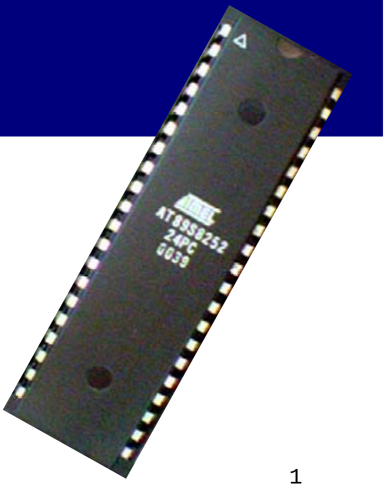
ATmega16A Microcontroller



Timers

Timers

- Timer 0,1,2
- 8 bits or 16 bits
- Clock sources:
  - □ Internal clock,
  - □ Internal clock with prescaler,
  - □ External clock (timer 2),
  - □ Special input pin

Features

- The choice of timers clock frequency (prescaler)
- Read / write counter status
- Waveform generation using a comparison register
- Frequency adjustment, PWM (Pulse Width Modulation)
- Interrupt request generation at regular intervals
- Triggered by an external event (capture)

Usage

- Wave generation
- Accurate program execution timing (event management),
- Signal timing measurement

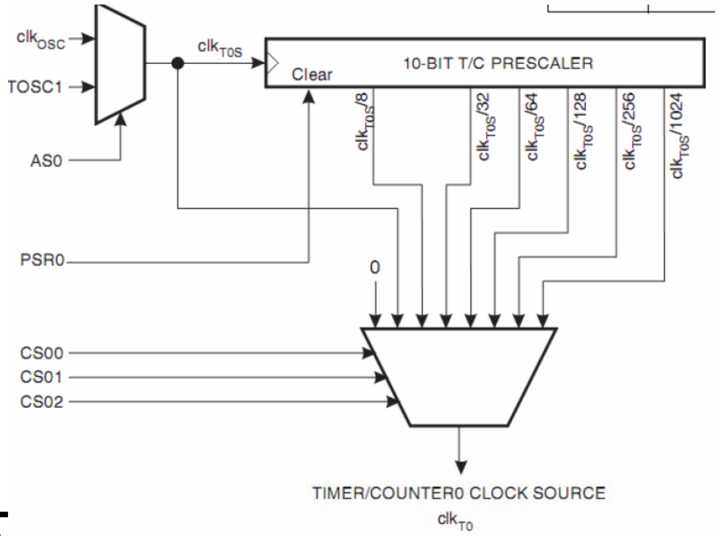
Configuration

Bit	7	6	5	4	3	2	1	0	
	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	TCCR0
Read/Write	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	

- Register **TCCRn** controls the timer behavior
  - **FOC0: Force Output Compare**
  - **WGM0[1:0]: Waveform Generation Mode**
  - **COM01:0: Compare Match Output Mode**
  - **CS02:0: Clock Select**

