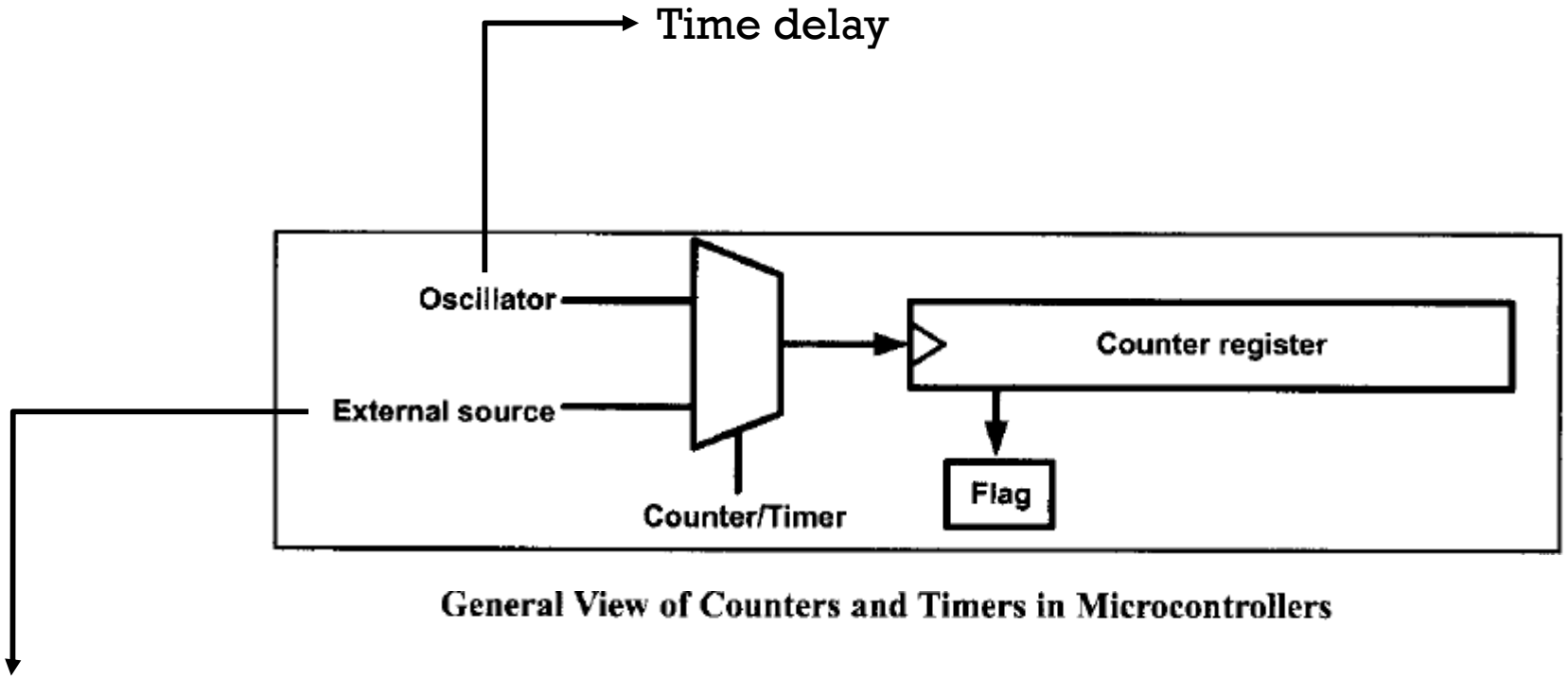
EMBEDDED SYSTEMS CMPE-453

Department of Computer Engineering

Timers/Counters



General View of Counters and Timers in Microcontrollers

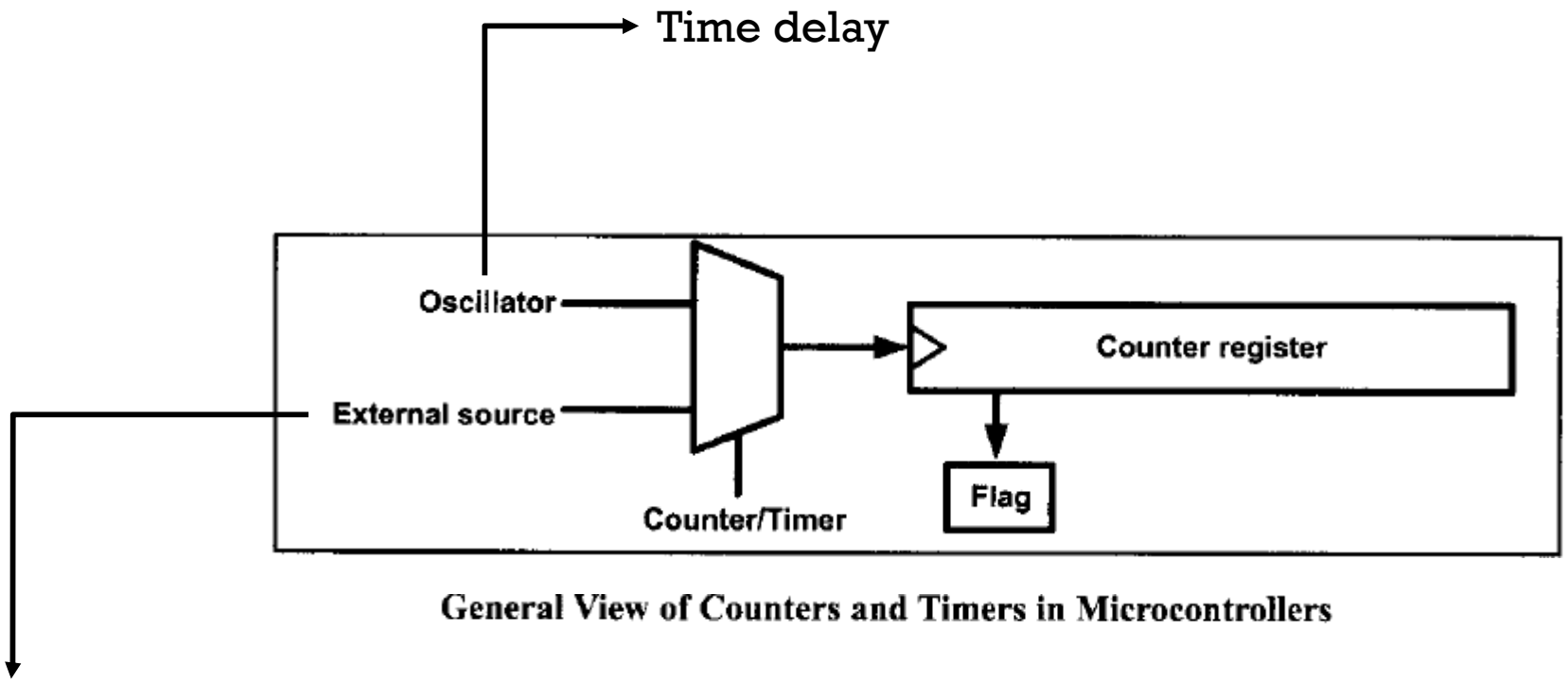
To generate a time delay

- Each osilator tick increases counter register
- How many ticks has occured in counter register
- Oscilator frequency→tick period
- A.O.Time = tick period x register content

Clear counter and wait it reaches upto a certain value

Example

- Oscillator freq. = 1MHz
- 1 tick for each $1/10^6$ secs.
- 1sec delay, we need 10^6 ticks



