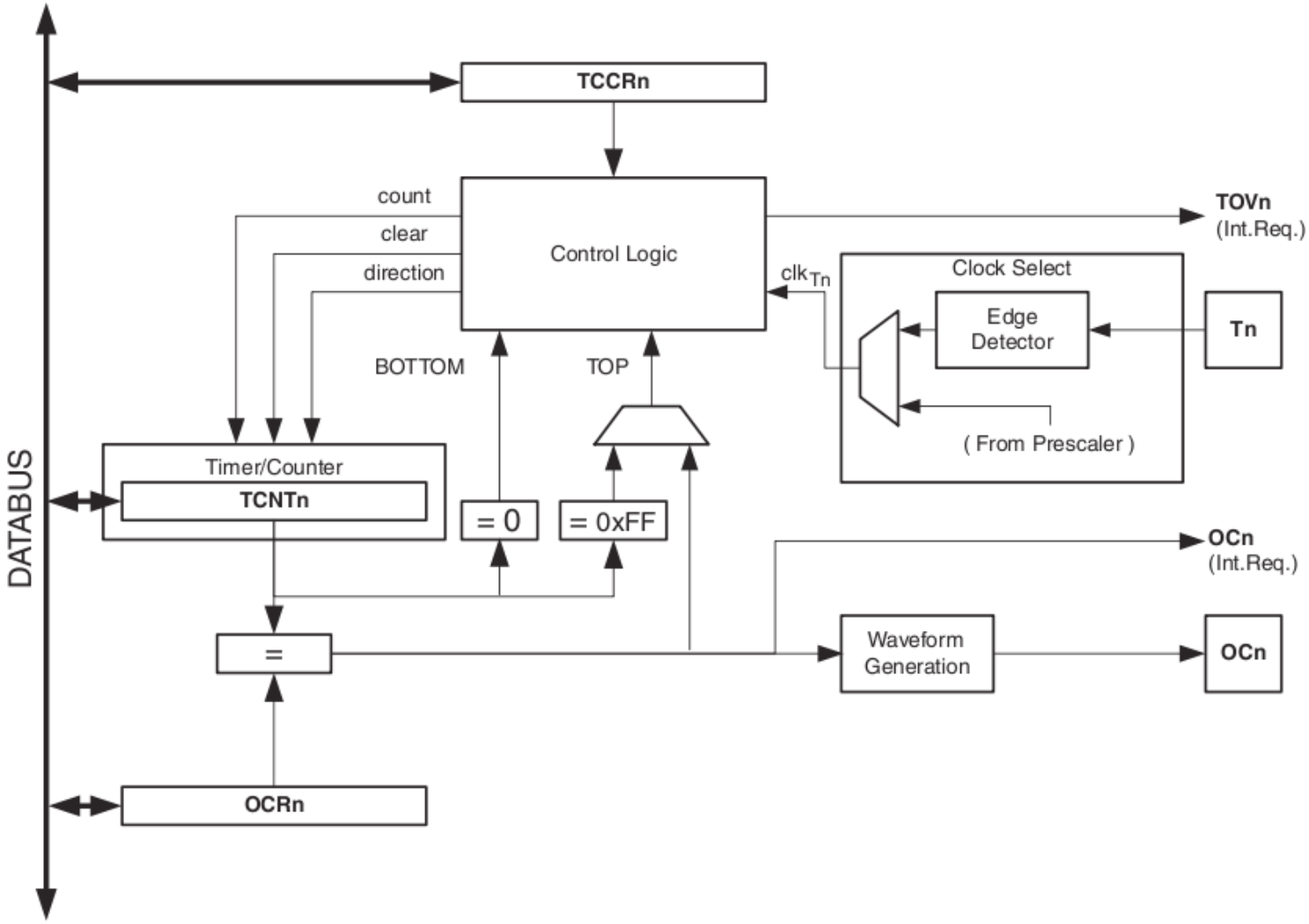
Timer / Zähler

Timer/Zähler

Hochschule Offenburg
University of Applied Sciences

- Spezial Register welches hoch bzw. runter gezählt werden kann.
- Zähler: Registerwert wird durch externes Signal in- oder dekrementiert.
- Timer: Register wird von einem internen Taktsignal in- oder dekrementiert.
- Abhängig vom Zählerstand kann
  - ein Ausgang gesetzt oder gelöscht werden (PWM).
  - ein Interrupt ausgelöst werden.