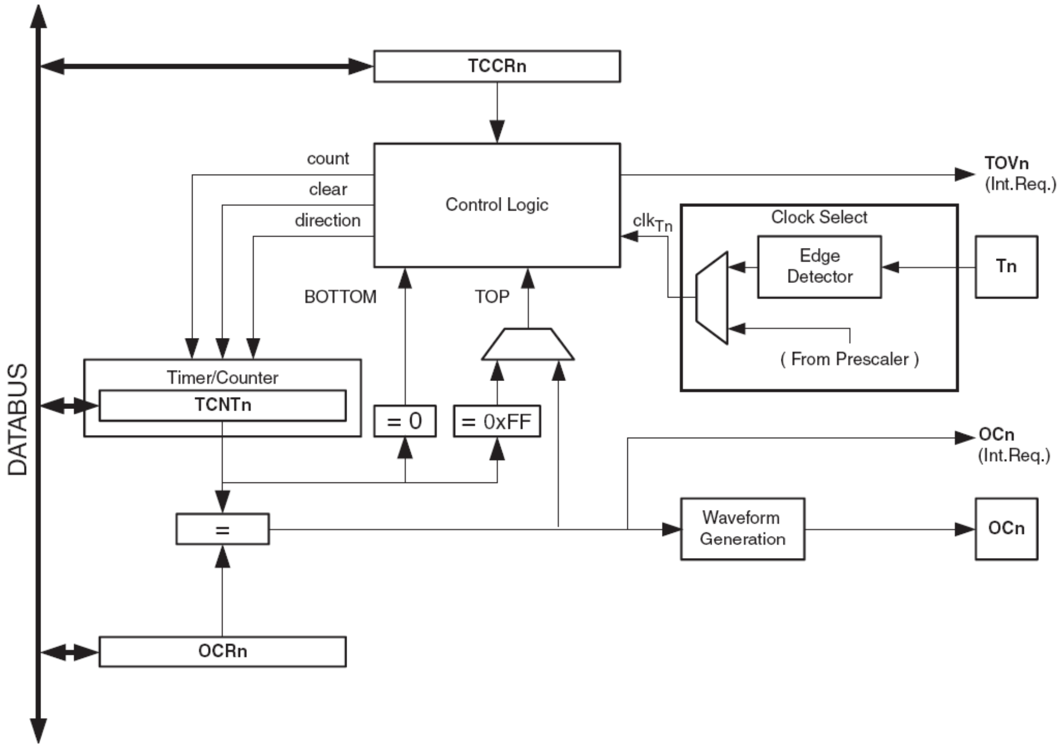
Clock selection

- Bits CS02: CS00
  - select clock frequency dividing at the counter input

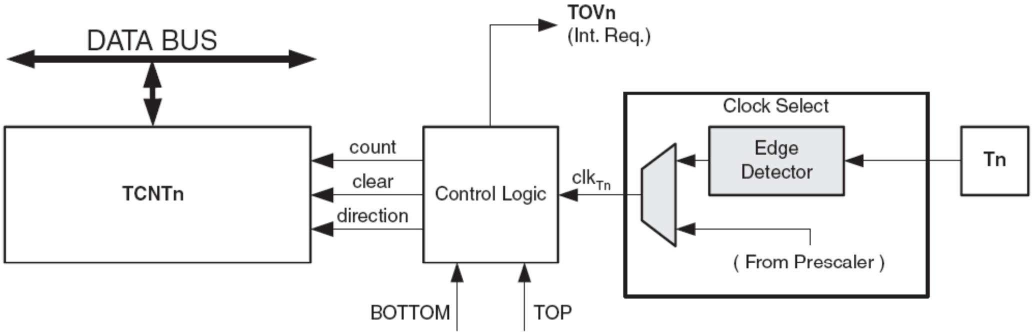


CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$clk_{I/O}$/(No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T0 pin.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

