Microcontrollers 1

Timers

Christiaen Slot – c.g.m.slot@saxion.nl
Hans Stokkink – j.s.d.stokkink@saxion.nl

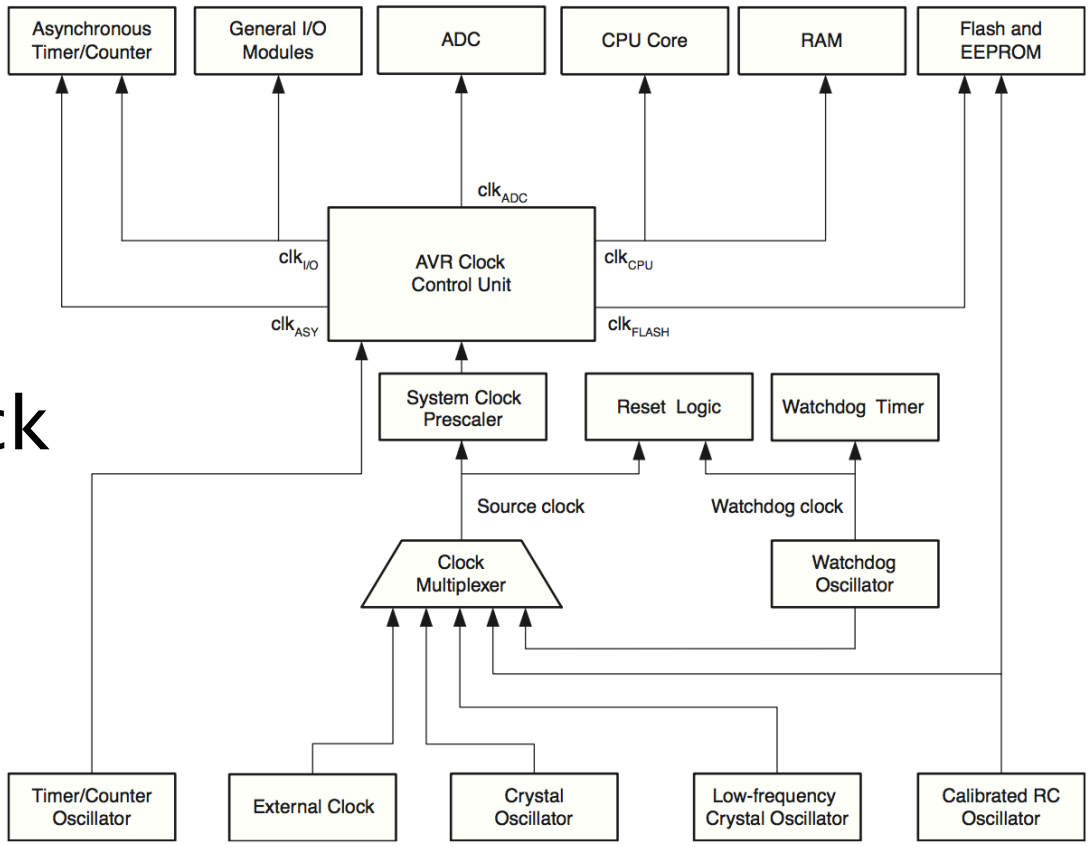
SAXION

Kom verder. Saxion.

Content

- the Atmega Clock(s)
- What does a timer do?
- Simple manual count example
- Prescalers & registers
- Timer with interrupt example

Figure 8-1. Clock Distribution



the Clock

Clock Oscillators

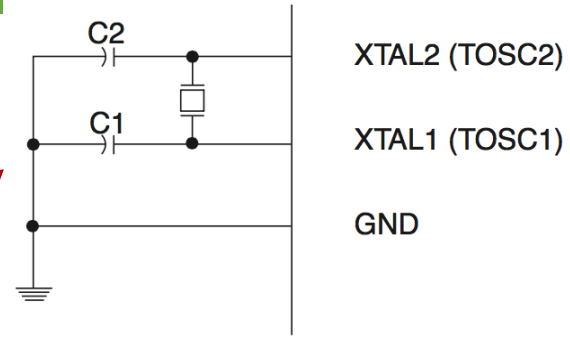


Table 8-1. Device Clocking Options Select[1]