General View of Counters and Timers in Microcontrollers

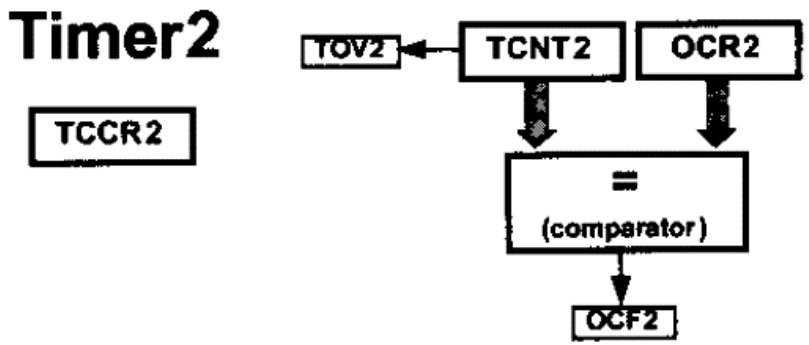
Second way for a time delay

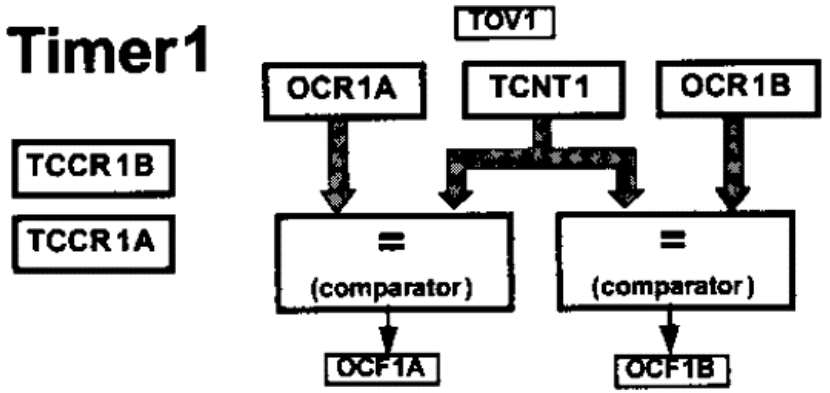
- Counter flag is cleared when it is overflow.
- Load counter register, wait upto it is cleared.

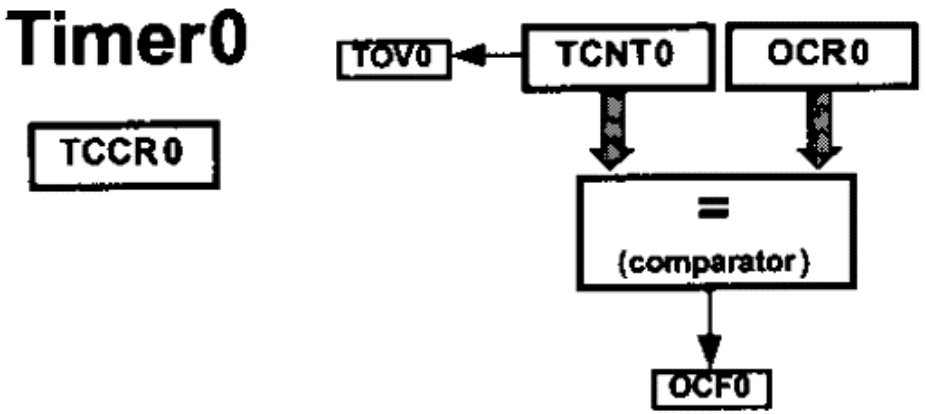
Example

- Oscillator freq. = 1MHz
- 4µsec delay, we need 4 ticks
- Load counter register with FC

FC – FD – FE –FF – overflow (00)