8-bit Timer/Counter 0



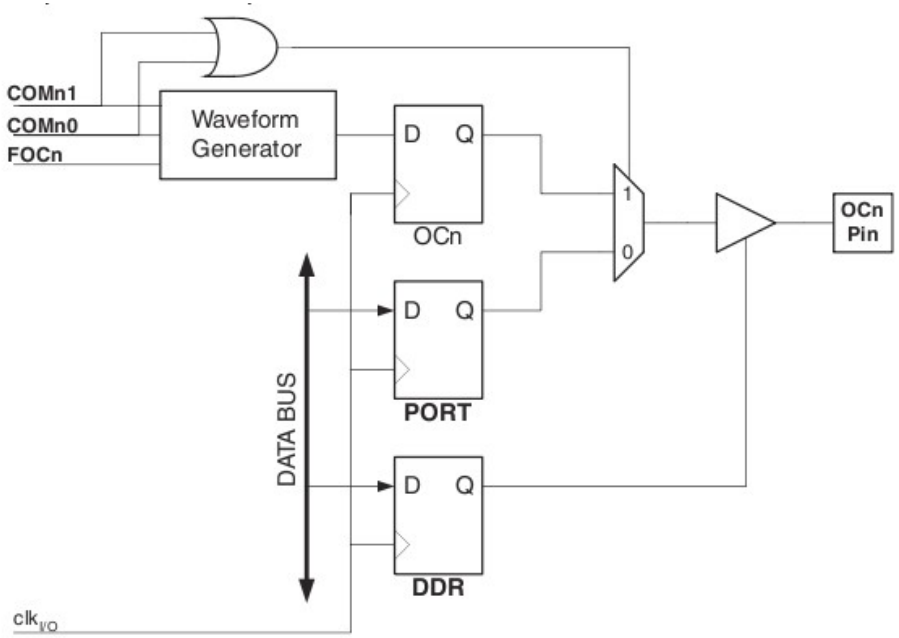
Clock Source

Hochschule Offenburg
University of Applied Sciences

- Externes Signal über T0 (Signal von PINB0).
- Internes Taktsignal mit Vorteiler.
- Einstellung über CS02...CS00 in TCCR0.

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.

Output Unit



- Bestimmt den Zustand am Pin PB3
- Einstellung über COM01 und COM00 in TCCR0
- Abhängig vom Mode
- FOC0 in TCCR0 erzwingt das Setzen eines bestimmten Zustandes

Interrupts

Hochschule Offenburg
University of Applied Sciences

- Interrupt bei Überlauf von TCNT0. TOV (Timer/Conter0 Overflow Flag) in TIFR.
- Interrupt bei Gleichheit von TCNT0 und OCR0. OCF0 (Output Compare Flag 0) in TIFR.
- Enable Bits:
  - TOIE0 (Timer/Counter0 Overflow Interrupt Enable)
  - OCIE0 (Timer/Counter0 Output Compare Match Interrupt Enable)

Normal Mode

Hochschule Offenburg
University of Applied Sciences

- Der Modus wird eingestellt über die Bits WGM00 und WGM01 in TCCR0
- Bei WGM01:0 = 0 ist der Normal Mode aktiv.
- Einfaches Hochzählen von TCNT0.
- TOV0 kann als 9-Bit verwendet werden.

