# EIE3105: Wave Generating and Capturing (Chapter 16 and 17)

Dr. Lawrence Cheung

Semester 1, 2020/21

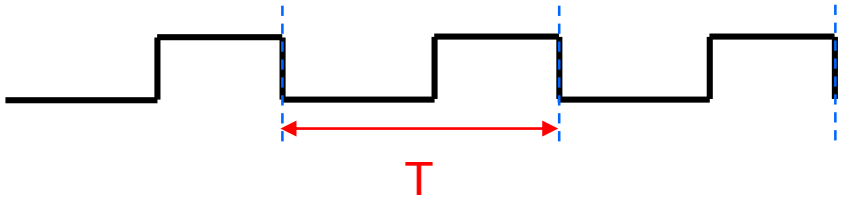
# Topics

- Wave characteristics
- Timer0 review
- Wave generating using Timer0
- Wave generating using Timer2
- Wave generating using Timer1
- Capturing

# Wave characteristics

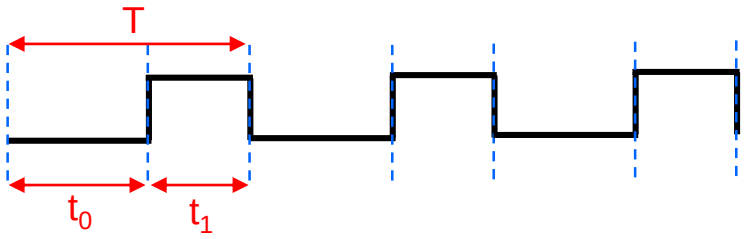
- Period
  - Frequency

$$f = \frac{1}{T}$$



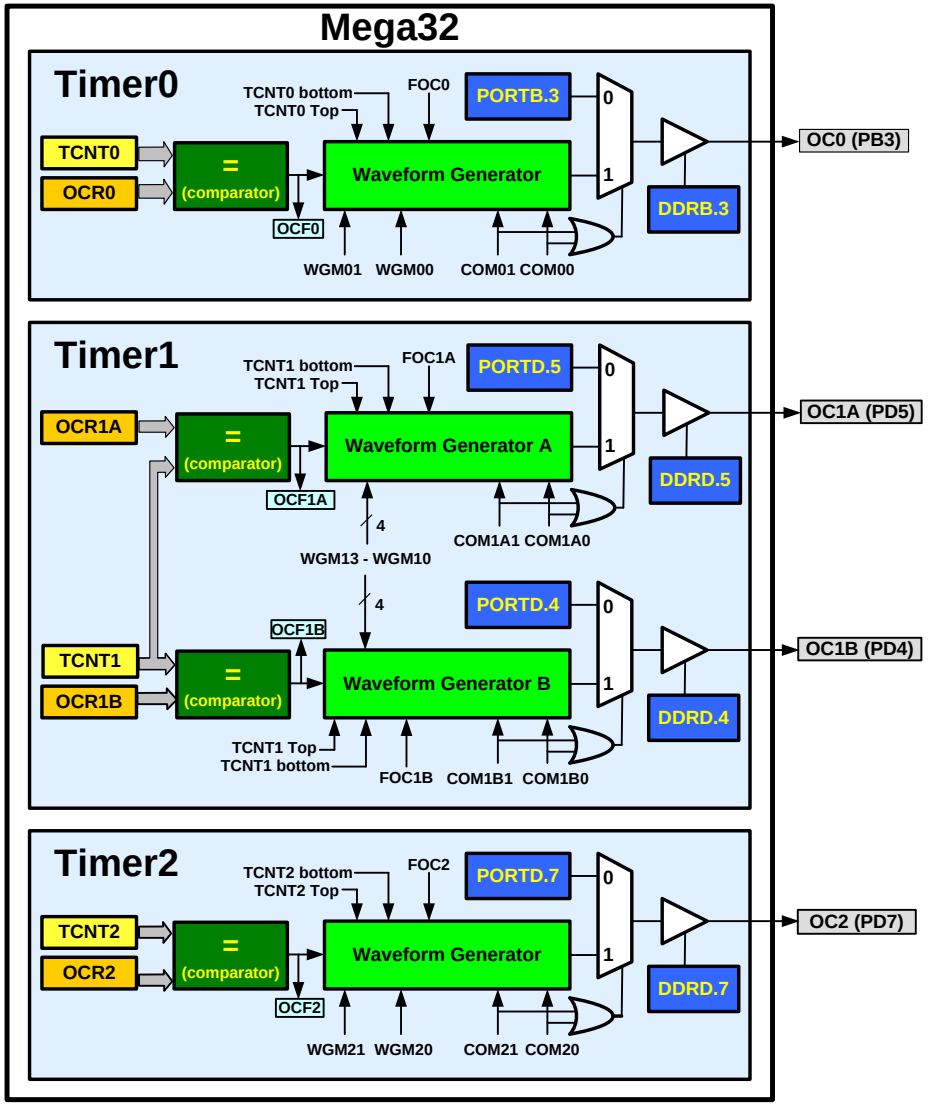

- Duty cycle

$$\text{duty cycle} = \frac{t_1}{T} \times 100 = \frac{t_1}{t_0 + t_1} \times 100$$

