MHz, and discuss the impact on performance.
(a) AT89C51 (b) P89C54X2 (c) DS5000 (d) DS89C4x0

Solution:

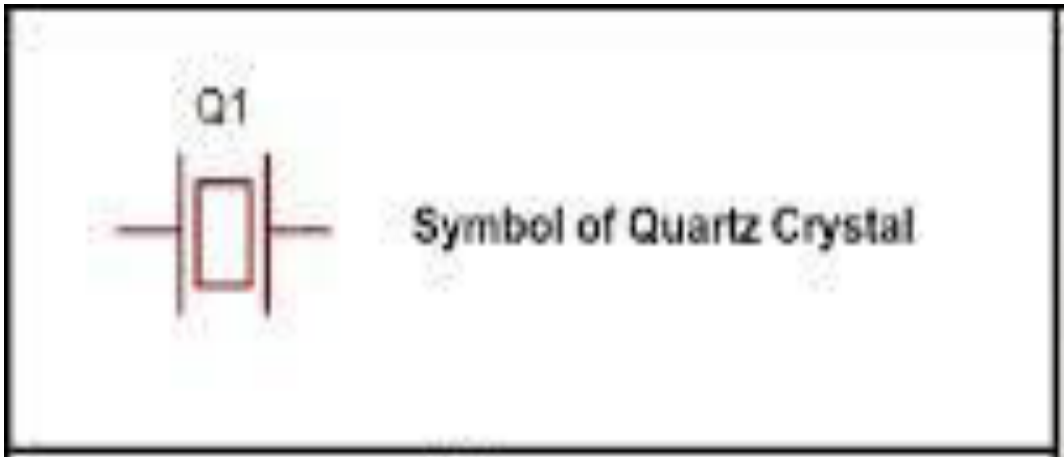
(a) 11.0592 MHz/12 = 921.6 kHz; MC is 1/921.6 kHz = 1.085 µs (microsecond) = 1085 ns
(b) 11.0592 MHz/6 = 1.8432 MHz; MC is 1/1.8432 MHz = 0.5425 µs = 542 ns
(c) 11.0592 MHz/4 = 2.7648 MHz; MC is 1/2.7648 MHz = 0.36 µs = 360 ns
(d) 11.0592 MHz/1 = 11.0592 MHz; MC is 1/11.0592 MHz = 0.0904 µs = 90 ns

This means that if we connect an AT89C51 and a DS89C4x0 to a crystal of the same frequency we get approximately 9 to 10 times performance boost for the DS89C4x0 chip over the AT89C51. See Example 3-20.

