CSE 315
Microprocessors & Microcontrollers

Tanvir Ahmed Khan

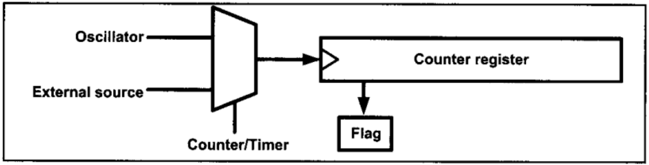
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology.

September 13, 2014

Today's Topic

ATmega16 Timer/Counter

Timer/Counter Basics



Timer Basic

Two ways to generate a time delay,

Timer Basic

Two ways to generate a time delay,

- clear the counter & wait until the counter reaches a certain number

Timer Basic

Two ways to generate a time delay,

- clear the counter & wait until the counter reaches a certain number
- load the counter & wait until the counter overflows

Timer Basic

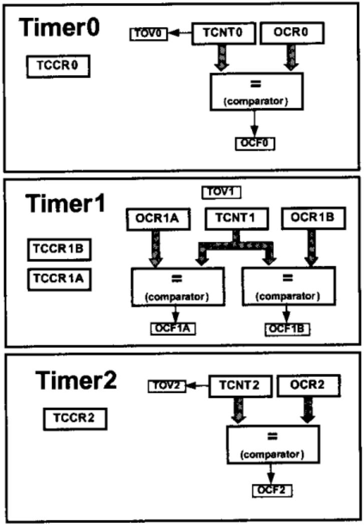
Two ways to generate a time delay,

- clear the counter & wait until the counter reaches a certain number
- **load the counter & wait until the counter overflows**

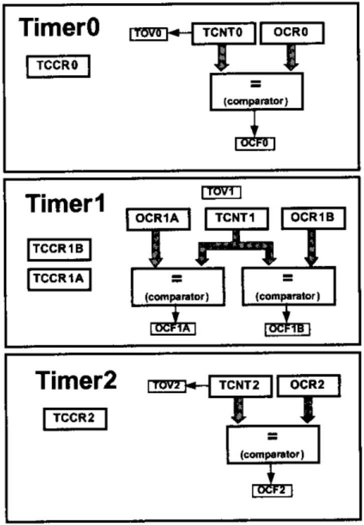
ATmega16 Timers

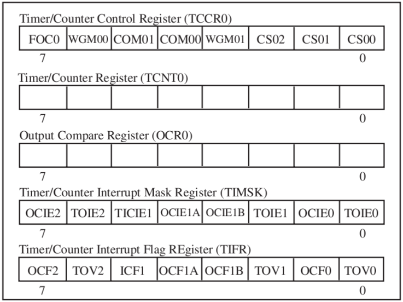
- Timer0, 8-bit
- Timer1, 16-bit
- Timer2, 8-bit

Timer Basic Registers



Timer Basic Registers





Timer/Counter Control Register (TCCR0)

FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00
7							0

Timer/Counter Register (TCNT0)

7							0

Output Compare Register (OCR0)

7							0

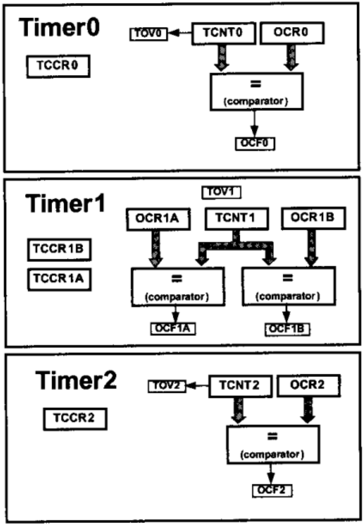
Timer/Counter Interrupt Mask Register (TIMSK)

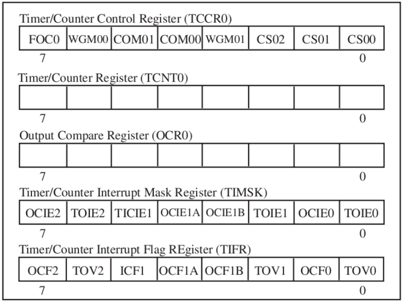
OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0
7							0

Timer/Counter Interrupt Flag REgister (TIFR)

OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0
7							0

Timer Basic Registers





Timer/Counter Control Register (TCCR0)

FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00
7							0

Timer/Counter Register (TCNT0)

7							0

Output Compare Register (OCR0)

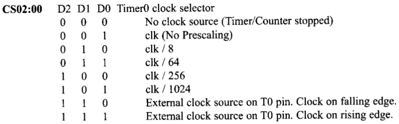
7							0

Timer/Counter Interrupt Mask Register (TIMSK)

OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0
7							0

Timer/Counter Interrupt Flag REgister (TIFR)

OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0
7							0



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.