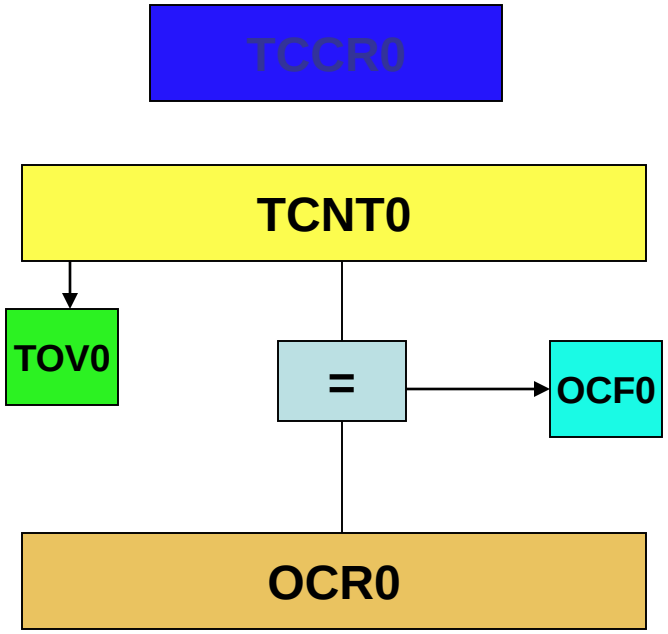



- Amplitude

# Waveform generators in ATmega32

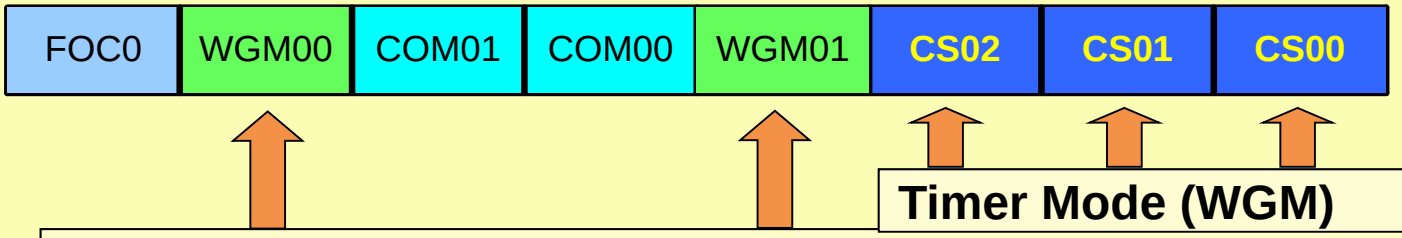


# Timer 0 Review

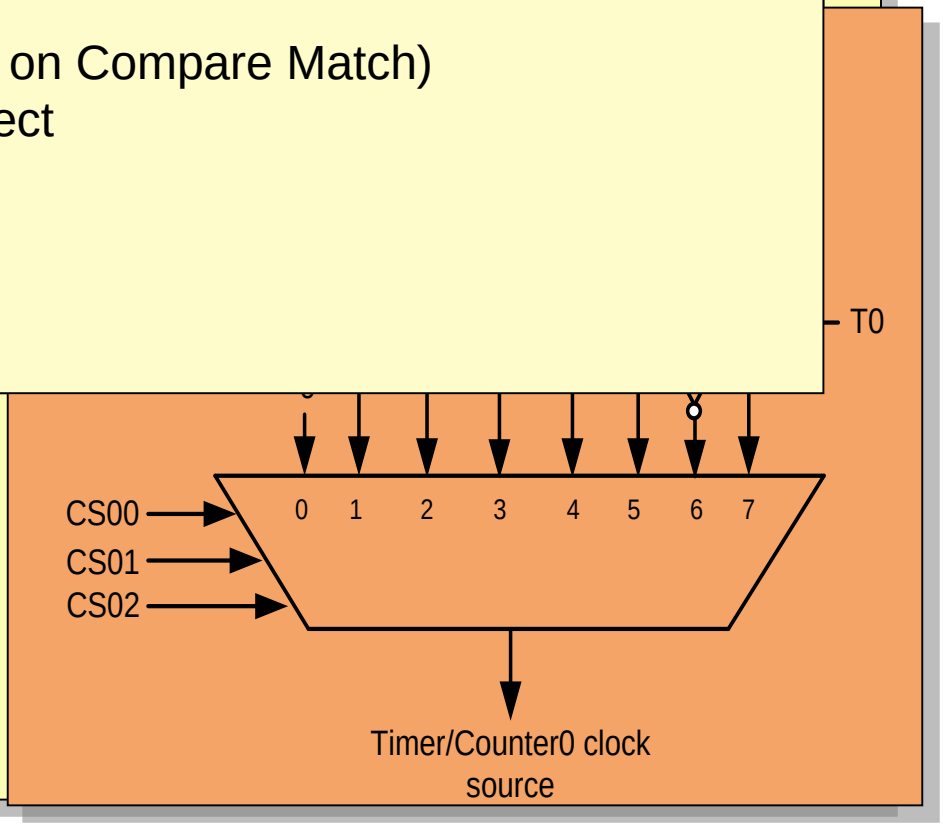



| OCF2 | TOV2 | ICF1 | OCF1A | OCF1B | TOV1 | OCF0 | TOV0 |
|---|---|---|---|---|---|---|---|

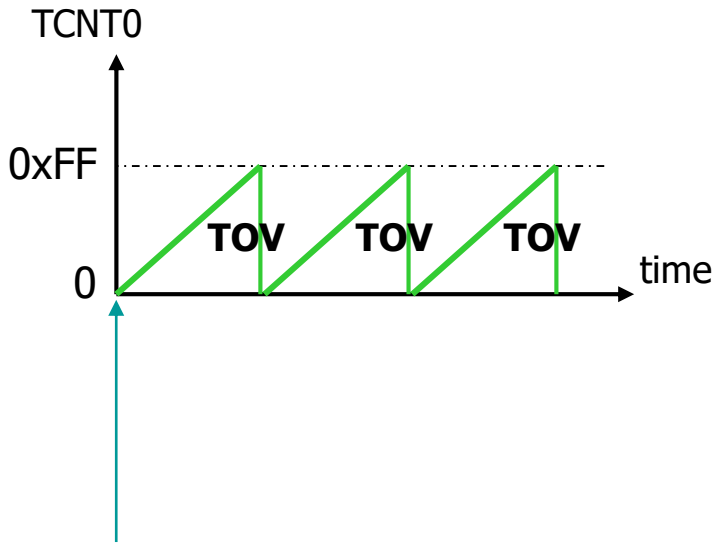
TIFR

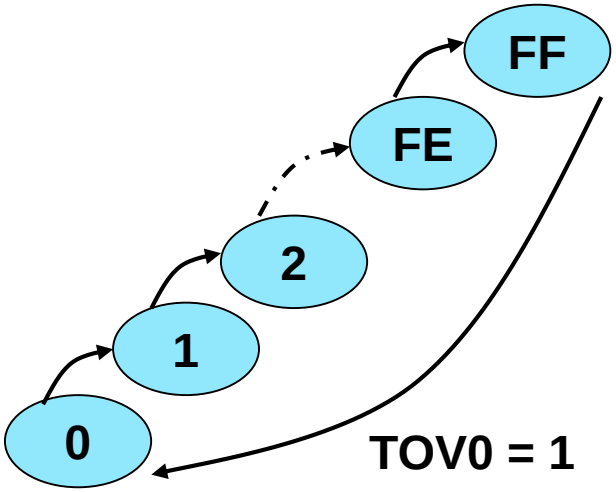



| WGM00 | WGM01 | Comment |
|---|---|---|
| 0 | 0 | Normal |
| 0 | 1 | CTC (Clear Timer on Compare Match) |
| 1 | 0 | PWM, phase correct |
| 1 | 1 | Fast PWM |