Beispiel 1

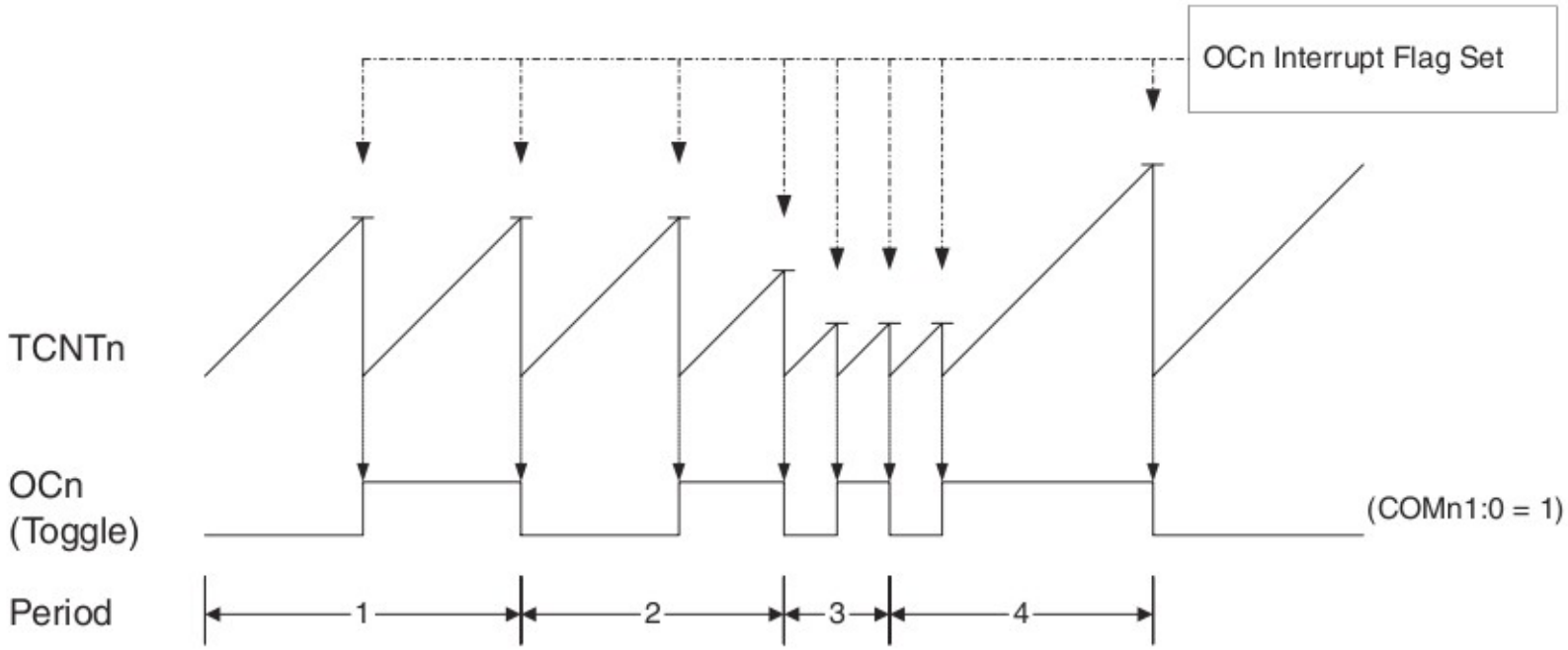
Hochschule Offenburg
University of Applied Sciences

```
#include <stdlib.h>
#include <avr/io.h>

int main(void)
{
    ...
    DDRB  &= ~_BV(PB0);              //PB0 Eingang
    PORTB |=  _BV(PB0);              //Pull-Up aktiv

    //Clock Source auf extern.(Fallende Flanke)
    TCCR0 = _BV(CS02) | _BV(CS01);
    for(;;)
    {
        ...
        if( TCNT0 == 25)
        {
            do something
            TCNT0 = 0;
        }
    }
}
```

Clear Timer on Compare Match (CTC)Mode

Hochschule Offenburg
University of Applied Sciences

- WGM01:0 = 2
- TCNT0 wird bis OCR0 inkrementiert.
- TOV tritt im „Normalfall“ nicht auf.
- T= N*(1+OCR0)/ fclk