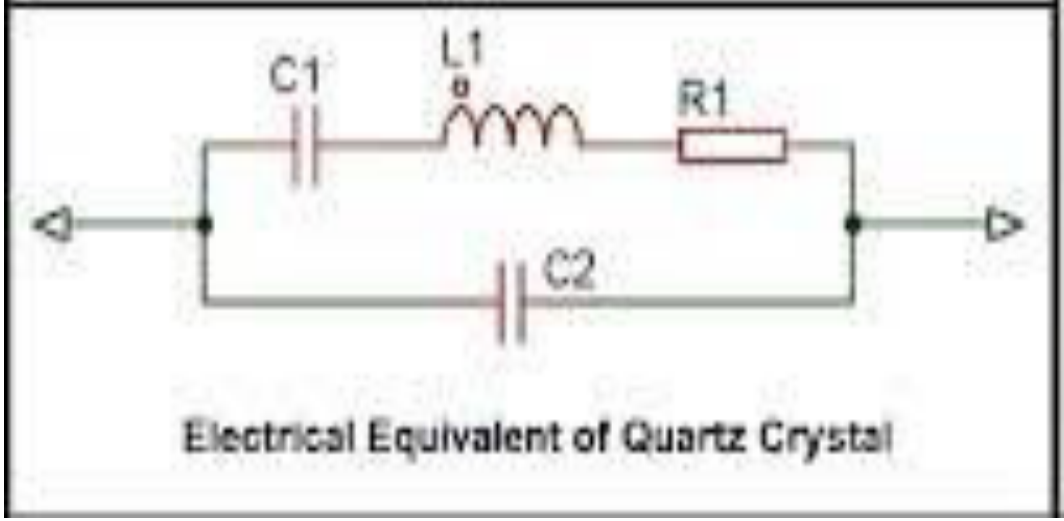
Clock Circuitry



Clock Circuitry



Symbol of Quartz Crystal



Electrical Equivalent of Quartz Crystal

Watchdog Timer/Reset Circuitry

- Self Reliant System.
- Most of the embedded systems do not have a reset button. Due to some software or hardware error, there is need to reset the processor.
- A watchdog timer or simply a watchdog, is a hardware timer for monitoring the firmware execution.
- In this case, a timer is set to a large value and it is decremented slowly. If the timer value reaches zero, the processor is reset through the signal. If everything is fine and healthy, the timer value is again set to the large value.

Watchdog Timer/Reset Circuitry

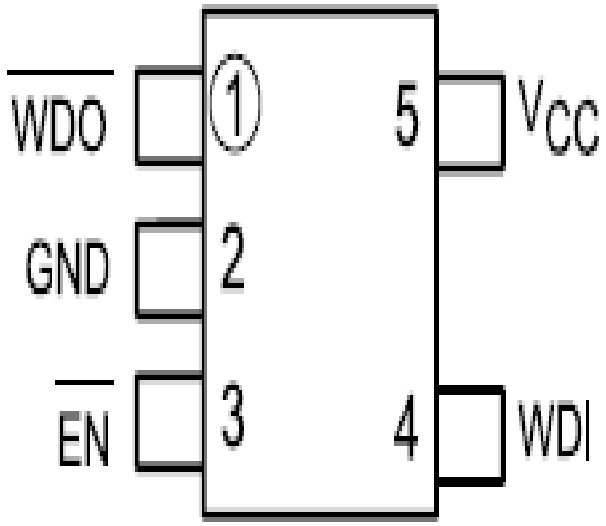
- In some case if reset button is provided and is pressed than a reset signal is sent to the processor.
- Most of the processors implement watchdog as a built-in component and provides status register to control the watchdog timer. If the processor doesn't contain a built in watchdog timer, the same can be implemented using an external watchdog timer IC circuit.

Watchdog Timer/Reset Circuitry

- Watchdog timer is used to regain control when a system has failed due to software error or due to failure of an external device to respond in the expected way.
- A watchdog timer can generate a **NMI** or a **RESET** when a time out value is reached.

Watchdog Timer/Reset Circuitry

The STWD100 watchdog timer circuits are self-contained devices which prevent system failures that are caused by certain types of hardware errors (such as, non-responding peripherals and bus contention) or software errors (such as a bad code jump and a code stuck in loop).

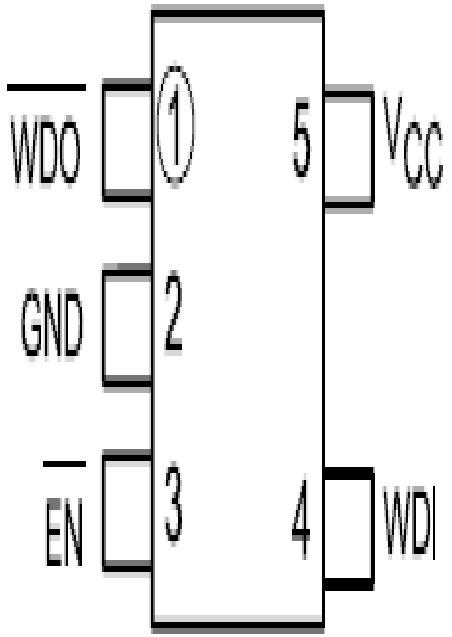


Watchdog Timer/Reset Circuitry

- The STWD100 watchdog timer has an input, WDI, and an output, WDO .
- The input is used to clear the internal watchdog timer periodically within the specified timeout period, t(wd).
- While the system is operating correctly, it periodically toggles the watchdog input, WDI.
- If the system fails, the watchdog timer is not reset, a system alert is generated and the watchdog output, WDO , is asserted.
- The STWD100 circuit also has an enable pin, EN , which can enable or disable the watchdog functionality.