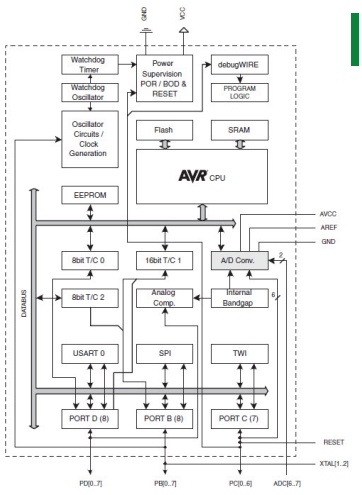
Device Clocking Option	CKSEL3..0
Low Power Crystal Oscillator	1111 - 1000
Full Swing Crystal Oscillator	0111 - 0110
Low Frequency Crystal Oscillator	0101 - 0100
Internal 128 kHz RC Oscillator	0011
Calibrated Internal RC Oscillator = default = 8 mHz	0010
External Clock	0000
Reserved	0001

Note: 1. For all fuses "1" means unprogrammed while "0" means programmed.

What does a timer do?

1. Start to count
2. Keep counting
3. If we're there: do something
4. Start again

Timers in ATmega328



- Three timers
  - timer0 & timer2: 8 bit
  - timer1: 16 bit
- Overflow and Compare Match
- Prescalers to scale the duration
- PWM generation
  - Fast PWM (single slope)
  - Phase/frequency correct PWM
- Input Capture (timestamps)
- … (see datasheet)

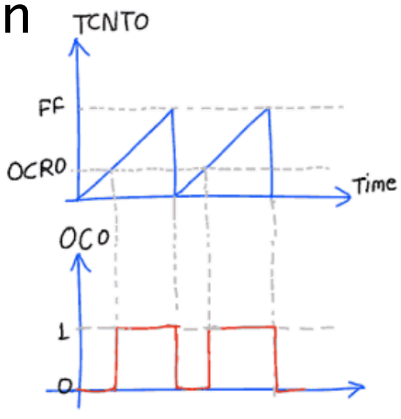
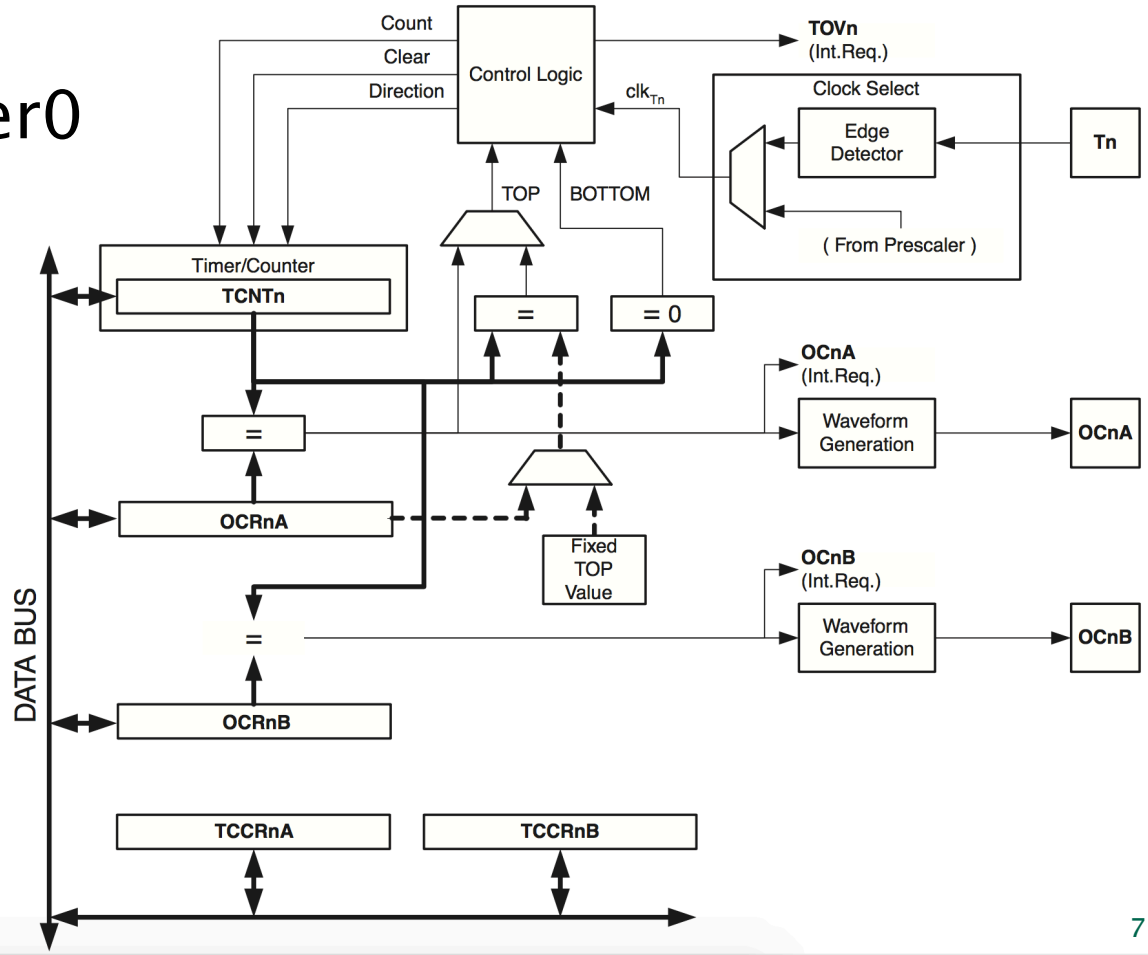