Programming Timer0

Programming Timer0

- Configure TCCR0,
  - `TCCR0 = 0b00000001;`

Programming Timer0

- Configure TCCR0,
  - `TCCR0 = 0b00000001;`
- Load TCNT0 with initial value,
  - `TCNT0 = 0b11110010;`

Programming Timer0

- Configure TCCR0,
  - `TCCR0 = 0b00000001;`
- Load TCNT0 with initial value,
  - `TCNT0 = 0b11110010;`
- Check continuously TOV0,
  - `while((TIFR & 0b00000001) == 0);`

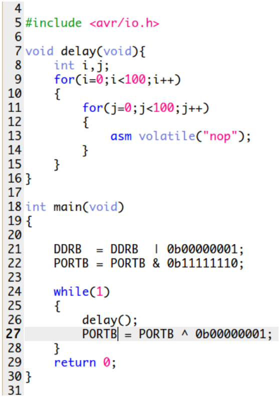
Programming Timer0

- Configure TCCR0,
  - `TCCR0 = 0b00000001;`
- Load TCNT0 with initial value,
  - `TCNT0 = 0b11110010;`
- Check continuously TOV0,
  - `while((TIFR & 0b00000001) == 0);`
- Stop the timer,
  - `TCCR0 = 0;`

Programming Timer0

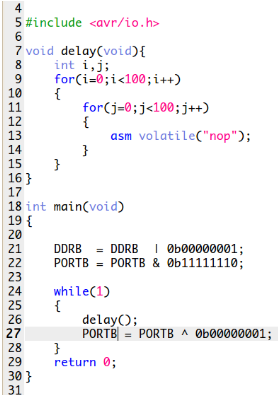
- Configure TCCR0,
    - `TCCR0 = 0b00000001;`
- Load TCNT0 with initial value,
    - `TCNT0 = 0b11110010;`
- Check continuously TOV0,
    - `while((TIFR & 0b00000001) == 0);`
- Stop the timer,
    - `TCCR0 = 0;`
- Clear TOV0,
    - `TIFR = TIFR | 0b00000001;`

Simple Blink Example

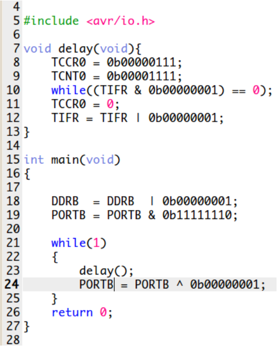


```c

#include <avr/io.h>

void delay(void){
    int i,j;
    for(i=0;i<100;i++)
    {
        for(j=0;j<100;j++)
        {
            asm volatile("nop");
        }
    }
}

int main(void)
{

    DDRB  = DDRB  | 0b00000001;
    PORTB = PORTB & 0b11111110;

    while(1)
    {
        delay();
        PORTB = PORTB ^ 0b00000001;
    }
    return 0;
}

```

Simple Blink Example



```c

#include <avr/io.h>

void delay(void){
    int i,j;
    for(i=0;i<100;i++)
    {
        for(j=0;j<100;j++)
        {
            asm volatile("nop");
        }
    }
}

int main(void)
{

    DDRB  = DDRB  | 0b00000001;
    PORTB = PORTB & 0b11111110;

    while(1)
    {
        delay();
        PORTB = PORTB ^ 0b00000001;
    }
    return 0;
}

```



```c

#include <avr/io.h>

void delay(void){
    TCCR0 = 0b00000111;
    TCNT0 = 0b00001111;
    while((TIFR & 0b00000001) == 0);
    TCCR0 = 0;
    TIFR = TIFR | 0b00000001;
}

int main(void)
{

    DDRB  = DDRB  | 0b00000001;
    PORTB = PORTB & 0b11111110;

    while(1)
    {
        delay();
        PORTB = PORTB ^ 0b00000001;
    }
    return 0;
}

```

Practice Problems

Practice Problems

- Finding the delay for a specific TCNT0 value

Practice Problems

- Finding the delay for a specific TCNT0 value
  - Clock = 8 MHz, TCNT0 = 0x3E, TCCR0 = 1

Practice Problems

- Finding the delay for a specific TCNT0 value
  - Clock = 8 MHz, TCNT0 = 0x3E, TCCR0 = 1
  - Clock = 8 MHz, TCNT0 = 0x00, TCCR0 = 5

Practice Problems

- Finding the delay for a specific TCNT0 value
  - Clock = 8 MHz, TCNT0 = 0x3E, TCCR0 = 1
  - Clock = 8 MHz, TCNT0 = 0x00, TCCR0 = 5
- Finding the value of TCNT0 for a specific delay

Practice Problems

- Finding the delay for a specific TCNT0 value
  - Clock = 8 MHz, TCNT0 = 0x3E, TCCR0 = 1
  - Clock = 8 MHz, TCNT0 = 0x00, TCCR0 = 5
- Finding the value of TCNT0 for a specific delay
  - Clock = 8 MHz, output signal frequency = 16 KHz

Practice Problems

- Finding the delay for a specific TCNT0 value
  - Clock = 8 MHz, TCNT0 = 0x3E, TCCR0 = 1
  - Clock = 8 MHz, TCNT0 = 0x00, TCCR0 = 5
- Finding the value of TCNT0 for a specific delay
  - Clock = 8 MHz, output signal frequency = 16 KHz
  - Clock = 8 MHz, output signal frequency = 125 Hz, with pre-scaler = 256

Reference

- The avr microcontroller & embedded system, *Chapter 9*
  - Muhammad Ali Mazidi
  - Sarmad Naimi
  - Sepehr Naimi