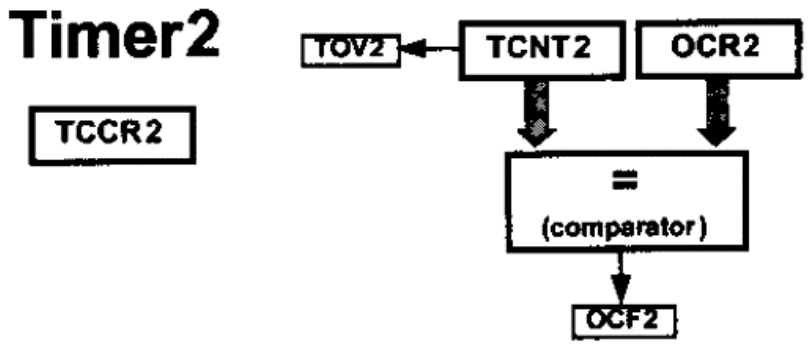


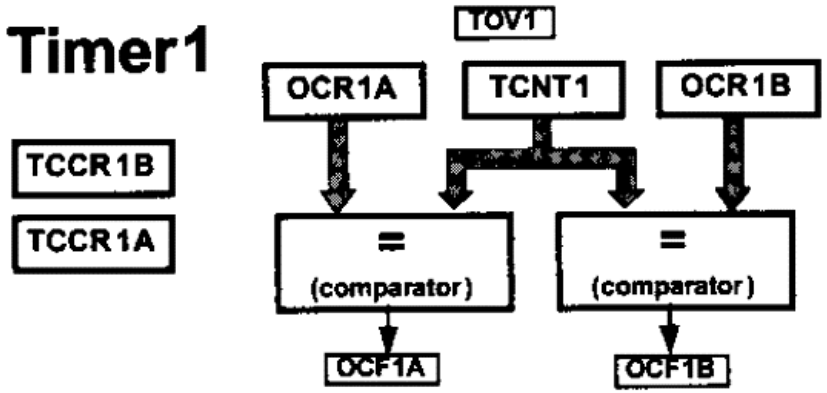


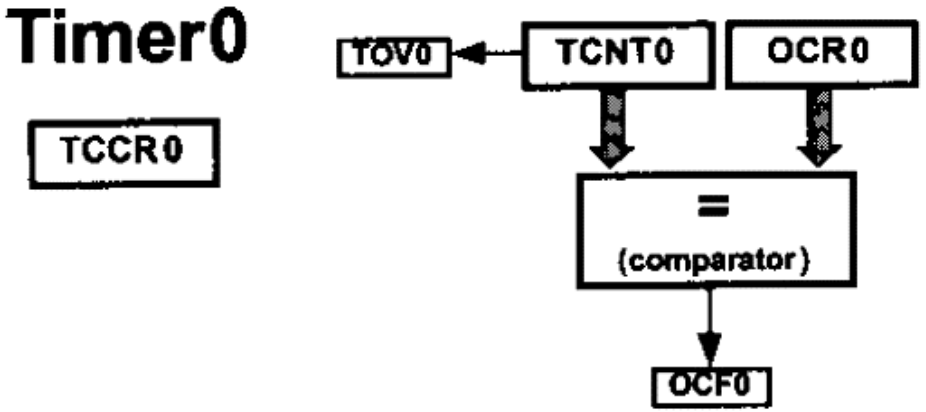


- TCNTn : 0 when reset, counts up for each pulse
- Load or read TCNTn

- TOVn : Timer overflow
- It is set when overflow







- TCCRn : 0 when reset, counts up for each pulse
- Specify to work as timer or counter

- OCRn : Timer overflow
- It is set when overflow

- OCFn : Output compare flag
- It is set when TCNTn and OCRn is equal

PROGRAMMING TIMER0

TCNT0

D7	D6	D5	D4	D3	D2	D1	D0

TCCR0 (Timer/Counter Control Register)

FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00

CS02:00

D2	D1	D0	Timer0 clock selector
0	0	0	No clock source (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 source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

WGM00, WGM01

D6	D3	Timer0 mode selector bits
0	0	Normal
0	1	CTC (Clear Timer on Compare Match)
1	0	PWM, phase correct
1	1	Fast PWM

EXAMPLES

Example 9-1

Find the value for TCCR0 if we want to program Timer0 in Normal mode, no prescaler. Use AVR's crystal oscillator for the clock source.

Solution:

TCCR0 =

0	0	0	0	0	0	0	1
FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00

Example 9-2

Find the timer's clock frequency and its period for various AVR-based systems, with the following crystal frequencies. Assume that no prescaler is used.

(a) 10 MHz (b) 8 MHz (c) 1 MHz

Solution:

(a) F = 10 MHz and T = 1/10 MHz = 0.1 µs

(b) F = 8 MHz and T = 1/8 MHz = 0.125 µs

(c) F = 1 MHz and T = 1/1 MHz = 1 µs

PROGRAMMING TIMER0

TIFR (Timer/Counter Interrupt Flag Register)

OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0

TOV0 D0 Timer0 overflow flag bit
0 = Timer0 did not overflow.
1 = Timer0 has overflowed (going from $FF to $00).

