Microprocessor systems

EMISY

Integrated and external peripherals in microprocessor systems, part 2
Lecture 7

Semester 19L – Summer 2020

© Maciej Urbanski, MSc

email: M.Urbanski@elka.pw.edu.pl

WUT

ISE

Important remark

This material is intended to be used by the students during the Microprocessor Systems course for educational purposes only. The course is conducted in the Faculty of Electronics, Warsaw University of Technology.

The use of this material in any other purpose than education is prohibited.

This material has been prepared based on many sources, considered by the author as valueable, however it is possible that the material contains errors and misstatements.

The author takes no responsibility for the usage of this material and any potential losses this usage can lead to. Furthermore the author will be very grateful for pointing out any errors found and also for any other useful remarks on the course material and potential upgrades.

Timer – what is it?

- Microcontroller is sequential circuit – it has no idea about time
  - Except it's clock signal source – it can compare handle events based on the clock signal
- There is a need to have a circuit that will allow to handle time – measure, generate delays, compare and count external events, etc.
- Timer maintains the timing of an operation in sync with a system clock.
- Usually microcontrollers have several internal timers/counters
  - They differ in size (bit lenght), functions and speed

Image source : https://www.conrad.pl/

Timers – what can they be used for?

- Timers are used to:
  - Measure time intervals between external events – input event capture
    - Can be used for measuring frequency
  - Generate pulses or sequences of pulses with desired periods – output compare
  - Generate delays in a controlled and reliable way
    - Precision
    - Repeatability
    - Robustness
  - Generate PWM signal to control various devices
  - Control external transmission – SPI, I2C, USART (both synchronous and asynchronous)
    - Many microcontrollers have internal protocol hardware drivers
  - Control watchdog circuits

Timers – how were they done in the past?

- Timers, GPIO ports, etc. were manufactured as standalone chips, like vintage Intel 8155
  - 2 programmable 8 bit GPIO port
  - 1 programmable 6 bit GPIO port
  - 1 programmable 14 bit binary counter/timer

- Modern microcontrollers have all of that included, for instance STM32F042G4U6:
  - 37 GPIO pins – that is more than four 8bit ports
  - Nine 16bit advanced timers (multi-channel)
  - One 32bit timer
  - Four 16bit general purpose timers
  - SysTick timer – separate timer for system, used for example for delay generation

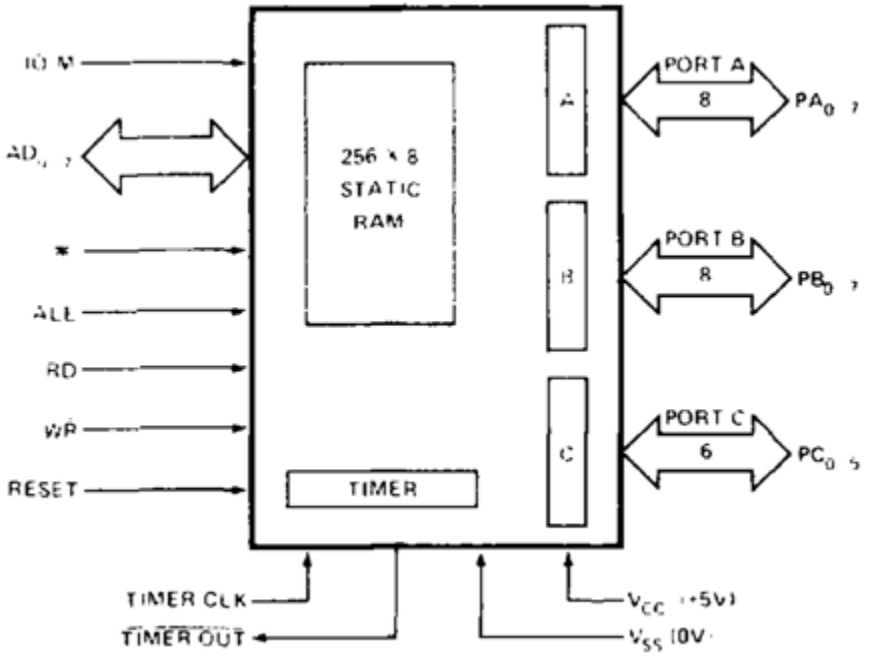


Image source : https://www.jdr.com

Timers in 8051

- 8051 microcontrollers usually have minimum 2 timers
  - AT89C1051 is the exception here
- They are called T0, T1, … , Tn

- These timers are 16-bits and they are using THn and TLn registers to store current timer (counter) value

- Operation mode selection and timer control can be done using TCON and TMOD registers

- Each 8051 timer can work as counter (count external pulses) or timer (counting clock cycles)

- When counting clock cycles the maximum counting frequency is equal to 1/12 of the clock frequency (why?)

- When working as counter the maximum counting frequency is limited to 1/24 of the clock frequency, because external pulse has to be at least one machine cycle long.