Figure 14-1. 8-bit Timer/Counter Block Diagram

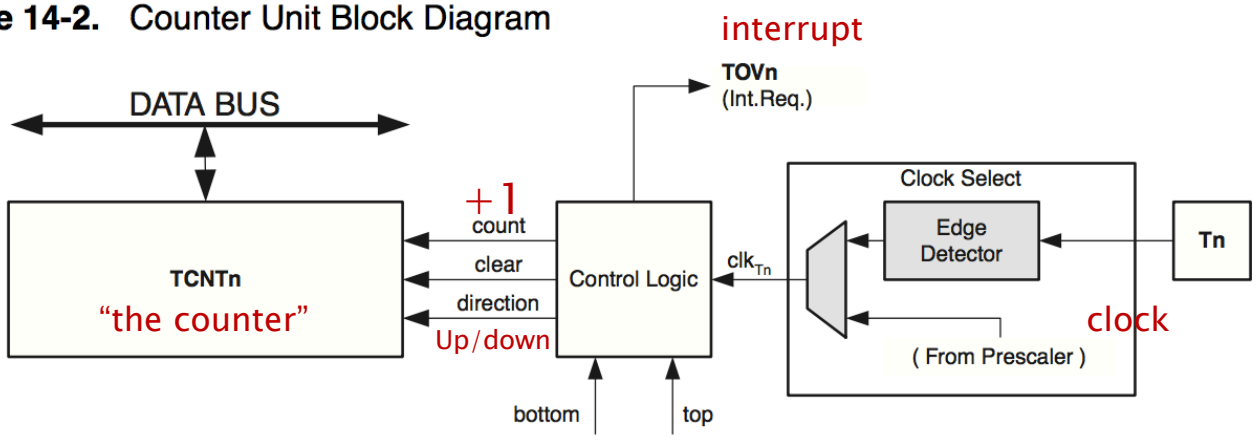
Timer0



Kom verder. Saxion.

Counter block diagram

gure 14-2. Counter Unit Block Diagram



Signal min. & max.
value of cnt reached

Content

- the Atmega Clock(s)
- What does a timer do?
- **Simple manual count example**
- Prescalers & registers
- Timer with interrupt example

What do we need?

1. Start to count
2. Keep counting
3. If we're there: do something
4. Start again

Pseudo code

```
Set up LED hardware
Set up timer

WHILE forever
  IF timer value IS EQUAL TO OR MORE THAN 1/20 sec THEN
    Reset counter
    Toggle LED
  END IF
END WHILE
```

How far to count with 1mHZ clock?

$$
\begin{aligned}
\text{Target Timer Count} &= \frac{1}{\text{Target Frequency}} \Big/ \frac{1}{\text{Timer Clock Frequency}} - 1 \\
&= \frac{1}{20} \Big/ \frac{1}{1000000} - 1 \\
&= \frac{.05}{0.000001} - 1 \\
&= 50000 - 1 \\
&= 49999
\end{aligned}
$$

code example (1)

```
# include <avr /io.h>
int main ( void )
{
    DDRB |= (1 << 0); // Set LED as output
    TCCR1B |= (1 << CS10 ); // Set up timer
    while(1)
    {
        if ( TCNT1 >= 49999) // it timer end?
        {
            PORTB ^= (1 << 0); // Toggle the LED
            TCNT1 = 0; // Reset timer value
        }
    }
}
```

code example (1)

```
# include <avr /io.h>
int main ( void )
{
    DDRB |= (1 << 0); // Set LED as output
    TCCR1B |= (1 << CS10 ); // Set up timer
    while(1)
    {
        if ( TCNT1 >= 49999) // it timer end?
        {
            PORTB ^= (1 << 0); // Toggle the LED
            TCNT1 = 0; // Reset timer value
        }
    }
}
```