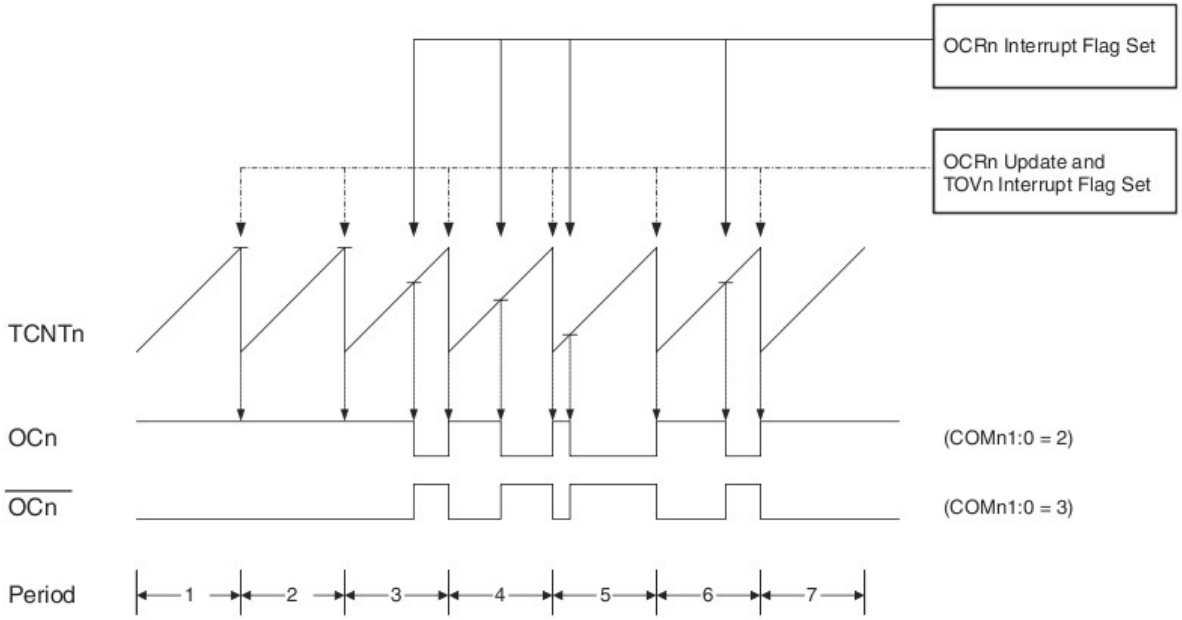
Beispiel 2

Hochschule Offenburg
University of Applied Sciences

```
ISR(TIMER0_COMP_vect)  // wird jede ms ausgeführt
{
    static uint16_t i=0;
    i++;
    if( i== 500)
    {
        i= 0;
        PORTA ^=_BV(PA4);
    }
}
int main(void)
{

    // CTG Mode, Prescaler=64 -> clk=250kHz
    TCCR0 = _BV(WGM01) | _BV(CS01) | _BV(CS00);

    //On Compare match Interrupt enable for timer 0
    TIMSK |= _BV(OCIE0);

    OCR0= 249; // 250kHz/(249+1) = 1kHz
    for(;;);
}
```

Fast PWM Mode

- WGM01:0 = 3
- TCNT0 wird bis 0xFF inkrementiert
- $f_{PWM} = f_{clk} / (N*256)$
  N = 1, 8, 64, 256, 1024
- Pulsbreite wird über OCR0 eingestellt.
- OCR0 wird gebuffert.



Beispiel 3

Hochschule Offenburg
University of Applied Sciences

```
int main(void)
{

    //Fast PWM Mode, clear OC0 on compare match,Prescaler=256 -> clk=62,5kHz
    TCCR0 = _BV(WGM00) | _BV(COM01)  | _BV(CS02)  |_BV(CS00);

    OCR=63 //Pulsbreite 25%

    DDRB |= _BV(PB3);   // PB3 Ausgang

    for(;;);

}
```

Phase Correct PWM Mode

- WGM01:0 = 1
- TCNT0 wird bis 0xFF inkrementiert
- $f_{PWM} = f_{clk} / (N*510)$
- Pulsbreite wird über OCR0 eingestellt.
- OCR0 wird gebuffert.