Timers in 8051 - modes

- There are four operation modes for 8051 timers

- Mode 0
  - Identical for T0 and T1
  - In this mode timer works as 13 bit counter
  - State of the counter is determined by the value of THn and lower 5 bytes of TLn
  - Counter overflow sets TFn flag in TCON register
  - Timer is enabled when TRn=1 and GATE=0 or ~INTn=1. GATE allows to enable via ~INTn – measure external pulses length
- Mode 1
  - Almost identical with T0, but 16 bit length
- Mode 2
  - Timers working as 8bit counters (using TLn registers) with auto overload handling
  - Overflow causes the rewrite of THn to TLn register and sets TFn flag
    - This allows to generate PWM signal
- Mode 3
  - Different operation for T0 and T1
  - T0 in mode 3 uses TL0 and TH0 as two independent 8 bit counters, working in Mode 0 or 1 (TL0) and TH0 working as machine cycle counter

Timers in 8051 – T2

- Timer T2 is available in 8052 series.
- Configured by T2CON register
- Can work in auto-reload mode
- Can work in capture mode
- Can work as clock for serial port communication
- Mode is defined using RCLK, TCLK, CP/~RL2, TR2 in T2CON register
- In all modes this timer works as 16-bit timer, using TL2 and TH2 registers

Timers in 8051 – modes 0 and 1

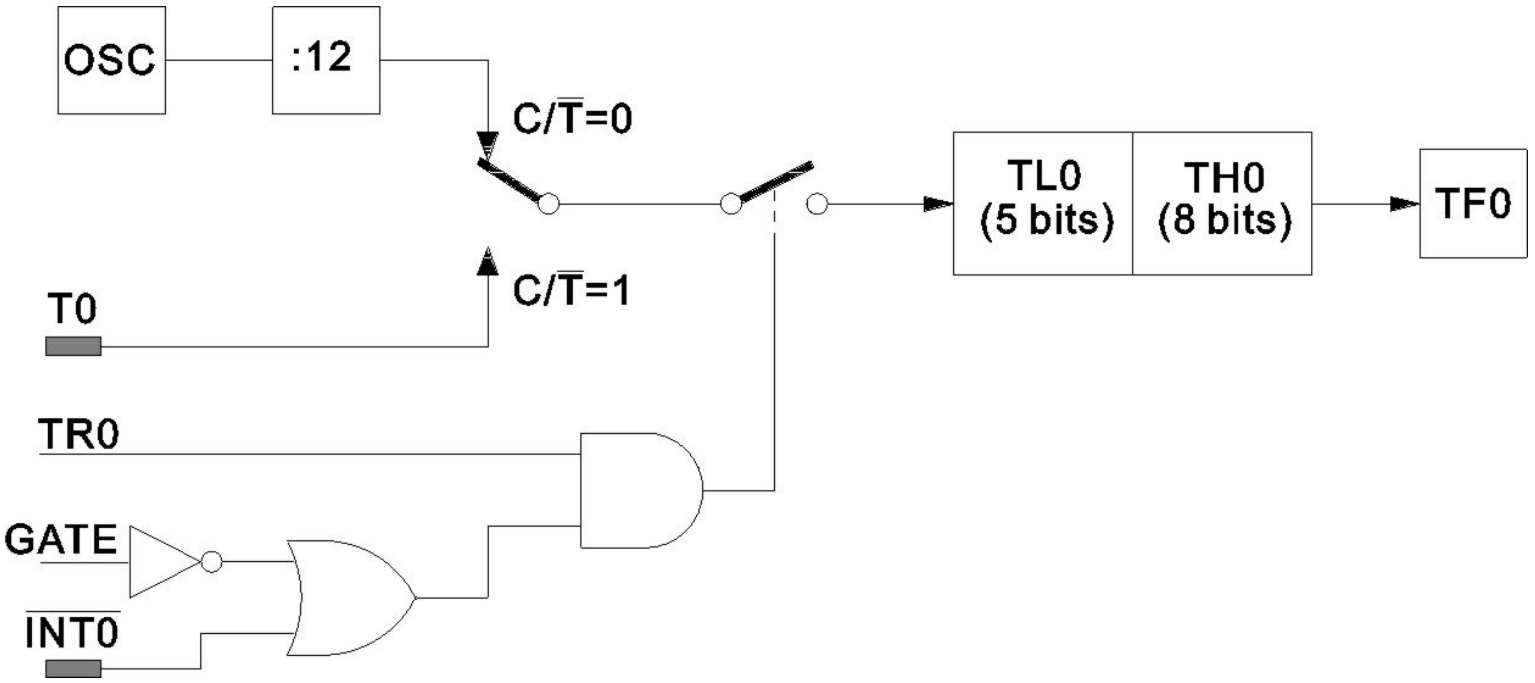


Image source : EMISY course by T. Starecki

Timers in 8051 – mode 2

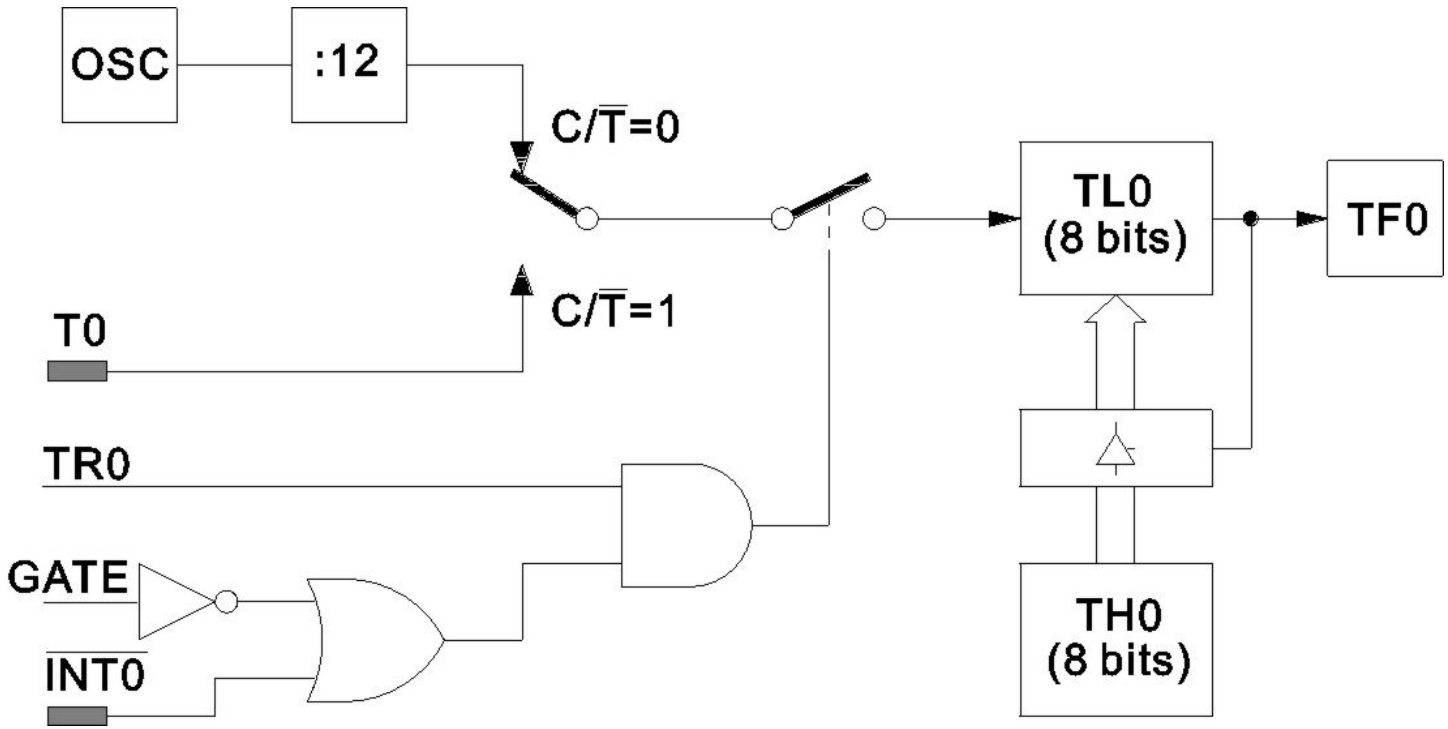


Image source : EMISY course by T. Starecki

Timers in 8051 – mode 3

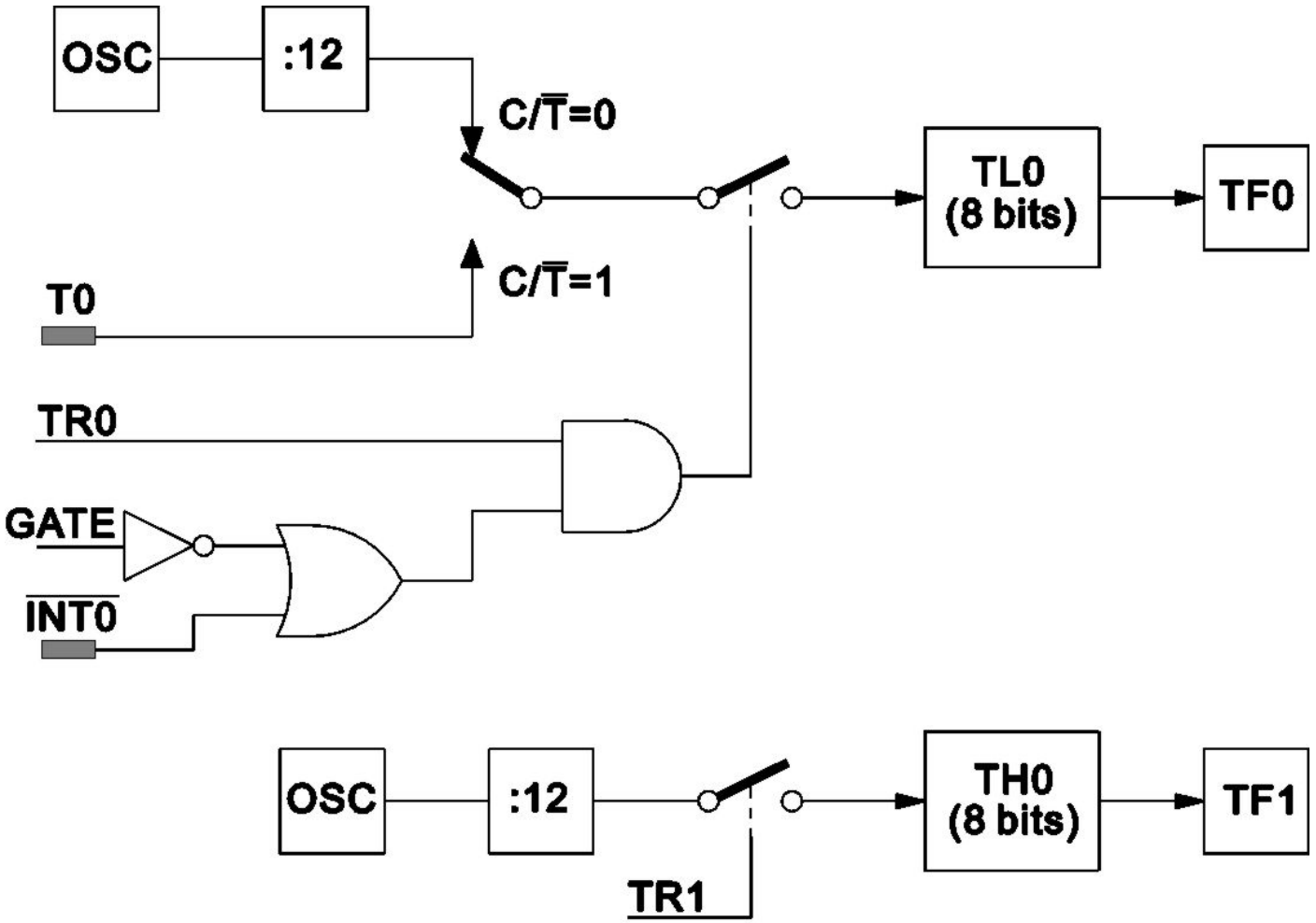
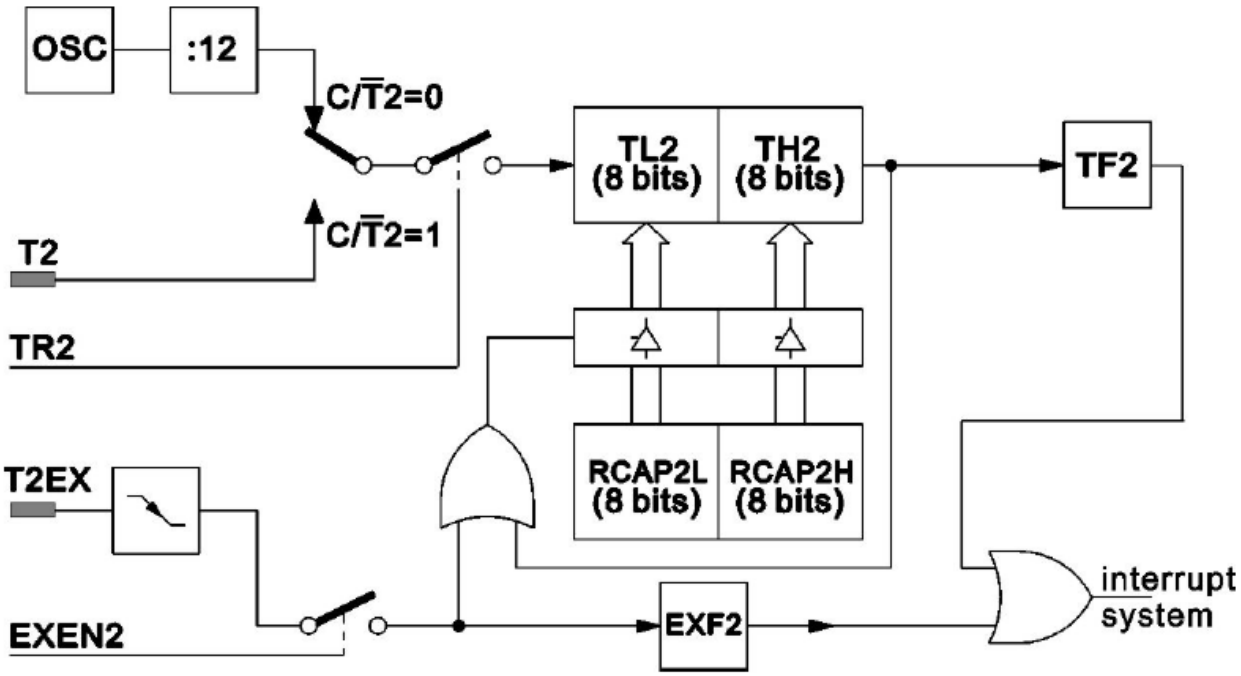


Image source : EMISY course by T. Starecki

8052 Timer T2 modes

- Auto-reload mode



T2CON (0C8H)

TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2

Timers – basic modes

- Capture mode

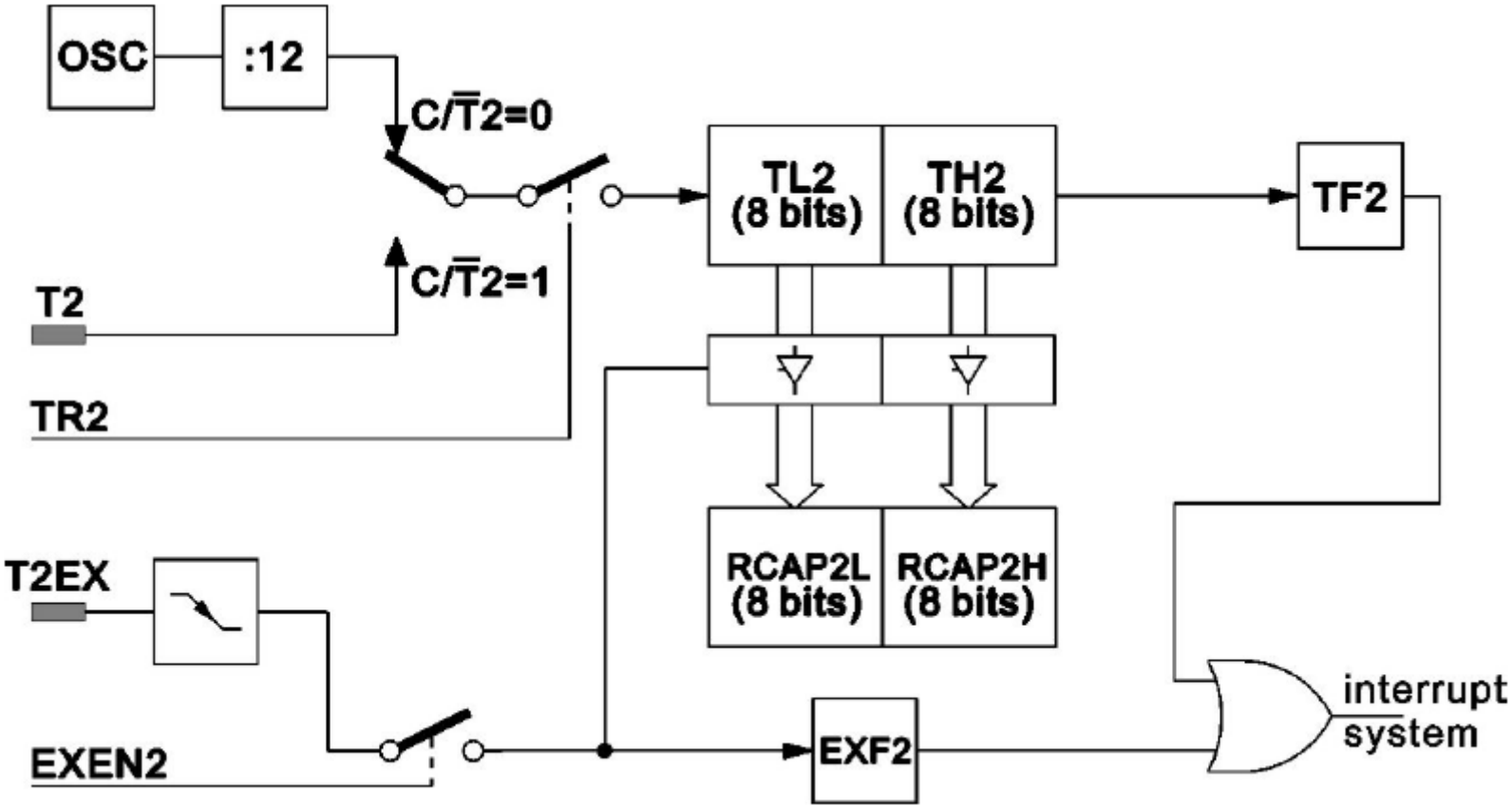
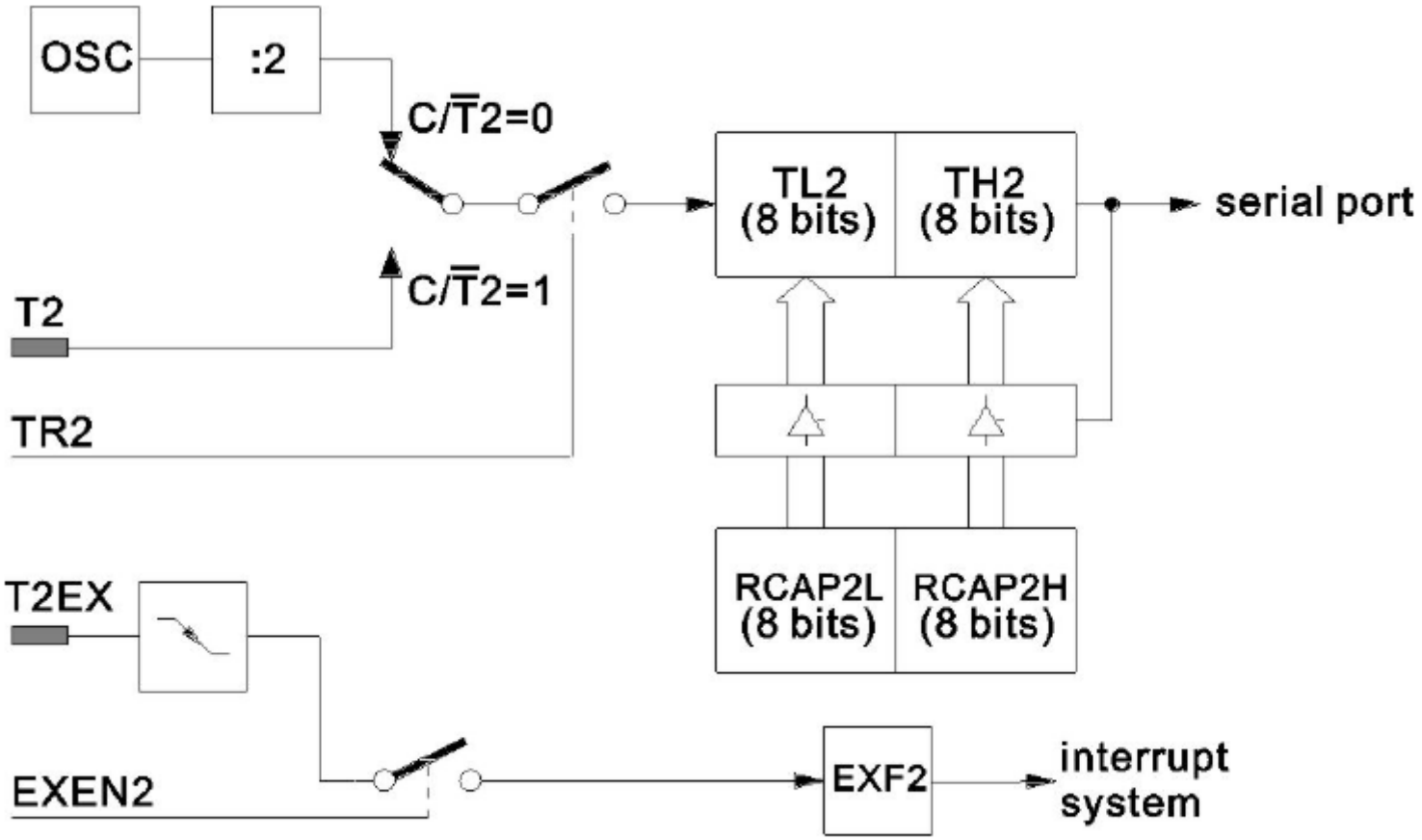