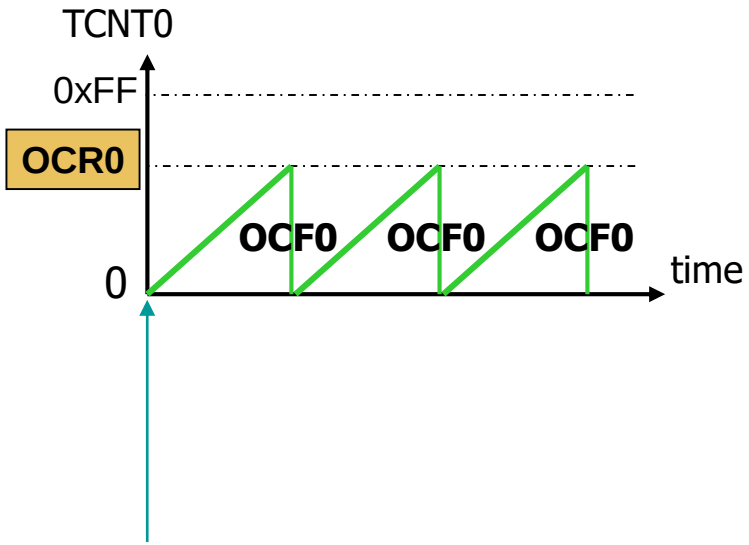
# Normal mode

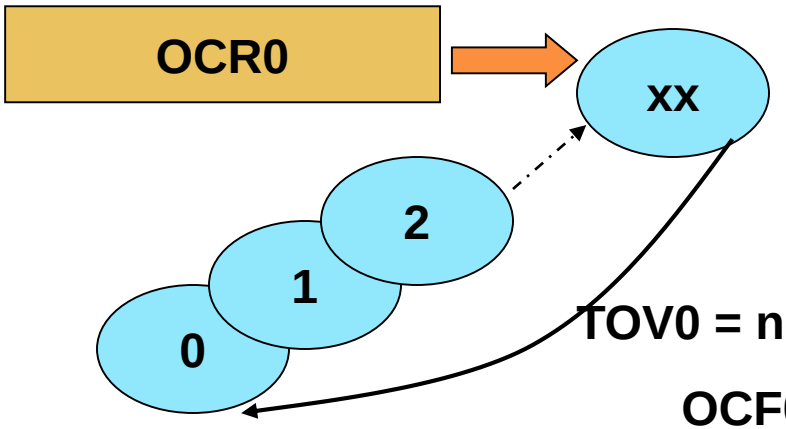



**TOV0:** **1**

# CTC mode



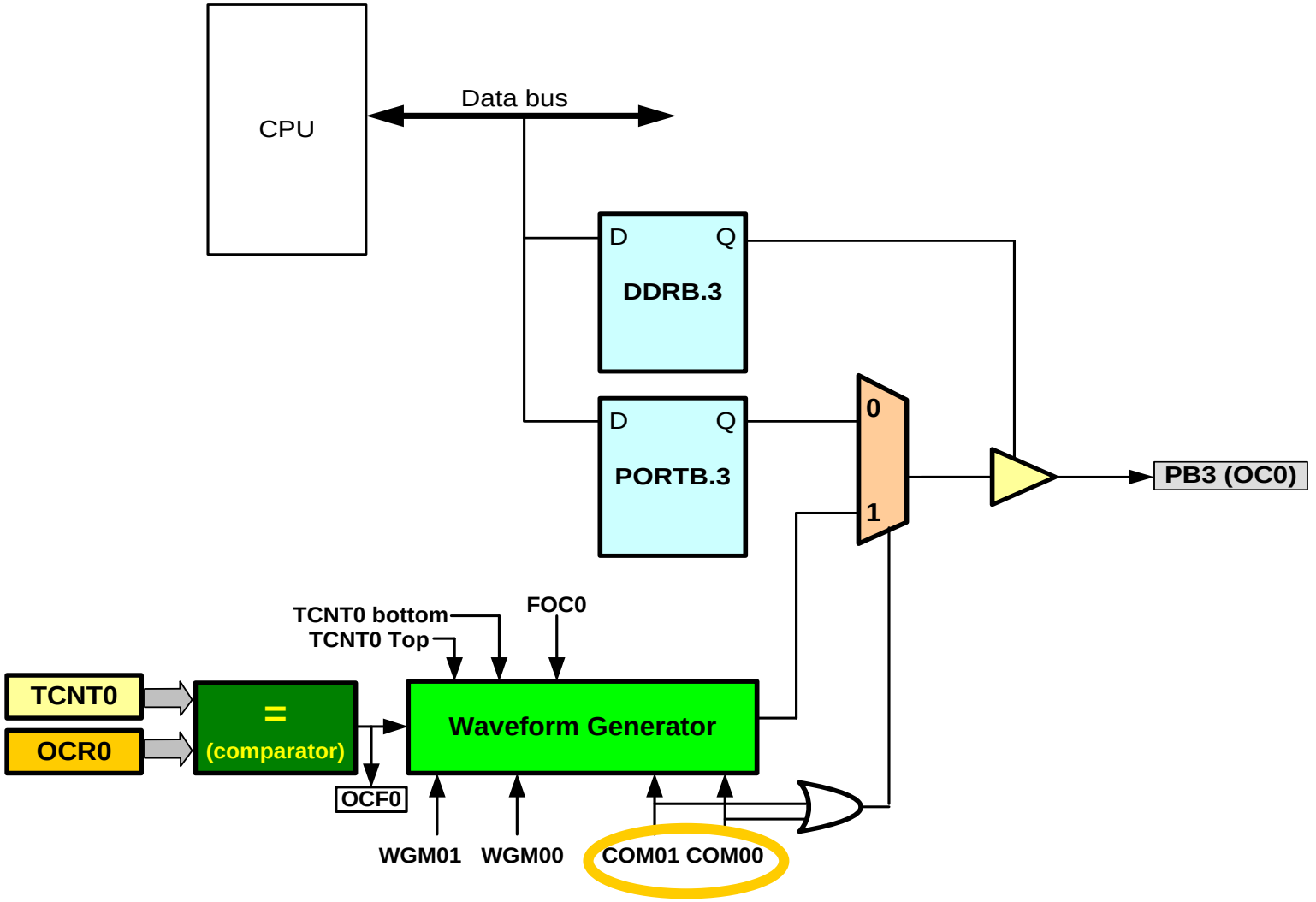

**TOV0:** **0**

**OCF0:** **1**

**TOV0 = no change**

**OCF0 = 1**

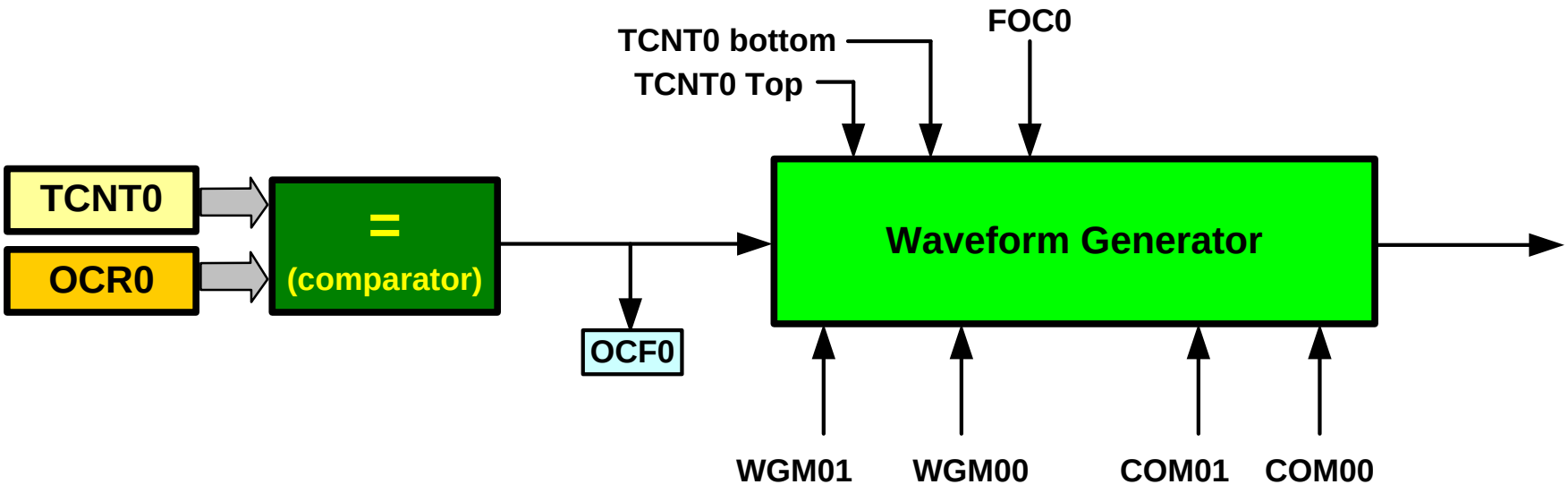
# Waveform Generator




| FOC0 | WGM00 | COM01 | COM00 | WGM01 | CS02 | CS01 | CS00 |
|---|---|---|---|---|---|---|---|

TCCR0

# Waveform Generator



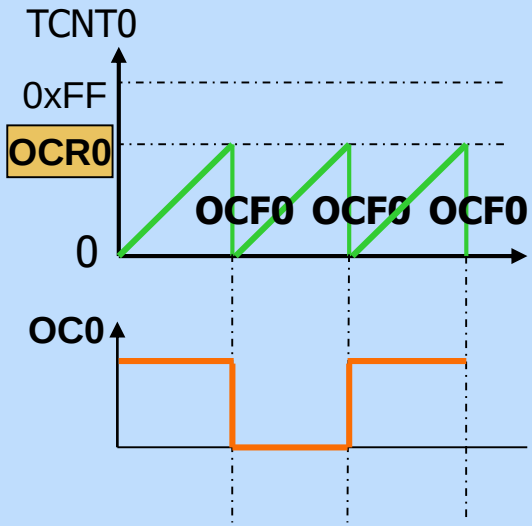

| FOC0 | WGM00 | COM01 | COM00 | WGM01 | CS02 | CS01 | CS00 |
|---|---|---|---|---|---|---|---|

TCCR0

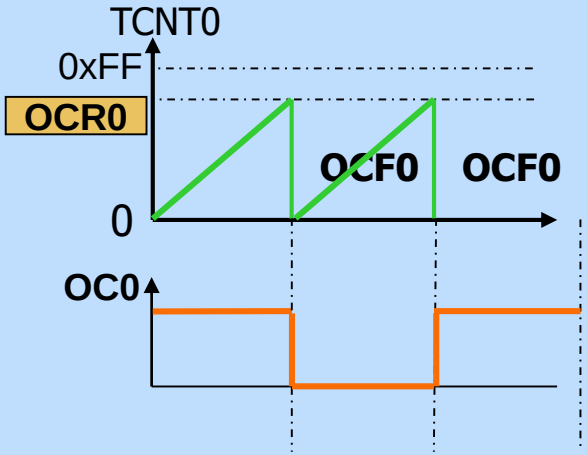
| FOC0 | WGM00 | COM01 | COM00 | WGM01 | CS02 | CS01 | CS00 |
|---|---|---|---|---|---|---|---|

TCCR0

**CTC, Toggle Mode**
**Duty cycle = 50%**
**Frequency = changeable**




$$F_{OC0} = \frac{f_{clk}}{2N(OCR0+1)}$$




(COM)

| COM | |
|---|---|
| 0 | |
| 0 | |
| 1 | |
| 1 | |

# Waveform Generator

$$F_{OC0} = \frac{f_{clk}}{2N(OCR0+1)} \Rightarrow 500KHz = \frac{8MHz}{2N(OCR0+1)} \Rightarrow N(OCR0+1) = \frac{8MHz}{1MHz}$$

$$\Rightarrow N(OCR0+1) = 8 \Rightarrow \begin{cases} N = 1 \text{ and } OCR0 = 7 \\ N = 8 \text{ and } OCR0 = 0 \end{cases}$$

Assuming XTAL = 8 MHz, make a pulse with duty cycle = 50% and frequency = 500KHz

```
LDI R20,7
OUT OCR0,R20
LDI R20,0x19
OUT TCCR0,R20
```