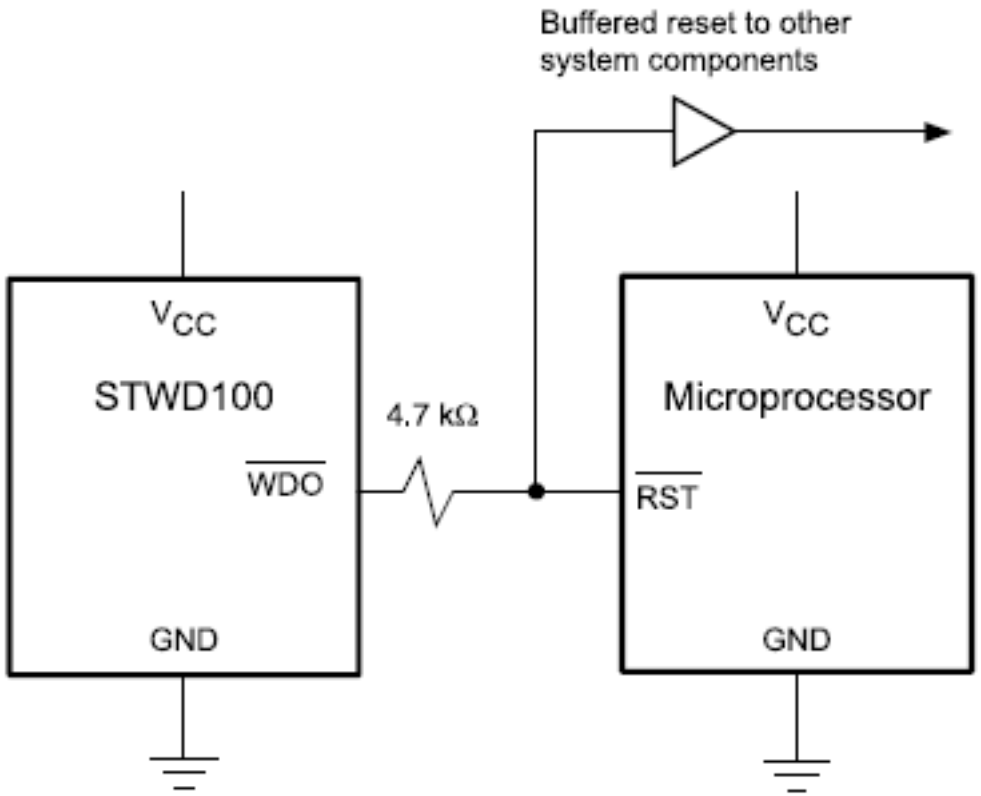
Watchdog Timer/Reset Circuitry

Pin number	Name	Description
1	$\overline{WDO}$	Watchdog output
2	GND	Ground
3	$\overline{EN}$	Enable pin
4	WDI	Watchdog input
5	V_{CC}	Supply voltage



Watchdog Timer/Reset Circuitry

Figure 5: Interfacing to microprocessors with bidirectional reset I/O



Watchdog Timer/Reset Circuitry

- The STWD100 device is used to detect an out-of-control MCU. The user/program has to ensure watchdog reset within the watchdog timeout period, otherwise the watchdog output is asserted and the MCU is restarted.
- The STWD100 can also be enabled or disabled by the chip enable pin.
- The **WDI input** has to be toggled within the watchdog timeout period, **t(wd)**, otherwise the watchdog output, WDO , is asserted.
- If the timer is not cleared within the **t(wd)**, , the WDO goes low.
- States *"Watchdog input (WDI)"* and *"Watchdog output (WDO)"* are valid under the condition that EN is in logical low state.

Watchdog Timer/Reset Circuitry

- Link(Forum)
- https://www.embedded.com/introduction-to-watchdog-timers/#