Image source : EMISY course by T. Starecki

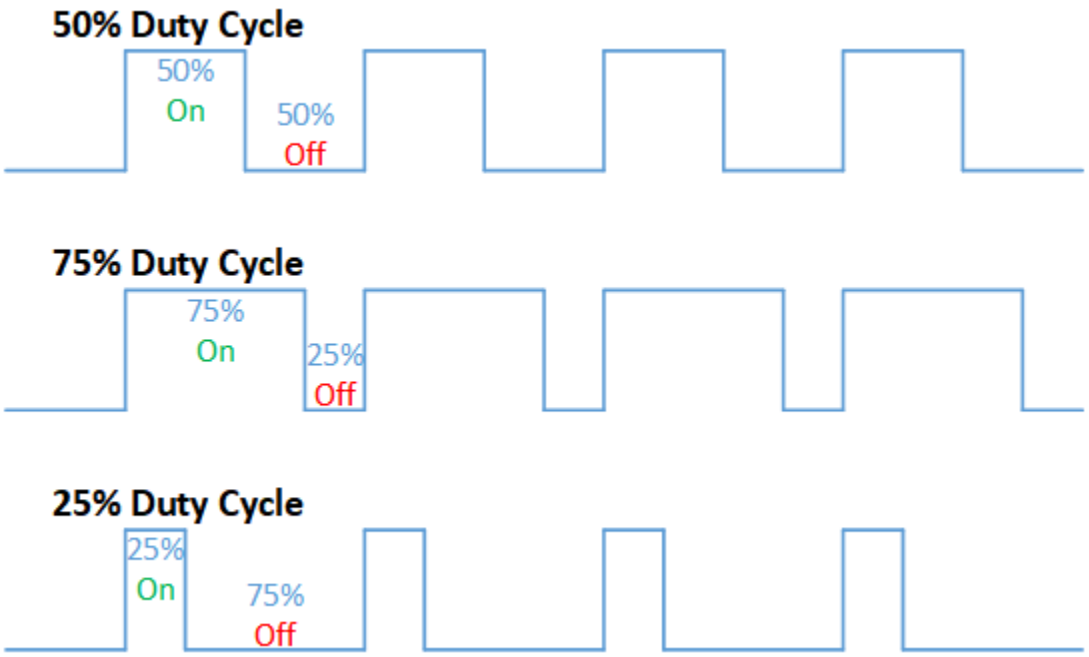
Timers – basic modes

- Baudrate generator mode



What is PWM and how to use it?

- Timers can be used as PWM signal generators
- PWM – Pulse Width Modulation is a method of reducing the average power delivered by an electrical signal, by effecively chopping it up into discrete parts
- PWM uses a rectangular pulse wave. Its pulse width is modulated which results in variation of the average power delivered to the receiver.
- This type of signal can be used to control various devices:
- LEDs (brightness)
- Heaters (temperature, PID controllers)
- Class D audio amplifiers