8-bit Timer/Counter Block Diagram

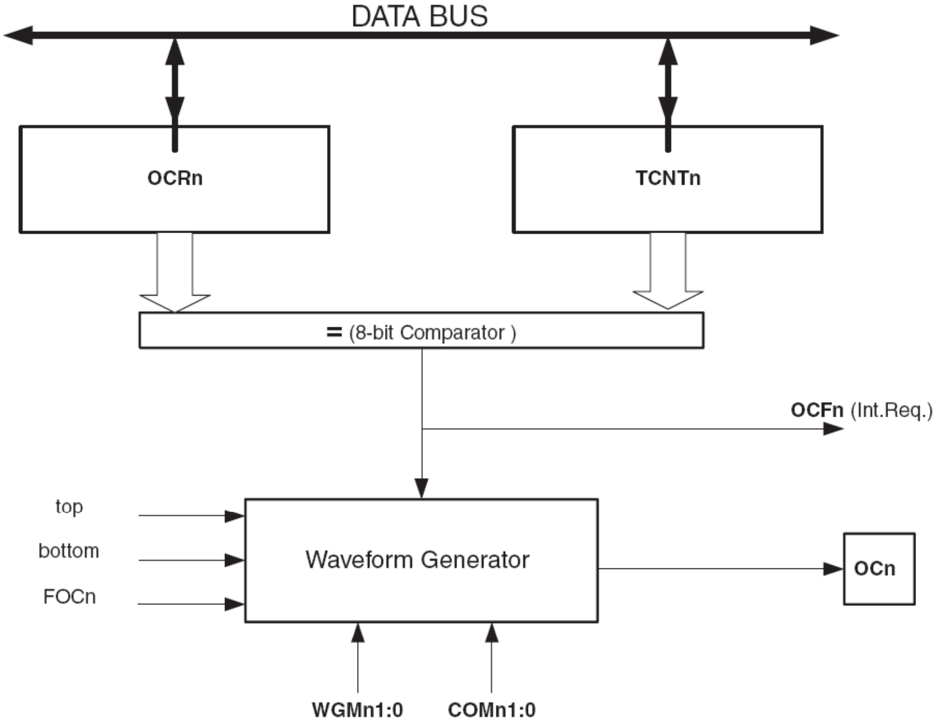


Counter Unit

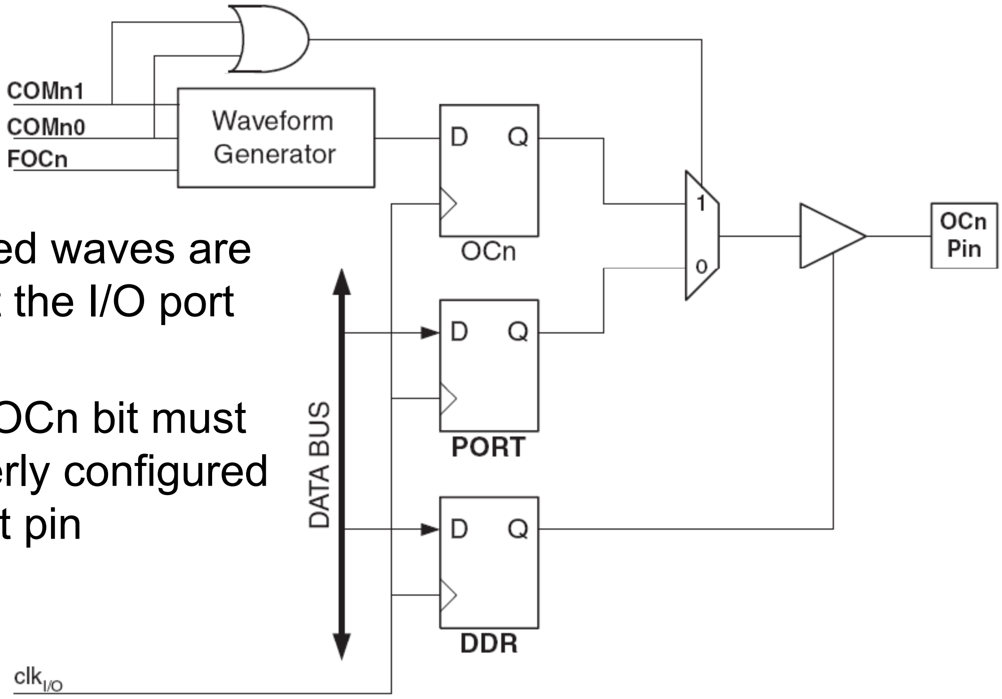


- **count** - Increment or decrement TCNT0 by 1.
- **direction** - Select between increment and decrement
- **clear** - Clear TCNT0
- **clk$_{Tn}$** – Timer/Counter clock
- **TOP** - Signalize that TCNT0 has reached maximum value
- **BOTTOM** - Signalize that TCNT0 has reached minimum value (zero).

Output Compare Unit



Output Compare Unit



- Generated waves are visible at the I/O port pins
- I/O port OCn bit must be properly configured as output pin

Modes of Operation

Mode	WGM01:00	Mode
0	00	Normal
1	01	PWM, Phase Correct
2	10	CTC
3	11	Fast PWM

The mode of operation is defined by the combination of the Waveform Generation mode (WGM01:0) and Compare Output mode (COM01:0) bits

Normal, CTC

COM0[1:0]	Description
00	Normal, OC0 disconnected.
01	Toggle OC0 on compare match
10	Clear OC0 on compare match
11	Set OC0 on compare match

PWM, Phase Correct

COM0[1:0]	Description
00	Normal, OC0 disconnected.
01	Reserved
10	Clear OC0 on compare match when up-counting. Set OC0 on compare match when downcounting
11	Set OC0 on compare match when up-counting. Clear OC0 oncompare match when downcounting