WGM00, WGM01

D6	D3	Timer0 mode selector bits
0	0	Normal
0	1	CTC (Clear Timer on Compare Match)
1	0	PWM, phase correct
1	1	Fast PWM

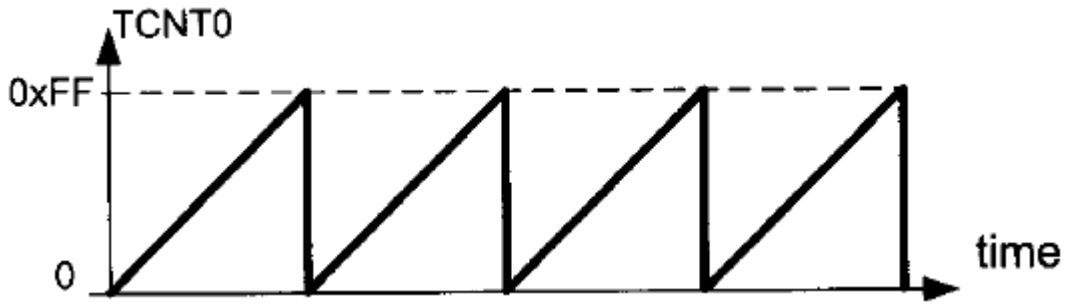
NORMAL MODE

TCCR0 (Timer/Counter Control Register)

FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00

WGM00, WGM01

D6	D3	Timer0 mode selector bits
0	0	Normal
0	1	CTC (Clear Timer on Compare Match)
1	0	PWM, phase correct
1	1	Fast PWM



TCNT0

D7	D6	D5	D4	D3	D2	D1	D0

PROGRAMMING TIMER0 IN NORMAL MODE

1. Load the TCNT0 register with the initial count value.

2. Load the value into the TCCR0 register, indicating which mode (8-bit or 16-bit) is to be used and the prescaler option. When you select the clock source, the timer/counter starts to count, and each tick causes the content of the timer/counter to increment by 1.

3. Keep monitoring the timer overflow flag (TOV0) to see if it is raised. Get out of the loop when TOV0 becomes high.

4. Stop the timer by disconnecting the clock source, using the following instructions:

5. Clear the TOV0 flag for the next round.

6. Go back to Step 1 to load TCNT0 again.

EXAMPLE-1

Write C program to toggle all the bits of PORTB continuously with some delay.
Use Timer0, Normal mode, and no prescaler options to generate the delay.

```
#include "avr/io.h"
void T0Delay ( );
int main ( )
{
      DDRB = 0xFF;       //PORTB output port

      while (1)
      {
            PORTB = 0x55;      //repeat forever
            T0Delay ( );       //delay size unknown
            PORTB = 0xAA;      //repeat forever
            T0Delay ( );
      }
}

void T0Delay ( )
{
      TCNT0 = 0x20;              //load TCNT0
      TCCR0 = 0x01;              //Timer0, Normal mode, no prescaler
      while ((TIFR&0x1)==0);     //wait for TOV0 to roll over
      TCCR0 = 0;
      TIFR = 0x1;                //clear TOV0
}
```

Step-1
Step-2
Step-3
Step-4
Step-4
Step-5
Be careful : to clear this flag, we write 1 !!

EXAMPLE-2

Write C program to toggle only bit 4 of PORTB continuously every 70µsec.
Use Timer0, Normal mode, and 1:8 prescaler. Assume XTAL = 8MHz.

```
#include "avr/io.h"

void T0Delay ( );

int main ( )
{
      DDRB = 0xFF;      //PORTB output port

      while (1)
      {
            T0Delay ( );            //Timer0, Normal mode
            PORTB = PORTB ^ 0x10;   //toggle PORTB.4
      }
}

void T0Delay ( )
{
      TCNT0 = 186;      //load TCNT0
      TCCR0 = 0x02;     //Timer0, Normal mode, 1:8 prescaler
      while ((TIFR&(1<<TOV0))==0);  //wait for TOV0 to roll over

      TCCR0 = 0;        //turn off Timer0
      TIFR = 0x1;       //clear TOV0
}
```