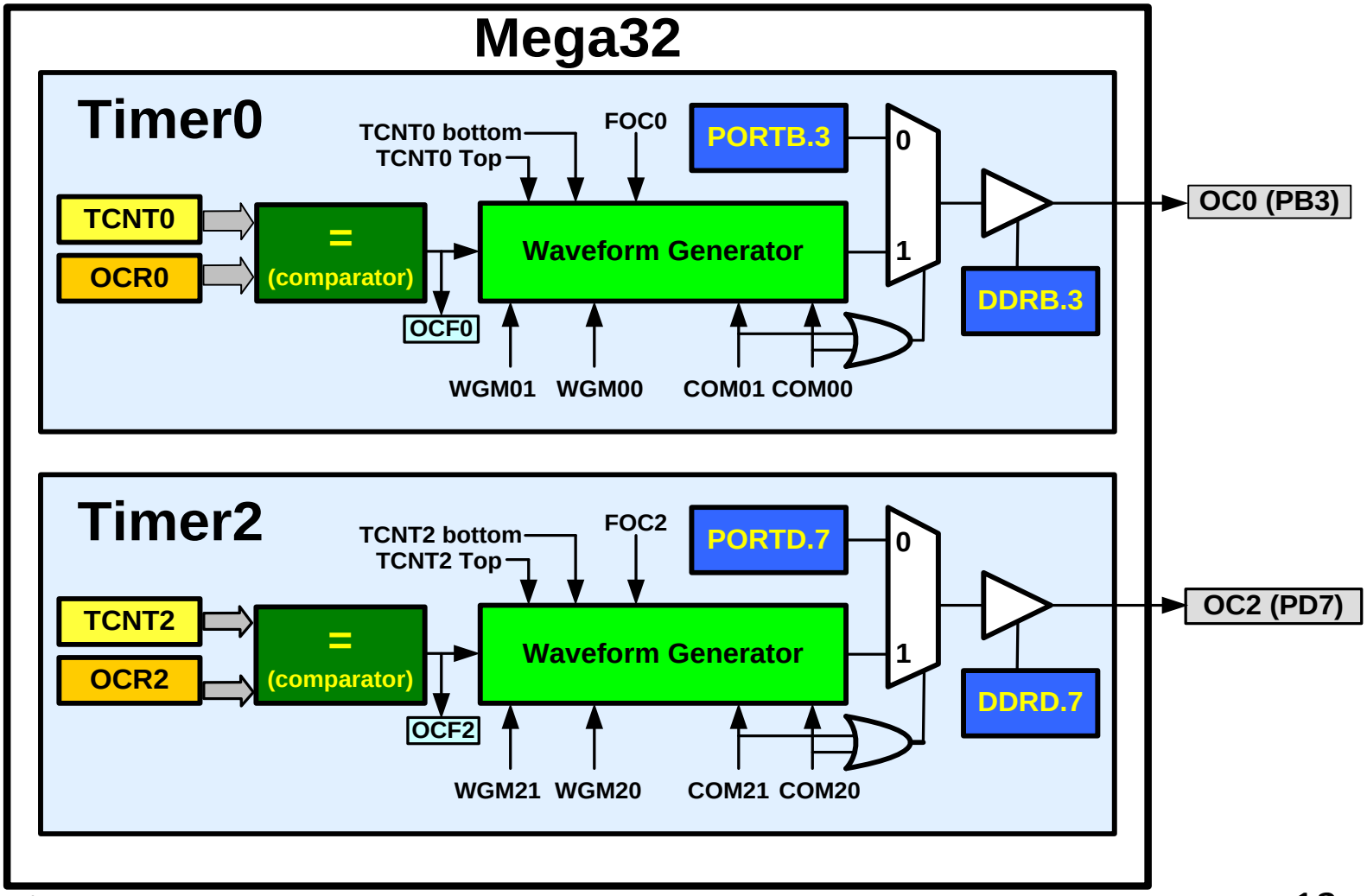
```
OCR0 = 7;
TCCR0 = 0x19; //prescaler = 1
```

```
LDI R20,0
OUT OCR0,R20
LDI R20,0x1A
OUT TCCR0,R20
```

```
OCR0 = 0;
TCCR0 = 0x1A; //prescaler = 8
```

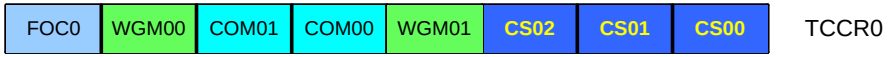
# Wave generating in Timer2

- Like Timer0



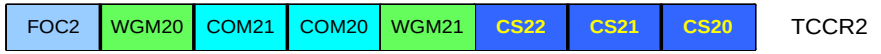
# The difference between Timer 0 and Timer 2

- Timer 0




| CS02 | CS01 | CS00 | Comment |
|---|---|---|---|
| 0 | 0 | 0 | Timer/Counter stopped |
| 0 | 0 | 1 | clk (No Prescaling) |
| 0 | 1 | 0 | clk / 8 |
| 0 | 1 | 1 | clk / 64 |
| 1 | 0 | 0 | clk / 256 |
| 1 | 0 | 1 | clk / 1024 |
| 1 | 1 | 0 | External clock (falling edge) |
| 1 | 1 | 1 | External clock (rising edge) |