Fast PWM

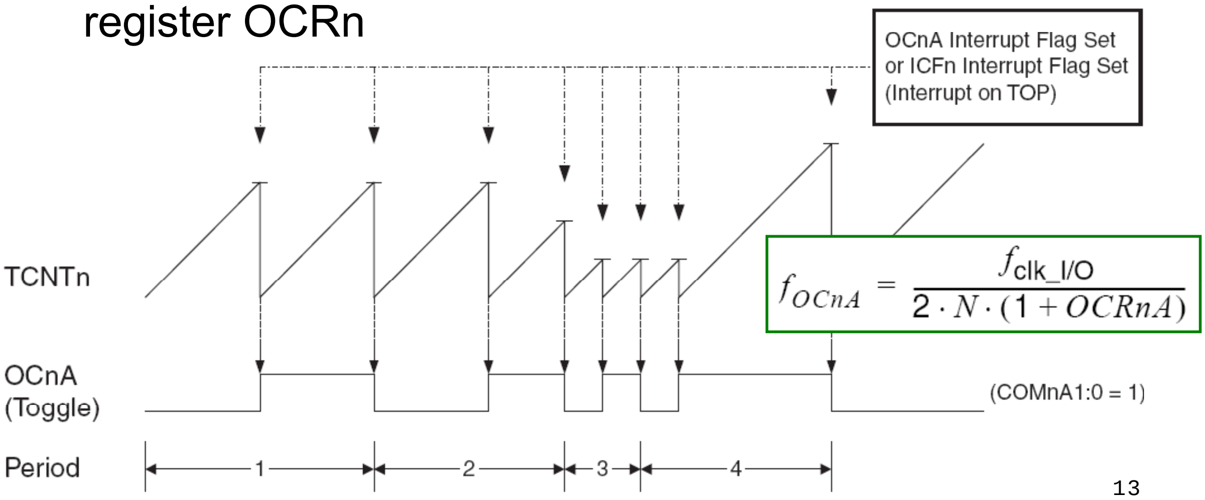
COM0[1:0]	Description
00	Normal, OC0 disconnected.
01	Reserved
10	Clear OC0 on compare match, set OC0 at BOTTOM
11	Set OC0 on compare match, clear OC0 at BOTTOM

Modes of Operation

- **Normal Mode**
  - □ The counting direction is always up (incrementing)
    - 8 bits (Timer 0,2) between 0 to 255
    - 16 bits (Timer 1) between 0 to 65535
  - □ The counter simply overruns when it passes its maximum value and then restarts from the bottom.
    - Timer overflow interrupt appears

Modes of Operation

- **CTC – Clear Timer on Compare Match**
  - □ When the counter value (TCNTn) reaches the value of OCRn, reset occurs
  - □ Generated wave frequency can be adjusted by writing register OCRn

