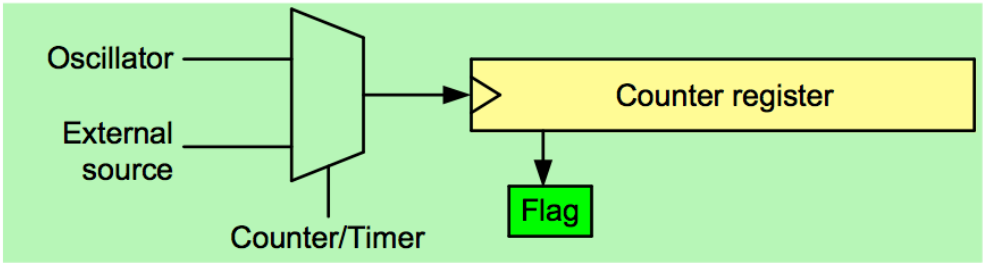
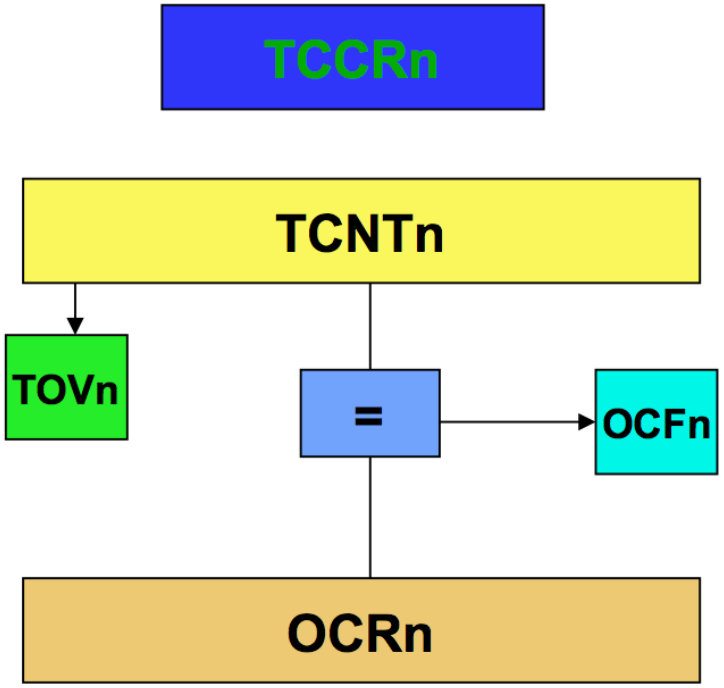
Timer in AVR

- TCNTn (Timer/Counter register)
- TOVn (Timer Overflow flag)
- TCCRn (Timer Counter control register)
- OCRn (output compare register)
- OCFn (output compare match flag)