Source: en.wikipedia.org

Timers in AVRs – example of ATTiny2313

Figure 27. 8-bit Timer/Counter Block Diagram

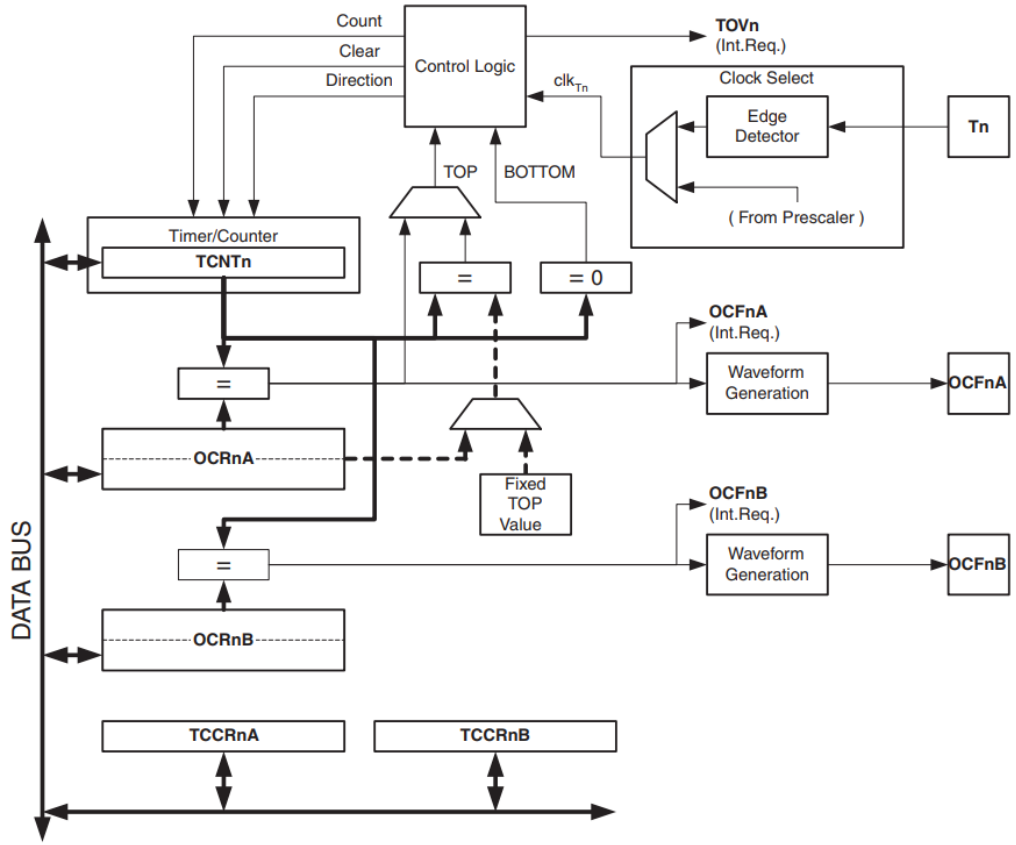


Image source : ATTiny 2313 datasheet

Timers in AVRs

Figure 40. 16-bit Timer/Counter Block Diagram[1]