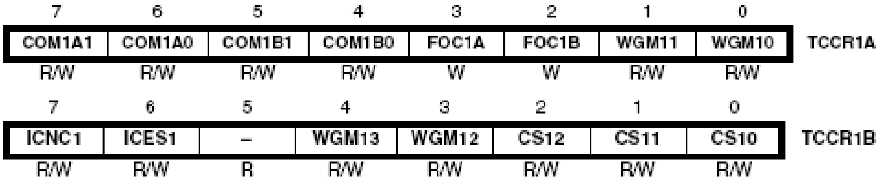


$$f_{OCnA} = \frac{f_{\text{clk\_I/O}}}{2 \cdot N \cdot (1 + OCRnA)}$$

CTC – Clear Timer on Compare Match

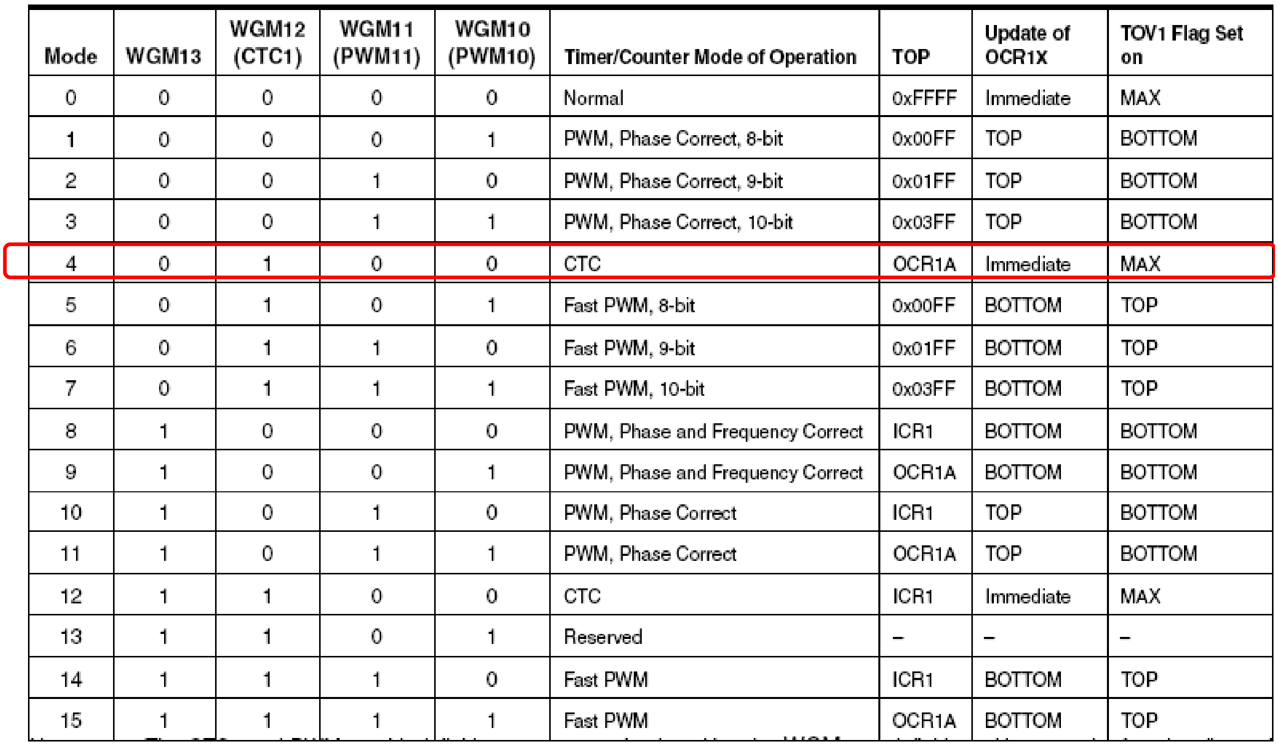
- **Example:**
- Timer1 in CTC mode (*Clear Timer on Compare Match*)
  - □ the selected clock source increments the timer
  - □ the current value is held in TCNT1 (starts at 0)
  - □ when TCNT1 = OCRA1, an interrupt is issued and the timer is reset
  - □ by choosing OCR1A and the clock frequency, the timer can be programmed for any time interval

Timer1 registers



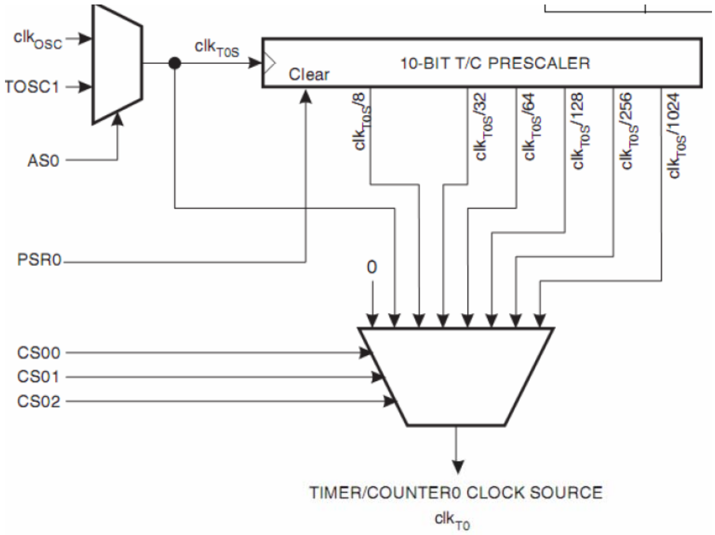
Bit	7	6	5	4	3	2	1	0	
	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	TCCR1A
	R/W	R/W	R/W	R/W	W	W	R/W	R/W	