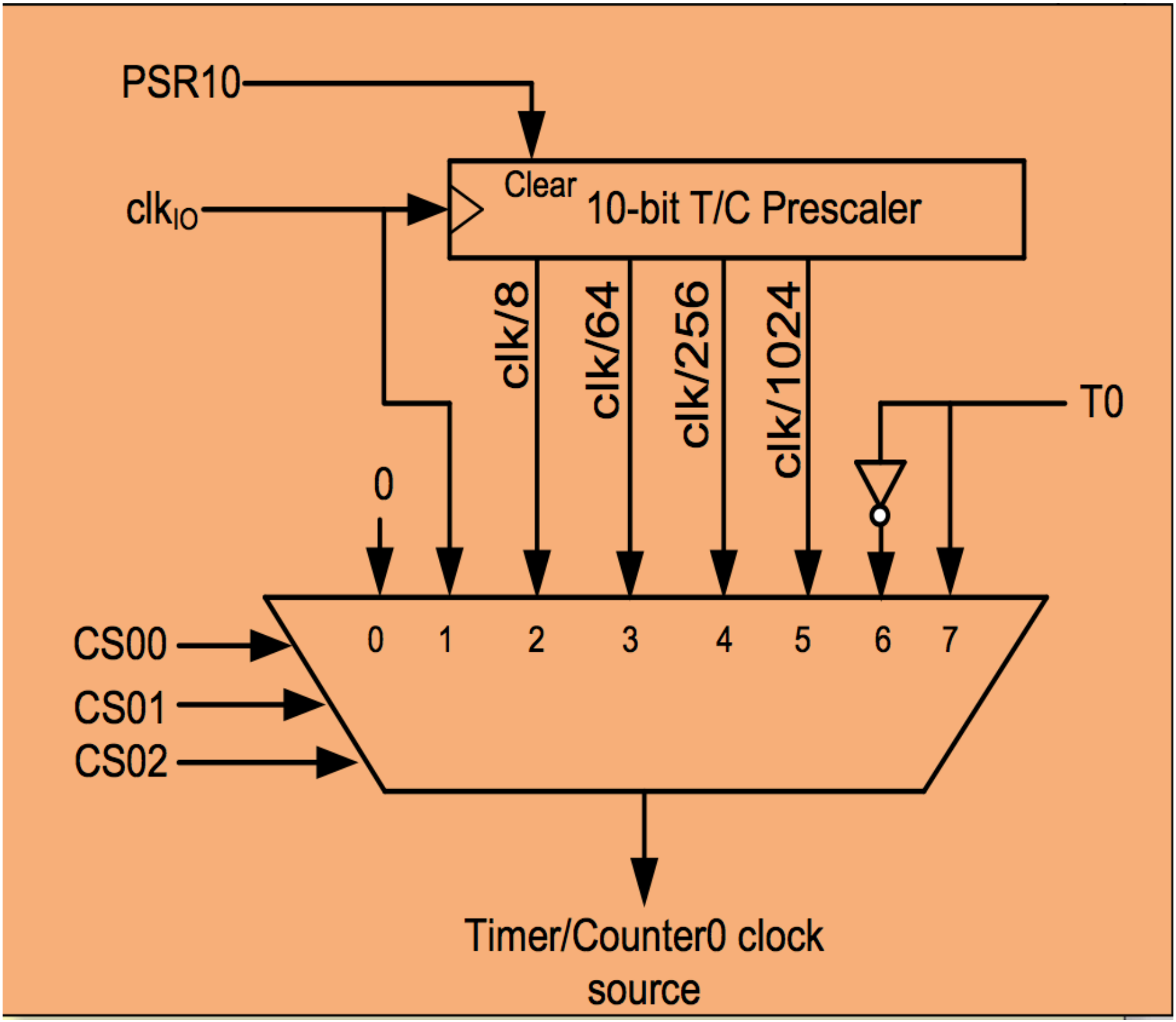


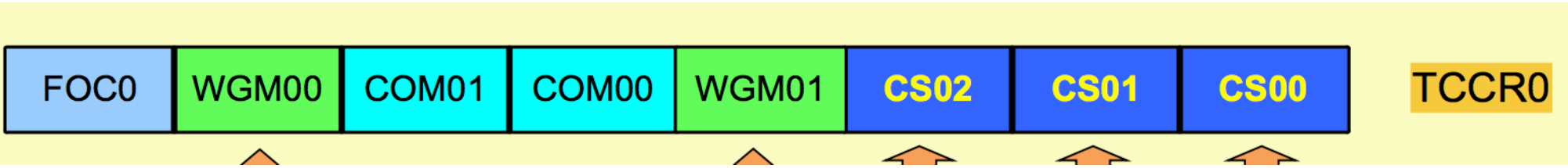


Comment:

All of the timer registers are byte-addressable I/O registers

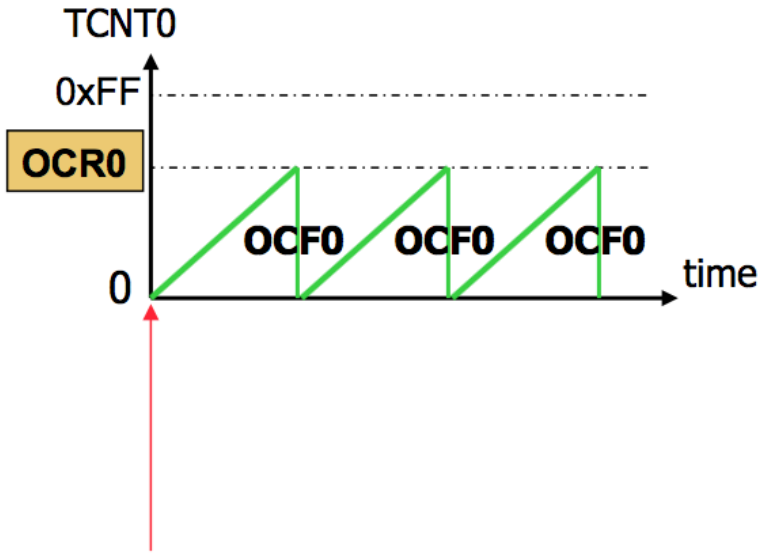

OCF2
TOV2
ICF1
OCF1A
OCF1B
TOV1
OCF0
TOV0
TIFR

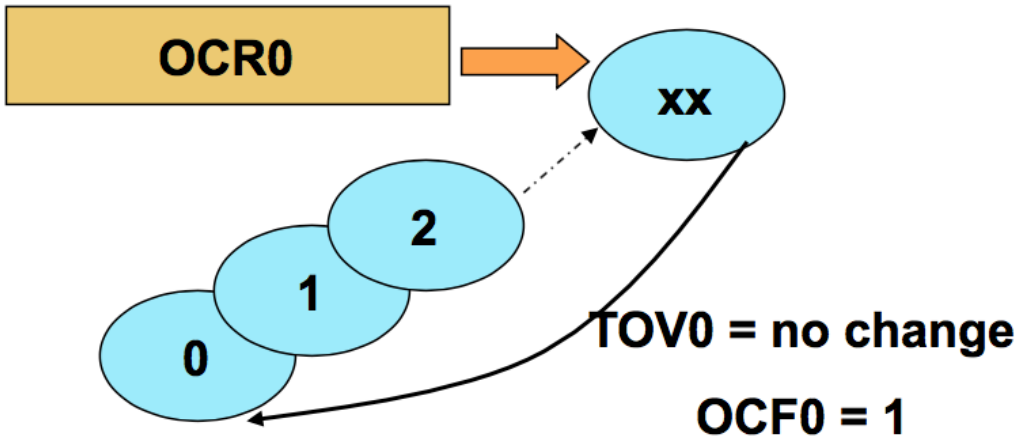

PSR10
clkIO
Clear
10-bit T/C Prescaler
clk/8
clk/64
clk/256
clk/1024
T0
0
0 1 2 3 4 5 6 7
CS00
CS01
CS02
Timer/Counter0 clock source


FOC0
WGM00
COM01
COM00
WGM01
CS02
CS01
CS00
TCCR0

WGM00	WGM01	Comment
0	0	Normal
0	1	CTC (Clear T
1	0	PWM, phase
1	1	Fast PWM

CTC (Clear Timer on Compare match) mode





TOV0: 0

OCF0: 1

Example 1: Write a program that waits 14 machine cycles in Normal mode.

$100
-$0E
$F2

FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	TCCR0
OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	TIFR

0 0 0 0 0 0 0 1

```
.INCLUDE "M32DEF.INC"

        LDI     R16,0x20
        SBI     DDRB,5    ;PB5 as an output
        LDI     R17,0
        OUT     PORTB,R17
BEGIN:  LDI     R20,0xF2
        OUT     TCNT0,R20         ;load timer0
        LDI     R20,0x01
        OUT     TCCR0,R20 ;Ti
AGAIN:  IN      R20,TIFR
        SBRS    R20,0 ;if TOV
        RJMP    AGAIN
        LDI     R20,0x0
        OUT     TCCR0,R20
        LDI     R20,(1<<TOV0)
        OUT     TIFR,R20

        EOR     R17,R16
        OUT     PORTB,R17
        RJMP    BEGIN
```

set byte immediate

toggle

SKIP

```
DDRB = 1<<5;
PORTB &= ~(1<<5);   //PB5=0
while (1)
{
```

Question: How to calculate the delay generated by the timer?

Answer:

1) Calculate how much a machine clock lasts.
   T= 1/f
2) Calculate how many machine clocks it waits.
3) Delay = T * number of machine cycles

In example 1 calculate the delay. XTAL = 10 MHz.