XTAL = 8MHz ➔ $T_{machine\ cycle}$ = 1/8 MHz
Prescaler = 1:8 ➔ T_{clock} = 8 × 1/8 MHz = 1 µs
70 µs/1 µs = 70 clocks ➔ 1 + 0xFF – 70 = 0x100 – 0x46 = 0xBA = **186**

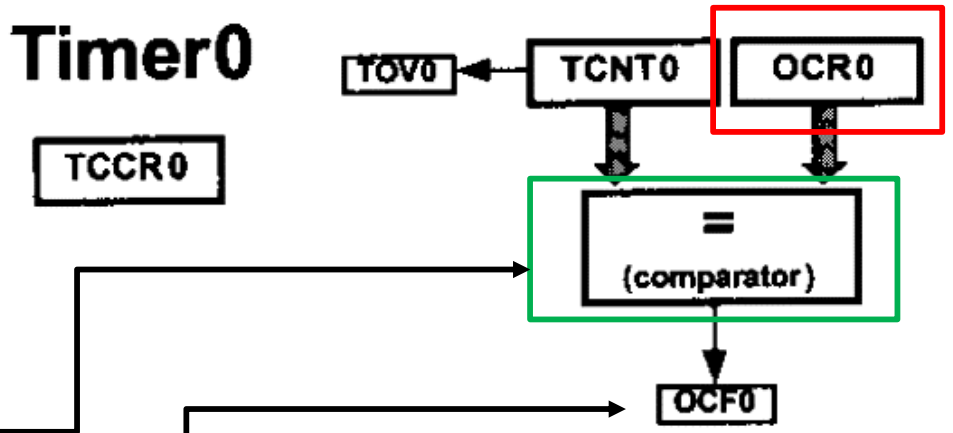
PROGRAMMING TIMER0 IN CTC MODE

TCCR0 (Timer/Counter Control Register)

FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00

WGM00, WGM01

D6	D3	Timer0 mode selector bits
0	0	Normal
0	1	CTC (Clear Timer on Compare Match)
1	0	PWM, phase correct
1	1	Fast PWM

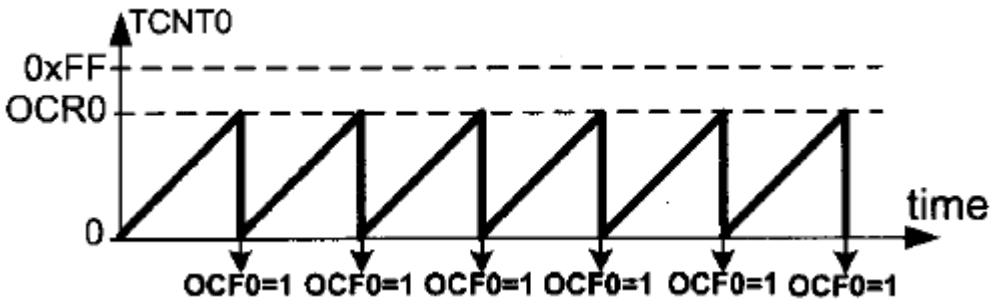


Timer is incremented by a clock up to

Timer will be cleared and OCF0 will be set.

PROGRAMMING TIMER0 IN CTC MODE

1. Load OCR0
2. Load TCCR0 to set the mode and to start the timer0
    1. As the timer (TCCR0) counts up : 00, 01, 02, ….. And reaches content of OCR0
    2. One more clock makes it 0 and OCF0 = 1
3. Stop timer0
4. OCF0 is set.