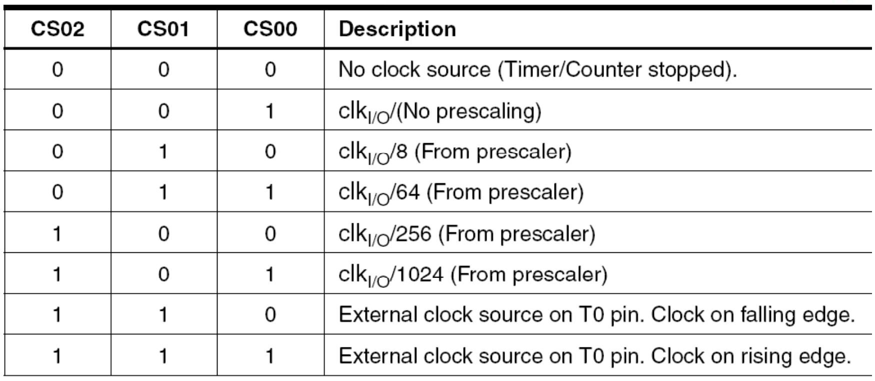
Bit	7	6	5	4	3	2	1	0	
	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	



Mode	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	Timer/Counter Mode of Operation	TOP	Update of OCR1X	TOV1 Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCR1A	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICR1	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCR1A	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICR1	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCR1A	TOP	BOTTOM
12	1	1	0	0	CTC	ICR1	Immediate	MAX
13	1	1	0	1	Reserved	–	–	–
14	1	1	1	0	Fast PWM	ICR1	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCR1A	BOTTOM	TOP

Prescaler clock division





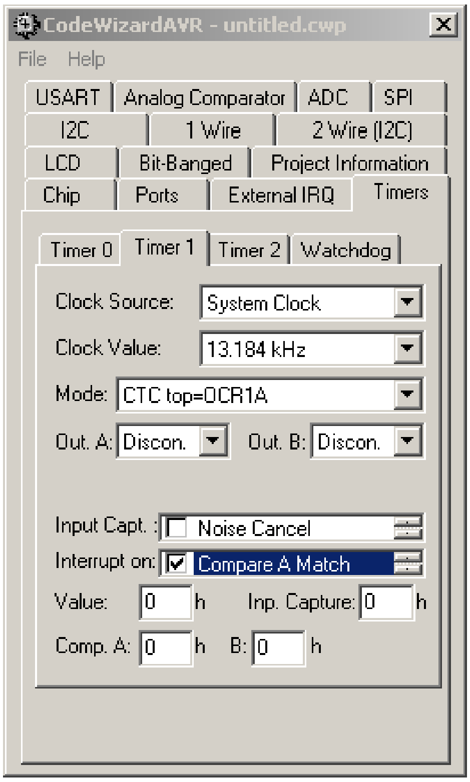
CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	clk$_{I/O}$/(No prescaling)
0	1	0	clk$_{I/O}$/8 (From prescaler)
0	1	1	clk$_{I/O}$/64 (From prescaler)
1	0	0	clk$_{I/O}$/256 (From prescaler)
1	0	1	clk$_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T0 pin.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

Example:

- Timer 1 intrerrupt generated each 1s
  - □ 1 s => low frequency
  - □ $f_{\text{cuarț}}$ =13.5MHz→division by 13,500,000 > 65536 (16 bits) → impossible
  - □ we need the prescaler to divide some more
  - □ prescaler: max divisor = 1024; 13.5MHz / 1024 = 13.184 kHz
  - □ we want 1Hz: we div ide again by 13184 = 3380h
  - □ OCR1AH = 33h, OCR1AL = 80h
  - □ we select the CTC mode; let’s set the remaining registers
  - □ from the 2 previous tables: TCCR1A = 0 and
  - □ TCCR1B= 00001101 = 0Dh

Good News !

- All these calculations can be done using *CodeWizard*
- You still need to read the *datasheet* for the explanation of the different modes



Clock Source: System Clock

Clock Value: 13.184 kHz

Mode: CTC top=OCR1A

Out. A: Discon. Out. B: Discon.

Input Capt.: □ Noise Cancel

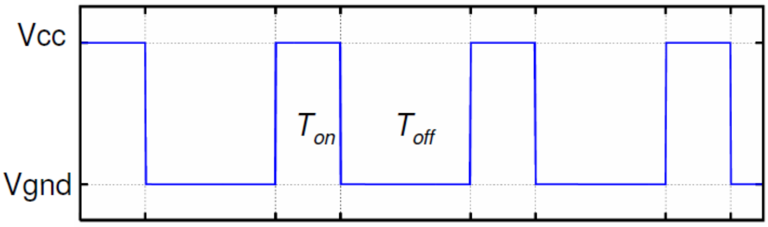
Interrupt on: ☑ Compare A Match

Value: 0 h Inp. Capture: 0 h

Comp. A: 0 h B: 0 h

Modes of Operation

- **Fast PWM mode**
  - □ useful for setting the speed of a motor or the light intensity of a light source