Solution 1 (inaccurate):

1) Calculating T:

T = 1/f = 1/10M = 0.1µs

2) Calculating num of machine cycles:

```
$100
-$F2
 $0E = 14
```

3) Calculating delay

14 * 0.1µs = 1.4 0µs

```
.INCLUDE "M32DEF.INC"

        LDI     R16,0x20
        SBI     DDRB,5   ;PB5 as an output
        LDI     R17,0
        OUT     PORTB,R17
BEGIN:  LDI     R20,0xF2
        OUT     TCNT0,R20          ;load timer0
        LDI     R20,0x01
        OUT     TCCR0,R20 ;Timer0,Normal mode,int clk
AGAIN:  IN      R20,TIFR           ;read TIFR
        SBRS    R20,0 ;if TOV0 is set skip next inst.
        RJMP    AGAIN
        LDI     R20,0x0
        OUT     TCCR0,R20          ;stop Timer0
        LDI     R20,0x01
        OUT     TIFR,R20           ;clear TOV0 flag

        EOR     R17,R16            ;toggle D5 of R17
        OUT     PORTB,R17          ;toggle PB5
        RJMP    BEGIN
```

```
start_timer:
    LDI   R20, 0x00        x
    OUT   TCNT0, R20       x
    LDI   R20, 0x02        x
    OUTP  TCCR0, R20       ; normal mode, clk/8   x
    RET

stop_timer:                                        -4
    LDI   R20, 0x00        y
    OUT   TCCR0, 0x00      ; clk stopped          y
    RET
```

The difference between Timer0 and Timer2

- **Timer0**

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)

- **Timer2**

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

Example 2: Assuming that XTAL = 10 MHz, write a program to generate a square wave with a period of 10 ms on pin PORTB.3.

- For a square wave with T = 10 µs we must have a time delay of 5 µs. Because XTAL = 10 MHz, the counter counts up every 0.1 µs. This means that we need 5 µs / 0.1 µs = 50 clocks. 256 - 50 = 206.

```
.INCLUDE "M32DEF.INC"

         LDI     R16,0x08
         SBI     DDRB,3   ;PB3 as an output
         LDI     R17,0
         OUT     PORTB,R17
BEGIN:   LDI     R20,206
         OUT     TCNT0,R20         ;load timer0
         LDI     R20,0x01
         OUT     TCCR0,R20 ;Timer0,Normal mode,int clk
AGAIN:   IN      R20,TIFR          ;read TIFR
         SBRS    R20,TOV0 ;if TOV0 is set skip next
         RJMP    AGAIN
         LDI     R20,0x0
         OUT     TCCR0,R20         ;stop Timer0
         LDI     R20,0x01
         OUT     TIFR,R20          ;clear TOV0 flag
         EOR     R17,R16           ;toggle D3 of R17
         OUT     PORTB,R17         ;toggle PB3
         RJMP    BEGIN
```

```
DDRB = 1<<3;
PORTB &= ~ (1<<3);
while (1)
{
  TCNT0 = 206;
  TCCR0 = 0x01;
  while((TIFR&0x01) == 0);
  TCCR0 = 0;
  TIFR = 1<<TOV0;
  PORTB = PORTB ^ (1<<3);
}
```

Rewrite example 2 using CTC

FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	TCCR0
OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	TIFR

- **For a square wave with T = 10 µs we must have a time delay of 5 µs. Because XTAL = 10 MHz, the counter counts up every 0.1 µs. This means that we need 5 µs / 0.1 µs = 50 clocks. Therefore, we have OCR0= 49.**

```
.INCLUDE "M32DEF.INC"
        LDI     R16,0x08
        SBI     DDRB,3   ;PB3 as an output
        LDI     R17,0
        OUT     PORTB,R17
        LDI     R20,49
        OUT     OCR0,R20 ;load timer0
BEGIN:  LDI     R20,0x09
        OUT     TCCR0,R20 ;Timer0,CTC mode,int clk
AGAIN:  IN      R20,TIFR          ;read TIFR
        SBRS    R20,OCF0 ;if OCF0 is set skip next
        RJMP    AGAIN
        LDI     R20,0x0
        OUT     TCCR0,R20         ;stop Timer0
        LDI     R20,0x02
        OUT     TIFR,R20          ;clear TOV0 flag
        EOR     R17,R16           ;toggle D3 of R17
        OUT     PORTB,R17         ;toggle PB3
        RJMP    BEGIN
```

```
DDRB |= 1<<3;
PORTB &= ~(1<<3);
while (1)
{
  OCR0 = 49;
  TCCR0 = 0x09;

while((TIFR&(1<<OCF0))==0);
  TCCR0 = 0; //stop timer0
  TIFR = 0x02;
  PORTB.3 = ~PORTB.3;
}
```