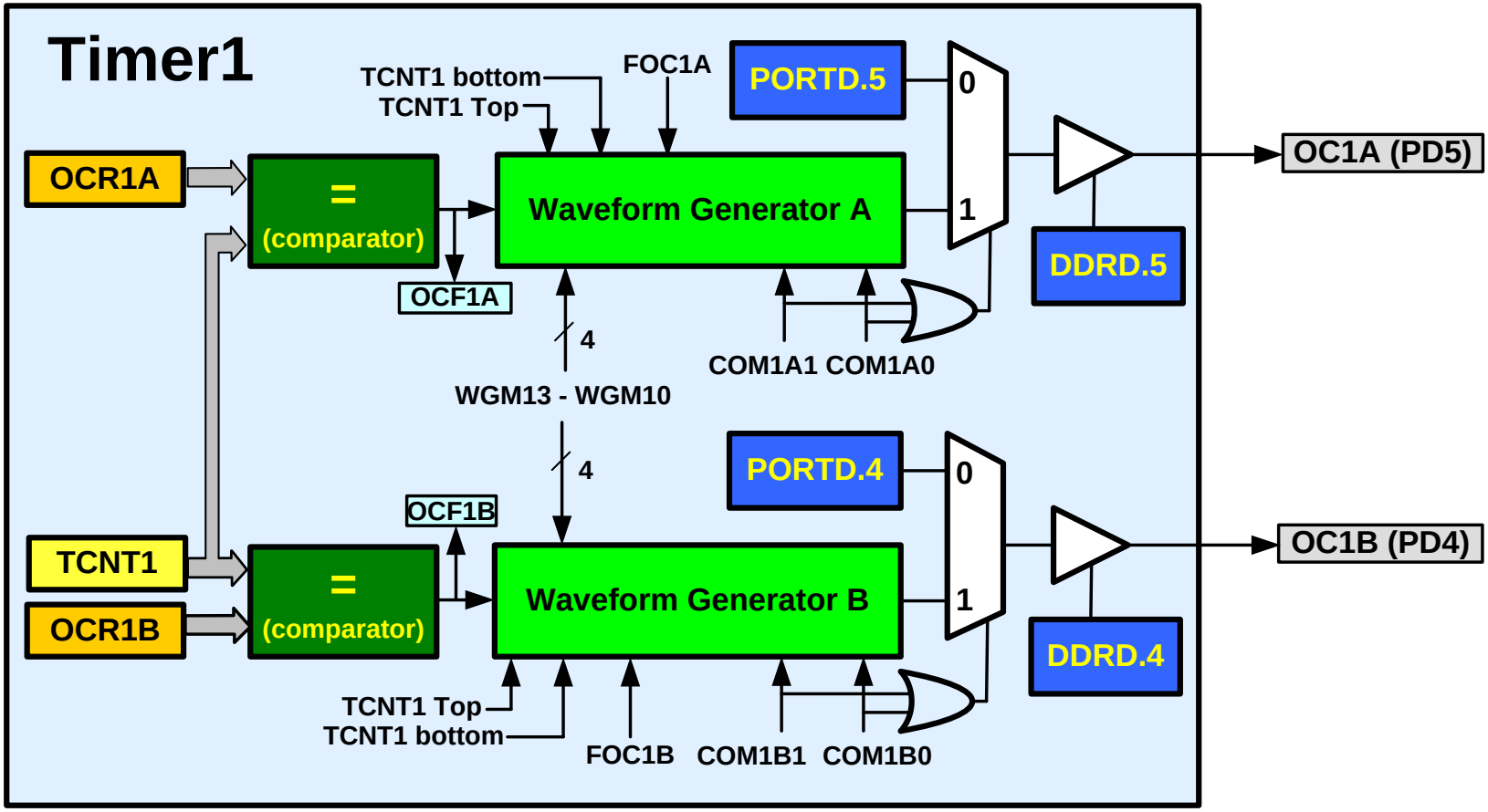
- Timer 2

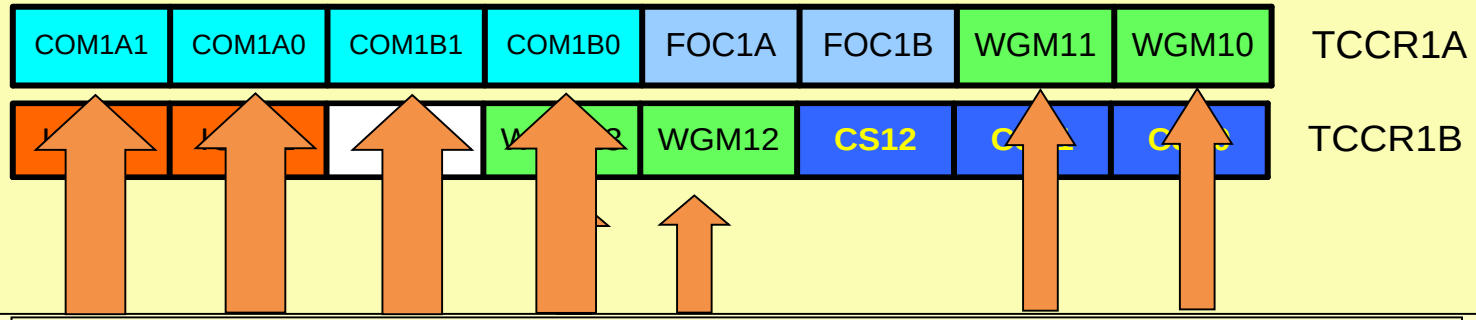



| CS22 | CS21 | CS20 | Comment |
|---|---|---|---|
| 0 | 0 | 0 | Timer/Counter stopped |
| 0 | 0 | 1 | clk (No Prescaling) |
| 0 | 1 | 0 | clk / 8 |
| 0 | 1 | 1 | clk / 32 |
| 1 | 0 | 0 | clk / 64 |
| 1 | 0 | 1 | clk / 128 |
| 1 | 1 | 0 | clk / 256 |
| 1 | 1 | 1 | clk / 1024 |

# Timer 1

- Timer 1 has two waveform generators.






**In non PWM modes**

**COM1A1:COM1A0** D7 D6 Compare Output Mode for Channel A

| COM1A1 | COM1A0 | Description |
|---|---|---|
| 0 | 0 | Normal port operation, OC1A disconnected |
| 0 | 1 | Toggle OC1A on compare match |
| 1 | 0 | Clear OC1A on compare match |
| 1 | 1 | Set OC1A on compare match |

**COM1B1:COM1B0** D5 D4 Compare Output Mode for Channel B

| COM1B1 | COM1B0 | Description |
|---|---|---|
| 0 | 0 | Normal port operation, OC1B disconnected |
| 0 | 1 | Toggle OC1B on compare match |
| 1 | 0 | Clear OC1B on compare match |
| 1 | 1 | Set OC1B on compare match |

# PWM modes in ATmega328p

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| (0x80) | COM1A1 | COM1A0 | COM1B1 | COM1B0 | – | – | WGM11 | WGM10 | TCCR1A |
| Read/Write | R/W | R/W | R/W | R/W | R | R | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| (0x81) | ICNC1 | ICES1 | – | WGM13 | WGM12 | CS12 | CS11 | CS10 | TCCR1B |
| Read/Write | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| 0x16 (0x36) | – | – | ICF1 | – | – | OCF1B | OCF1A | TOV1 | TIFR1 |
| Read/Write | R | R | R/W | R | R | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

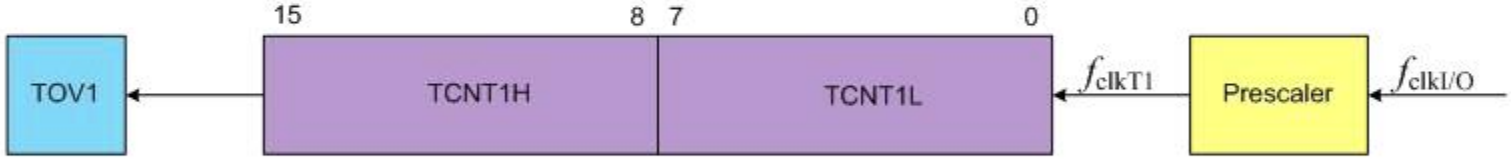



**Table 13-5.** Clock Select Bit Description

| CS12 | CS11 | CS10 | Description |
|---|---|---|---|
| 0 | 0 | 0 | No clock source (Timer/Counter stopped). |
| 0 | 0 | 1 | $clk_{I/O}$/1 (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 T1 pin. Clock on falling edge. |
| 1 | 1 | 1 | External clock source on T1 pin. Clock on rising edge. |

# PWM modes in ATmega328p

| | |
|---|---|
| BOTTOM | The counter reaches the BOTTOM when it becomes zero (0x00). |
| MAX | The counter reaches its MAXimum when it becomes 0xFF (decimal 255). |
| TOP | The counter reaches the TOP when it becomes equal to the highest value in the count sequence. The TOP value can be assigned to be the fixed value 0xFF (MAX) or the value stored in the OCR2A Register. The assignment is dependent on the mode of operation. |

# PWM modes in ATmega328p

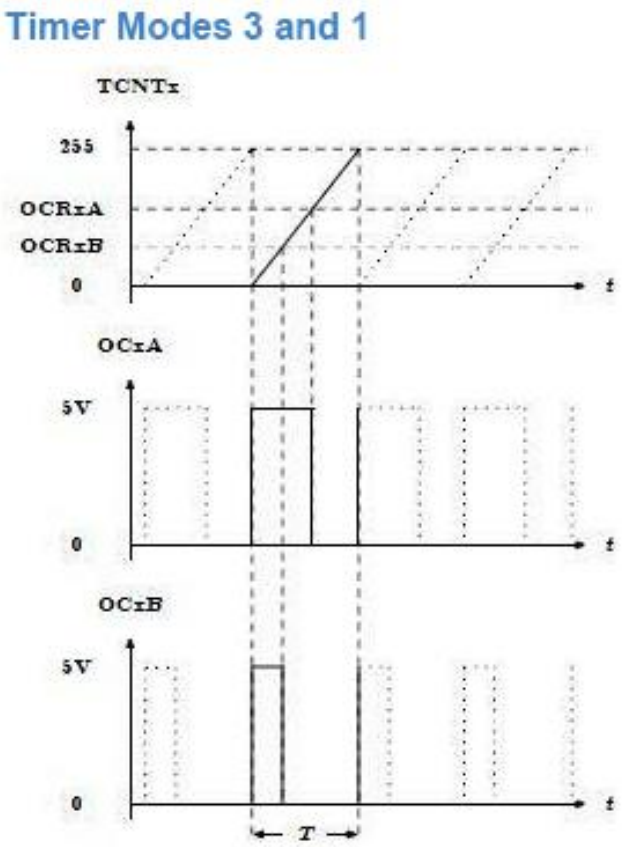
Table 1 Waveform Generation Mode Bit Description

| Mode | WGM2 | WGM1 | WGM0 | Timer/Counter Mode of Operation | TOP | Update of OCRx at | TOV Flag Set on[1][2] |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Normal | 0xFF | Immediate | MAX |
| 1 | 0 | 0 | 1 | PWM, Phase Correct | 0xFF | TOP | BOTTOM |
| 2 | 0 | 1 | 0 | CTC | OCRA | Immediate | MAX |
| 3 | 0 | 1 | 1 | Fast PWM | 0xFF | BOTTOM | MAX |
| 4 | 1 | 0 | 0 | Reserved | – | – | – |
| 5 | 1 | 0 | 1 | PWM, Phase Correct | OCRA | TOP | BOTTOM |
| 6 | 1 | 1 | 0 | Reserved | – | – | – |
| 7 | 1 | 1 | 1 | Fast PWM | OCRA | BOTTOM | TOP |

Notes:

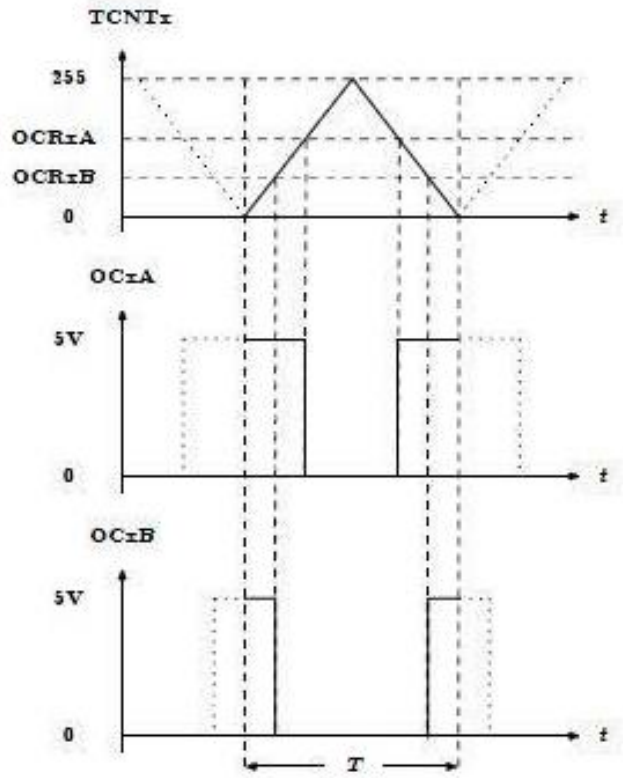
1. MAX = 0xFF
2. BOTTOM = 0x00
3. In normal operation the Timer/Counter Overflow Flag (TOV0) will be set in the same timer clock cycle as the TCNT0 becomes zero.
4. Whenever TCNT0 equals OCR0A or OCR0B, the comparator signals a match. A match will set the Output Compare Flag (OCF0A or OCF0B) at the **next timer clock cycle**.

# PWM modes in ATmega328p




$f_{OCnxPWM} = \frac{f_{clk\_I/O}}{N \cdot 256}$

(a) Fast PWM




(b) Phase-Correct PWM

$f_{OCnxPCPWM} = \frac{f_{clk\_I/O}}{N \cdot 510}$

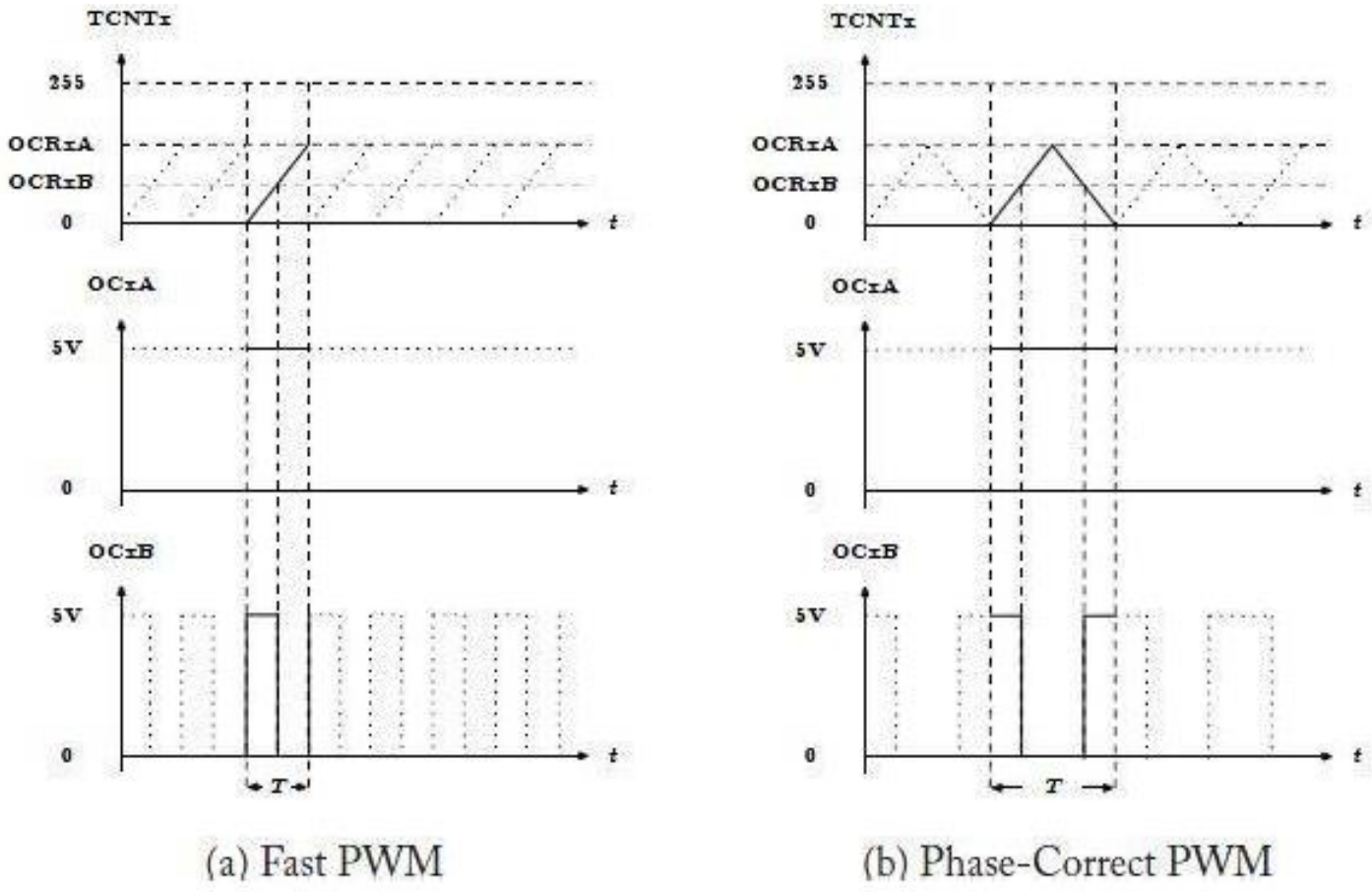
$f_{OC2B} = f_{CLK}$ / 256 = 16 MHz / 256 = 62.5 KHz ≃ 64 KHz

$f_{OC2B} = f_{CLK}$ / 510 = 16 MHz / 510 = 31.3725 MHz

For N = 1

# PWM modes in ATmega328p

Timer Modes 7 and 5



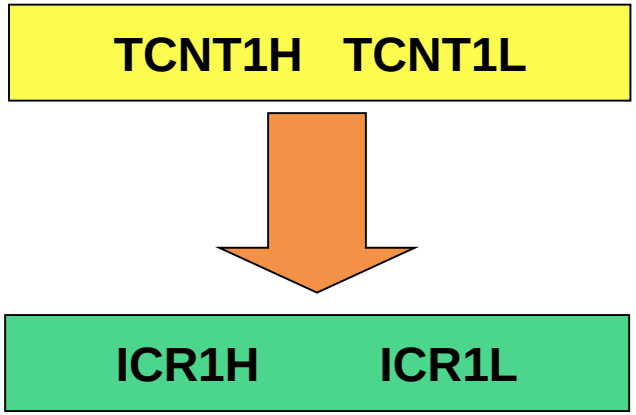

OCRxA = Frequency of oscillator / (Frequency of generated wave × N) – 1
(N = prescaler)
Duty cycle = (OCRxB + 1) / (OCRxA + 1) × 100%.

# PWM modes in ATmega328p

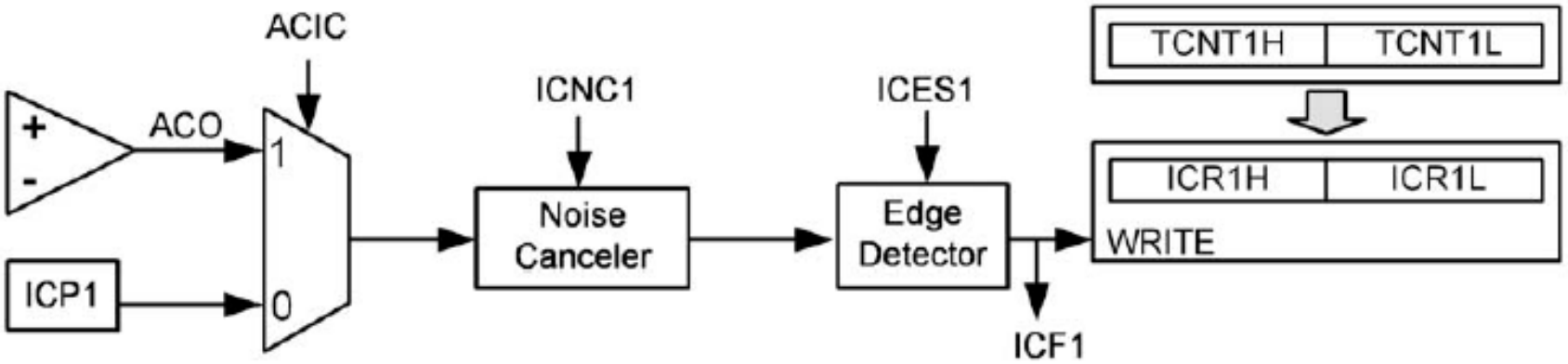
```
#include "avr/io.h"
int main(void)
{
        TCCR0A = (1 << COM0A1) | (0 << COM0A0) | //00
                 (1 << COM0B1) | (0 << COM0B0) |
                 (1 << WGM01)  | (1 << WGM00);
        TCCR0B = (1 << WGM02) |
                 (0 << CS02)   | (0 << CS01)   | (1 << CS00);

        OCR0A = 249;  //64kHz
        OCR0B = 49;   //20% duty cycle
        DDRD = 0b00100000; // PD5 (OC0B)

        while(1);
}
```