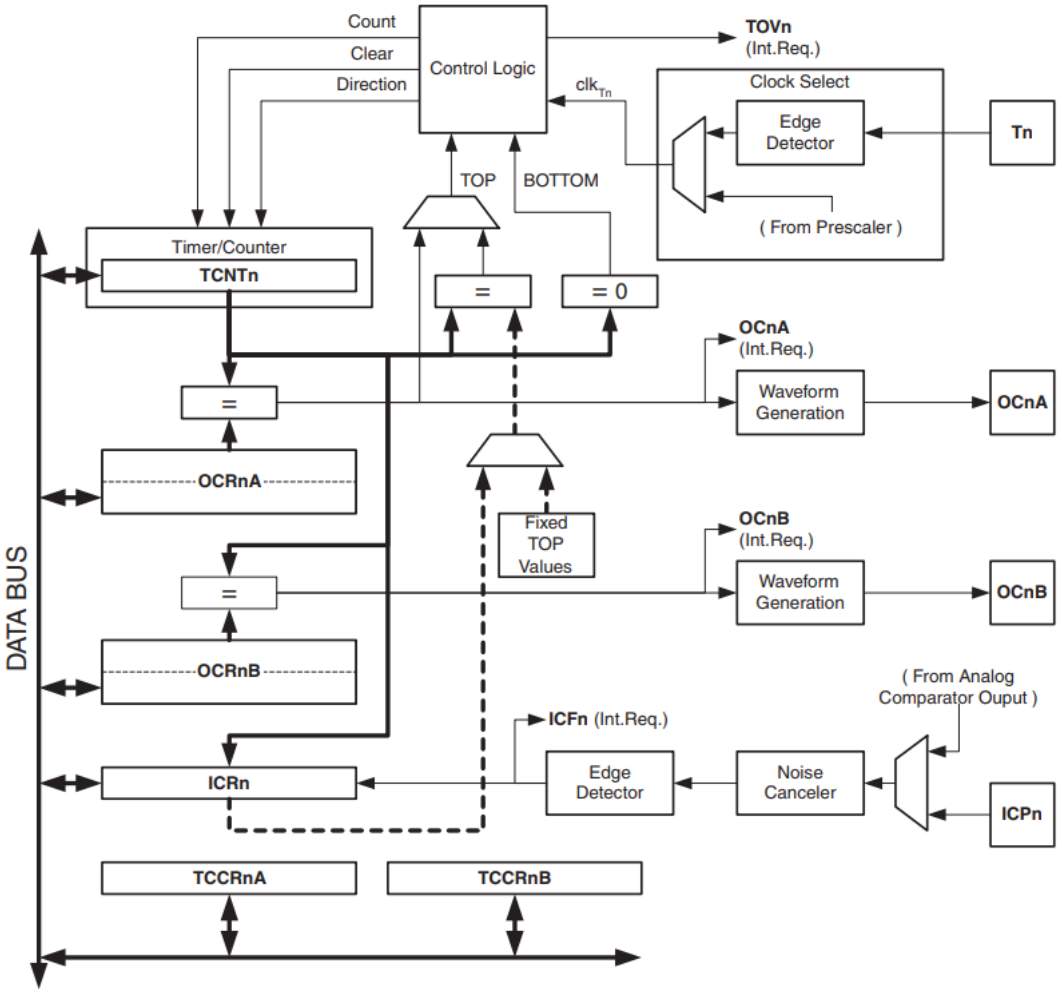


Image source : ATTiny2313 datasheet

Timers in STM32 microcontrollers

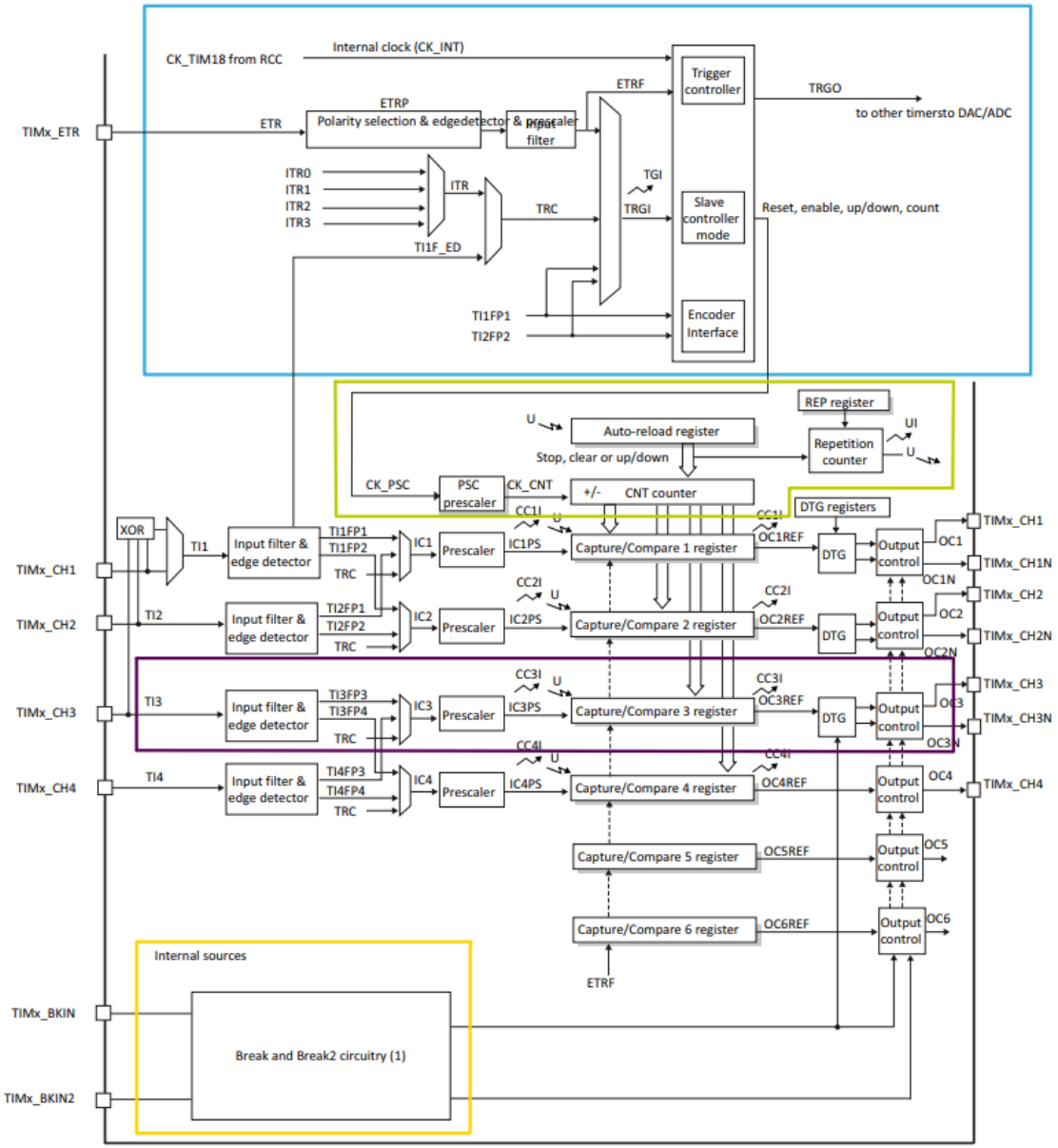


Image source : STM32 microcontroller datasheet

Further reading

- EFM8BB1 Reference Manual by Silicon Labs:
- https://www.silabs.com/documents/public/reference-manuals/efm8bb1-rm.pdf

- 8051 reference manual:
- http://ww1.microchip.com/downloads/en/DeviceDoc/doc4316.pdf