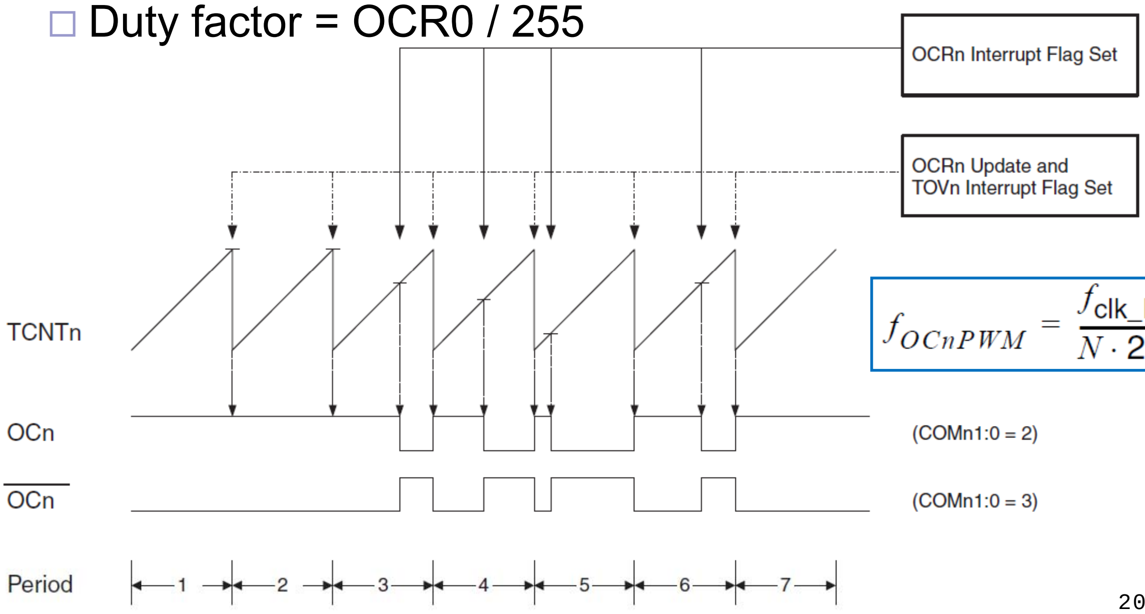
$$D = \frac{T_{on}}{T_{on} + T_{off}}$$

  - □ The mean value can be within maximum and minimum values, depending on *D*

$$V_{MED} = DV_{CC} + (1 - D)V_{GND}$$

Fast PWM mode

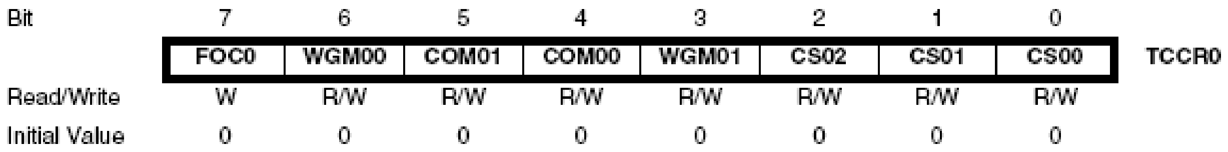
- □ Frequency is fixed by the clock select bits CSn2:0
- □ Duty factor is set by writing OCR0 register
- □ Duty factor = OCR0 / 255



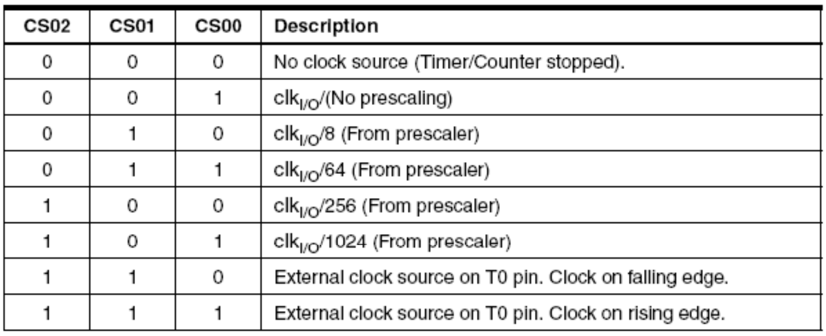
$$\boxed{f_{OCnPWM} = \frac{f_{\text{clk\_I/O}}}{N \cdot 256}}$$

Fast PWM mode

- **Example**: How to calculate the PWM frequency
  - Use the timer/counter0 in PWM mode



Bit	7	6	5	4	3	2	1	0	
	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	TCCR0
Read/Write	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	



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.