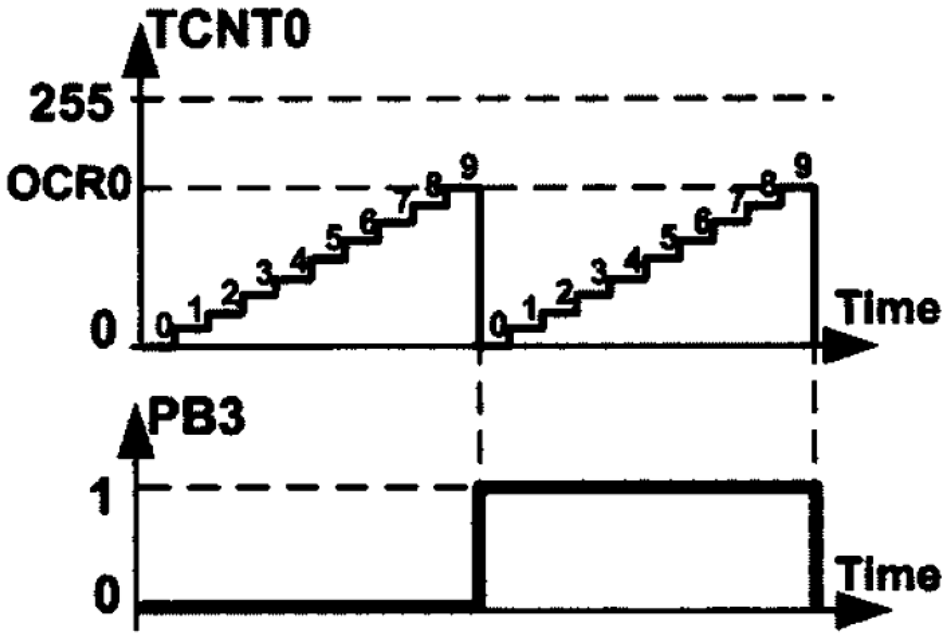


EXAMPLE-3

Find the delay generated by Timer0 when OCR0 is loaded with 9 in CNC mode. XTAL = 8MHz.

Solution:

OCR0 is loaded with 9 and TCNT0 is cleared; Thus, after 9 clocks TCNT0 becomes equal to OCR0. On the next clock, the OCF0 flag is set and the reset occurs. That means the TCNT0 is cleared after 9 + 1 = 10 clocks. Because XTAL = 8 MHz, the counter counts up every 0.125 µs. Therefore, we have 10 × 0.125 µs = 1.25 µs.



EXAMPLE-4

Due to prescaler = 1024 each timer clock lasts 1024 × 0.125 µs = 128 µs. Thus, in order to generate a delay of 25.6 ms we should wait 25.6 ms / 128 µs = 200 clocks. Therefore the OCR0 register should be loaded with 200 – 1 = 199.

Assuming XTAL = 8 MHz, write a program to generate a delay of 25.6 ms. Use Timer0, CTC mode, with prescaler = 1024.

```
int main ()
```

Void T0Delay()

TCCR0 (Timer/Counter Control Register)

FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00

//01001101

CS02:00	D2	D1	D0	Timer0 clock selector
	0	0	0	No clock source (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 source on T0 pin. Clock on falling edge.
	1	1	1	External clock source on T0 pin. Clock on rising edge.

WGM00, WGM01

D6	D3	Timer0 mode selector bits
0	0	Normal
0	1	CTC (Clear Timer on Compare Match)
1	0	PWM, phase correct
1	1	Fast PWM

```
}
```

PROGRAMMING TIMER0 AS COUNTER

- Timers can be used as counters if we provide pulses from outside the chip.
- T0 (PB0) and T1 (PB1) for Timer0 and Timer1

EXAMPLE-5

1 Hz clocl t counter

And displ

```
#include "avr/io.h"
int main ( )
{
        PORTB = 0x01;           //activate pull-up of PB0
        DDRC = 0xFF;            //PORTC as output
        DDRD = 0xFF;            //PORTD as output

        TCCR0 = 0x06;           //output clock source
        TCNT0 = 0x00;

        while (1)                                                    //01001101
        {
                do
                {
                        PORTC = TCNT0;
                }while((TIFR&(0x1<<TOV0))==0);//wait for TOV0 to roll over
                TIFR = 0x1<<TOV0;       //clear TOV0
                PORTD ++;               //increment PORTD
        }
}
```

CS02:00	D2	D1	D0	Tir
	0	0	0	
	0	0	1	
	0	1	0	
	0	1	1	
	1	0	0	
	1	0	1	
	1	1	0	
	1	1	1	

External clock source on T0 pin. Clock on rising edge.

ctor bits

on Compare Match)

ct