Content

- the Atmega Clock(s)
- What does a timer do?
- Simple manual count example
- **Prescalers & registers**
- Timer with interrupt example

Pre-scaler (cs bits)

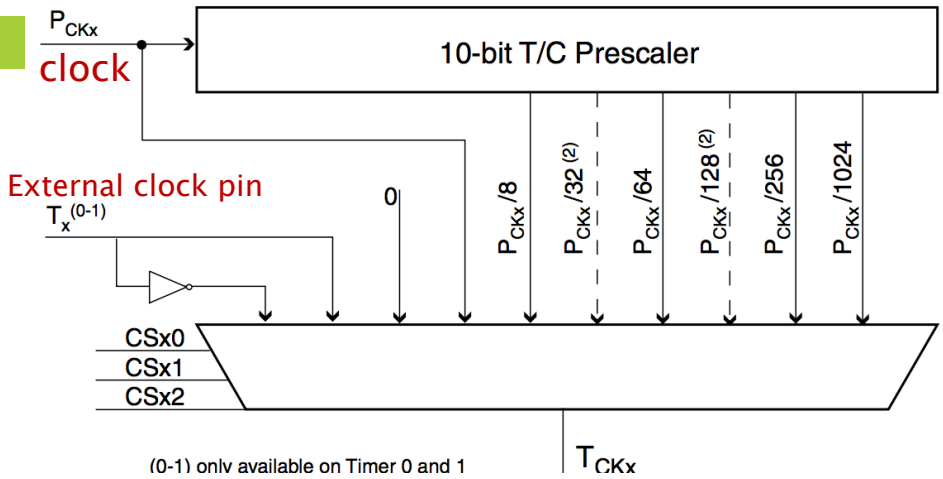


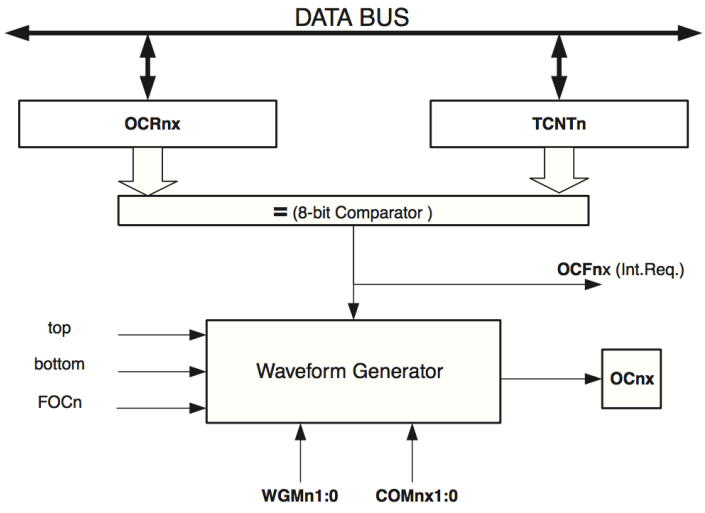
Table 15-5. Clock Select Bit Description

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.

And now check automatically …

- Counter mode: Timer overflow
- Clear Timer on Compare Match mode *(CTC)*

If a timer reaches a value or overflows
→ generate interrupt



Timer 1 Control Register A/B

TCCR1A – Timer/Counter1 Control Register A

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	

TCCR1B – Timer/Counter1 Control Register B

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	

Compare Output Mode channel A/B

Waveform Generation Mode (PWM)

Clock Select bits (start/stop & prescalers)

code example (2)

```
#include <avr/interrupt.h>
void main()
{
    DDRB = (1 << PB5);
    cli();              // disable global interrupts
    TCCR1A = 0;         // set entire TCCR1A register to 0
    TCCR1B = 0;
    TIMSK1 |= (1 << TOIE1 ); // timer overflow interrupt enable
    TCCR1B |= (1 << CS11) | (1 << CS10); // prescaler
    sei();
    while(1) {// doe iets ander }
}
// interrupt service routine
ISR(TIMER1_OVF_vect)
{
    PORTB ^= (1 << PB5);
}
```

Timer0

Figure 14-1. 8-bit Timer/Counter Block Diagram