How to calculate the PWM frequency

- PWM → the frequency is constant, the duty cycle varies
- Assume $f_{crystal}$ = 13.5MHz, we divide it by:
  - □ prescaler: max 1024
  - □ maximum value for the 8 bit timer register: 256
  - □ we have f_{PWM} = 13500000/1024/256 = 51 Hz
  - □ Note: 51Hz is enough for light bulbs or motors, but a 51Hz flicker is visible on LEDs
  - □ we choose a lower prescaler: 256
  - □ f_{PWM} = 13500000/256/256 = 205 Hz
  - □ Prescaler=256 → CS02:00 = 100 (see previous table)

Timer/Counter 0 Control Register

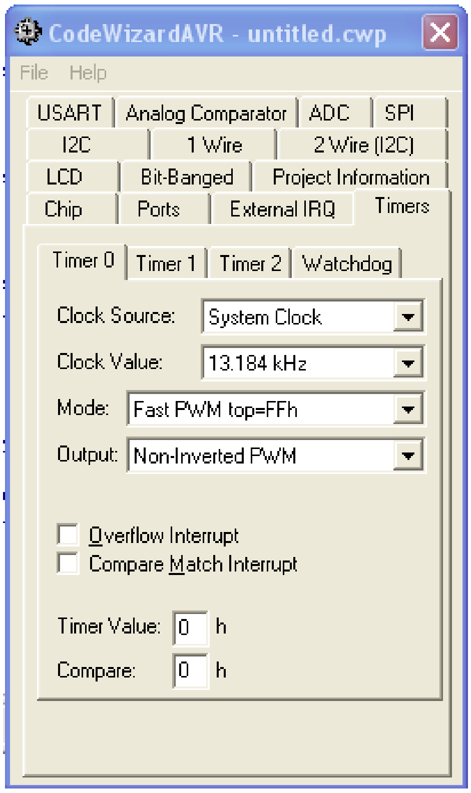
Bit	7	6	5	4	3	2	1	0	
	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	TCCR0
Read/Write	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Mode	WGM01 (CTC0)	WGM00 (PWM0)	Timer/Counter Mode of Operation	TOP	Update of OCR0	TOV0 Flag Set-on
0	0	0	Normal	0xFF	Immediate	MAX
1	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	1	0	CTC	OCR0	Immediate	MAX
3	1	1	Fast PWM	0xFF	BOTTOM	MAX

COM01	COM00	Description
0	0	Normal port operation, OC0 disconnected.
0	1	Reserved
1	0	Clear OC0 on compare match, set OC0 at BOTTOM, (non-inverting mode)
1	1	Set OC0 on compare match, clear OC0 at BOTTOM, (inverting mode)

- table COM 01:00 is for the Fast PWM mode
- we choose WGM 01:00 = 11, COM 01:00 = 10 CS 02:00 = 100
- the final value is: TCCR0 = 01101100 = 6Ch

Code Wizard



Clock Source: System Clock

Clock Value: 13.184 kHz

Mode: Fast PWM top=FFh

Output: Non-Inverted PWM

☐ Overflow Interrupt

☐ Compare Match Interrupt

Timer Value: 0 h

Compare: 0 h

Sample program in PWM mode

```
// timer0 init in PWM
// Clock source: System Clock/256, Clock value: 52734 Hz, Mode: Fast PWM top=FFh, OC0:
    Non-Inverted PWM
TCCR0=0x6C;
TCNT0=0x00;
OCR0=0x00;

// 4 different light intensities for LED, set using 4 different values of the OCR0 register
// pause 1 second between each intensity change
void main (void)
{
    while(TRUE)
    {
        OCR0 = 0; delay_ms(1000);           // no light
        OCR0 = 4; delay_ms(1000);           // little light
        OCR0 = 16; delay_ms(1000);          // medium light
        OCR0 = 253; delay_ms(1000);  // full light
    }
}
```