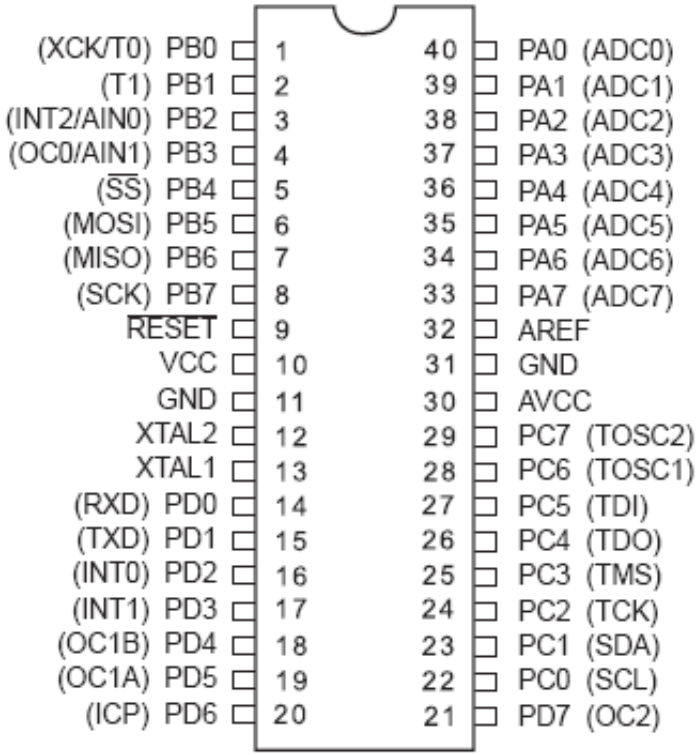
# Capturing

- Usages
  - Measuring duty cycle
  - Measuring period

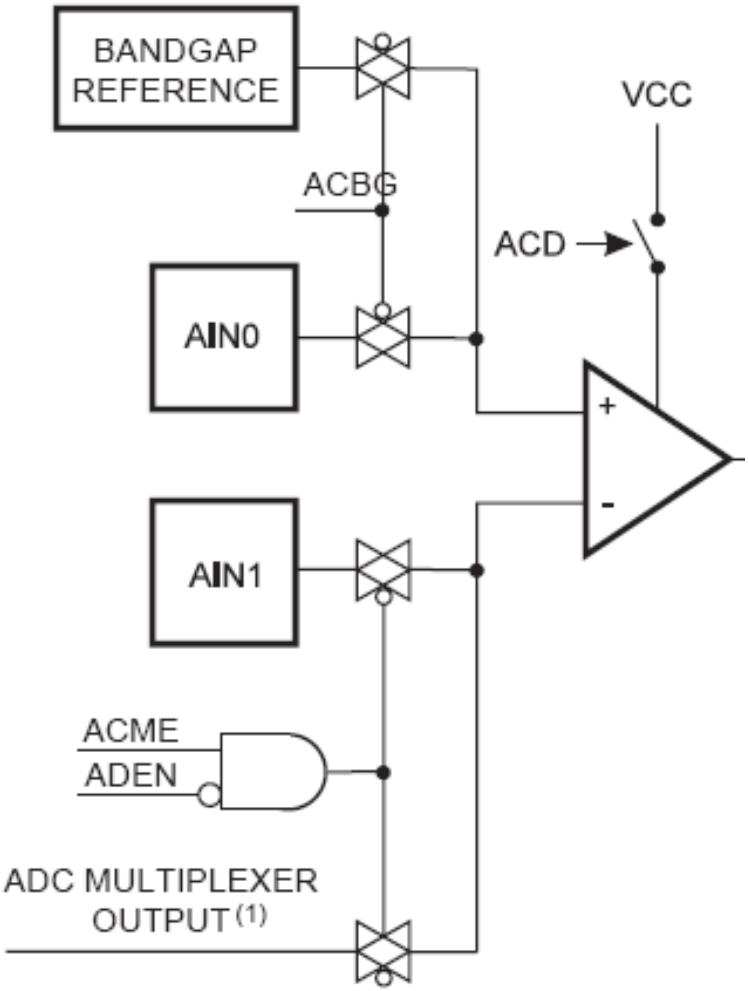


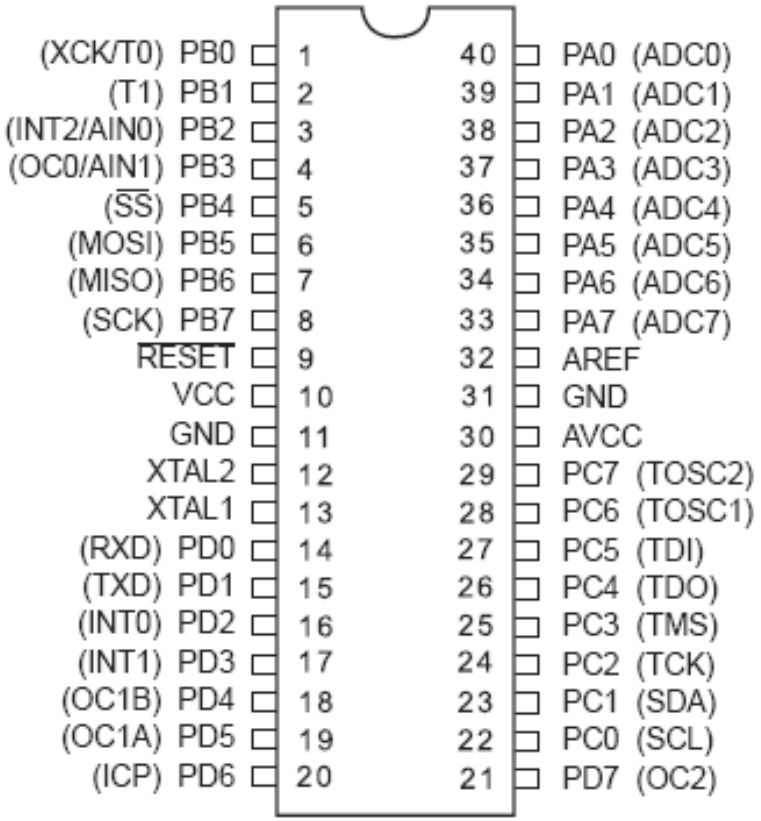
# Capturing


ACIC
ACO
+
-
1
0
ICP1
ICNC1
Noise Canceler
ICES1
Edge Detector
TCNT1H
TCNT1L
ICR1H
ICR1L
WRITE
ICF1

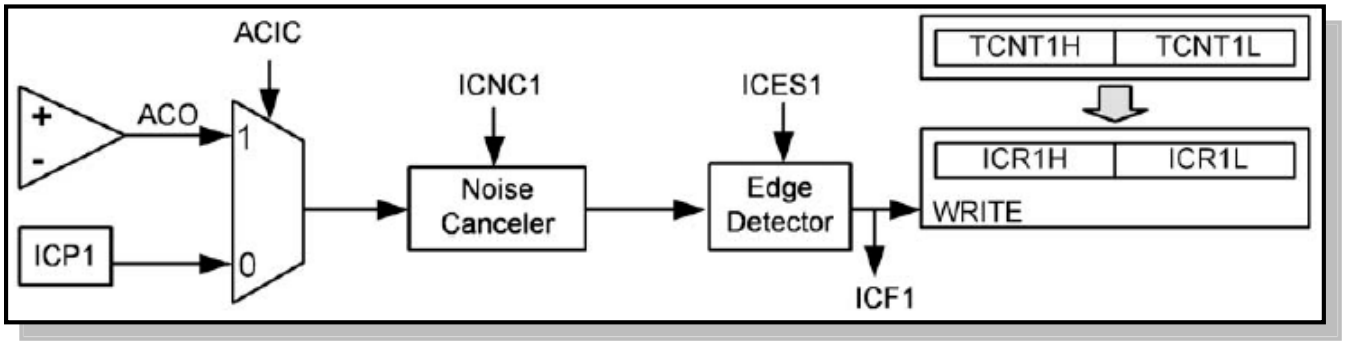

(XCK/T0) PB0 1
(T1) PB1 2
(INT2/AIN0) PB2 3
(OC0/AIN1) PB3 4
(SS) PB4 5
(MOSI) PB5 6
(MISO) PB6 7
(SCK) PB7 8
RESET 9
VCC 10
GND 11
XTAL2 12
XTAL1 13
(RXD) PD0 14
(TXD) PD1 15
(INT0) PD2 16
(INT1) PD3 17
(OC1B) PD4 18
(OC1A) PD5 19
(ICP) PD6 20
40 PA0 (ADC0)
39 PA1 (ADC1)
38 PA2 (ADC2)
37 PA3 (ADC3)
36 PA4 (ADC4)
35 PA5 (ADC5)
34 PA6 (ADC6)
33 PA7 (ADC7)
32 AREF
31 GND
30 AVCC
29 PC7 (TOSC2)
28 PC6 (TOSC1)
27 PC5 (TDI)
26 PC4 (TDO)
25 PC3 (TMS)
24 PC2 (TCK)
23 PC1 (SDA)
22 PC0 (SCL)
21 PD7 (OC2)

# Comparator




Copied from ATmega32 datasheet page 196




| ICNC1 | ICES1 | - | WGM13 | WGM12 | CS12 | CS11 | CS10 |
|---|---|---|---|---|---|---|---|

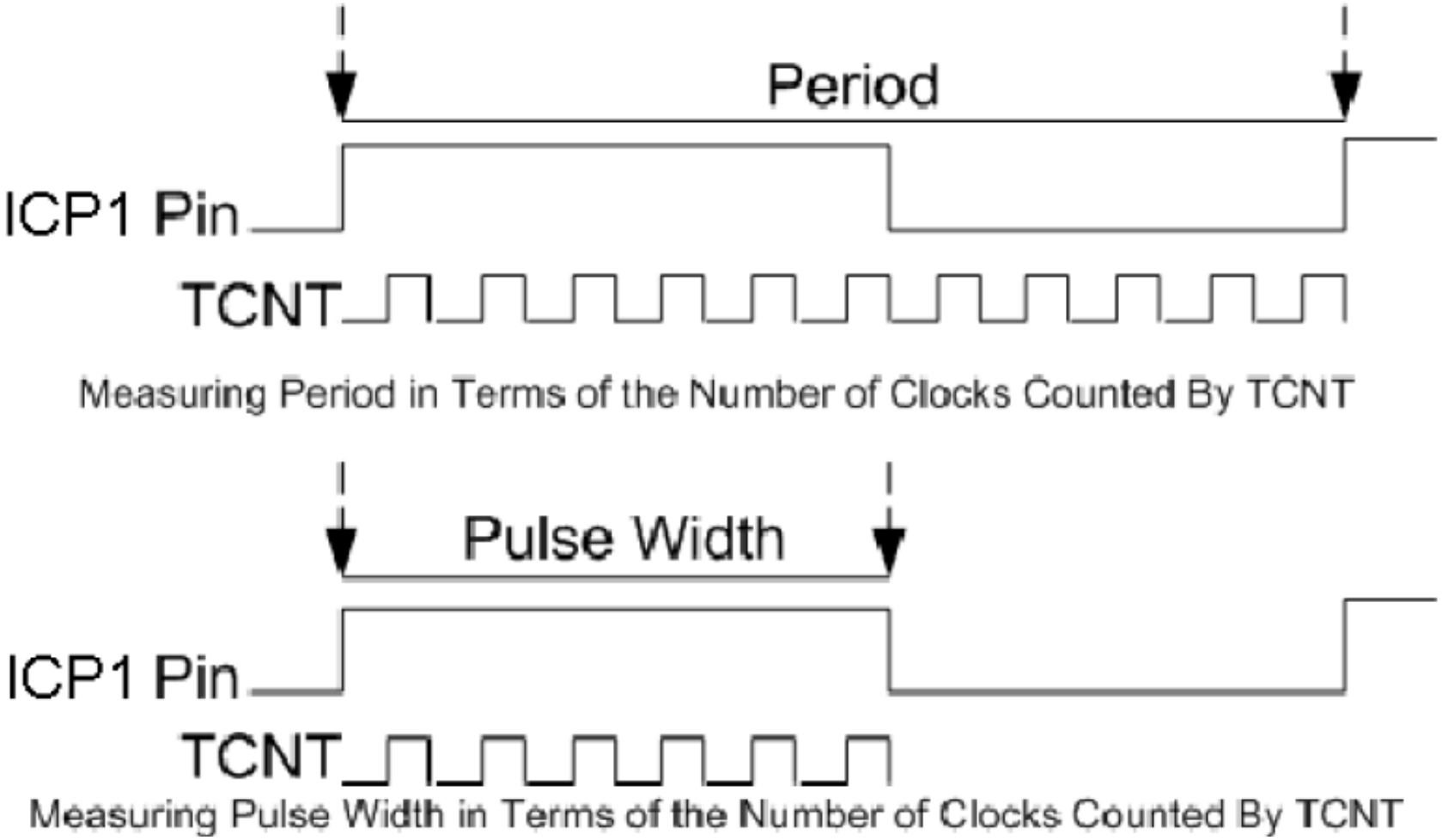
TCCR1B

**ICNC1:** Input Capture Noise Canceller
0:disabled
1:Enabled (captures after 4 successive equal valued samples)
**ICSES1:** Input Capture Edge Select
0: Falling edge
1: Rising edge

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|
| ACD | ACBG | ACO | ACI | ACIE | ACIC | ACIS1 | ACIS0 | ACSR |
| R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | |

**ACIC:** Analog Comparator Input Capture Enable
0: ICP1 provides the capture signal
1: analog comparator is connected to the capturer

# Measuring duty cycle and period



# Capturing in ATmega328p

```
#include “avr/io.h”
int main()
{
        //measure the pulse width of a pulse
        unsigned char t1;
        DDRD = 0xFF;      //PORTD as output
        PORTB = 0xFF;
        TCCR1A = 0;       //Timer Mode = Normal
        TCCR1B = (1 <<ICES1) | (1 << CS12) | (0 << CS11) | (0 << CS10);
        //rising edge, prescaler = 256, no noise canceller

        TIFR1 = (1<<ICF1);         //clear ICF1 (The Input Capture Flag)
        while ((TIFR1&(1<<ICF1)) == 0); //wait while ICF1 is clear
        t1 = ICR1L;       //first edge value (ICR, low byte)
        TIFR1 = (1<<ICF1);         //clear ICF1
        TCCR1B = (0 <<ICES1) | (1 << CS12) | (0 << CS11) | (0 << CS10);
        //falling edge
        while ((TIFR1&(1<<ICF1)) == 0); //wait while ICF1 is clear
        PORTD = ICR1L – t1;        //pulse width = falling - rising
        TIFR1 = (1<<ICF1);         //clear ICF1
        while (1);
}
```

# Capturing in ATmega328p

```
#include “avr/io.h”
int main()
{
        //measure the period of a pulse
        unsigned char t1;
        DDRD = 0xFF;      //PORTD as output
        PORTB = 0xFF;
        TCCR1A = 0;       //Timer Mode = Normal
        TCCR1B = (1 <<ICES1) | (1 << CS12) | (0 << CS11) | (0 << CS10);
        //rising edge, prescaler = 256, no noise canceller

        TIFR1 = (1<<ICF1);         //clear ICF1 (The Input Capture Flag)
        while ((TIFR1&(1<<ICF1)) == 0); //wait while ICF1 is clear
        t1 = ICR1L;       //first edge value (ICR, low byte)
        TIFR1 = (1<<ICF1);         //clear ICF1
        while ((TIFR1&(1<<ICF1)) == 0); //wait while ICF1 is clear
        PORTD = ICR1L - t1;        //period = second edge - first edge
        TIFR1 = (1<<ICF1);         //clear ICF1
        while (1);
}
```

# Reference Readings

- Chapter 16 and 17 – *The AVR Microcontroller and Embedded Systems : Using Assembly and C*, M. A. Mazidi, S. Naimi, and S. Naimi, Pearson, 2014.

End