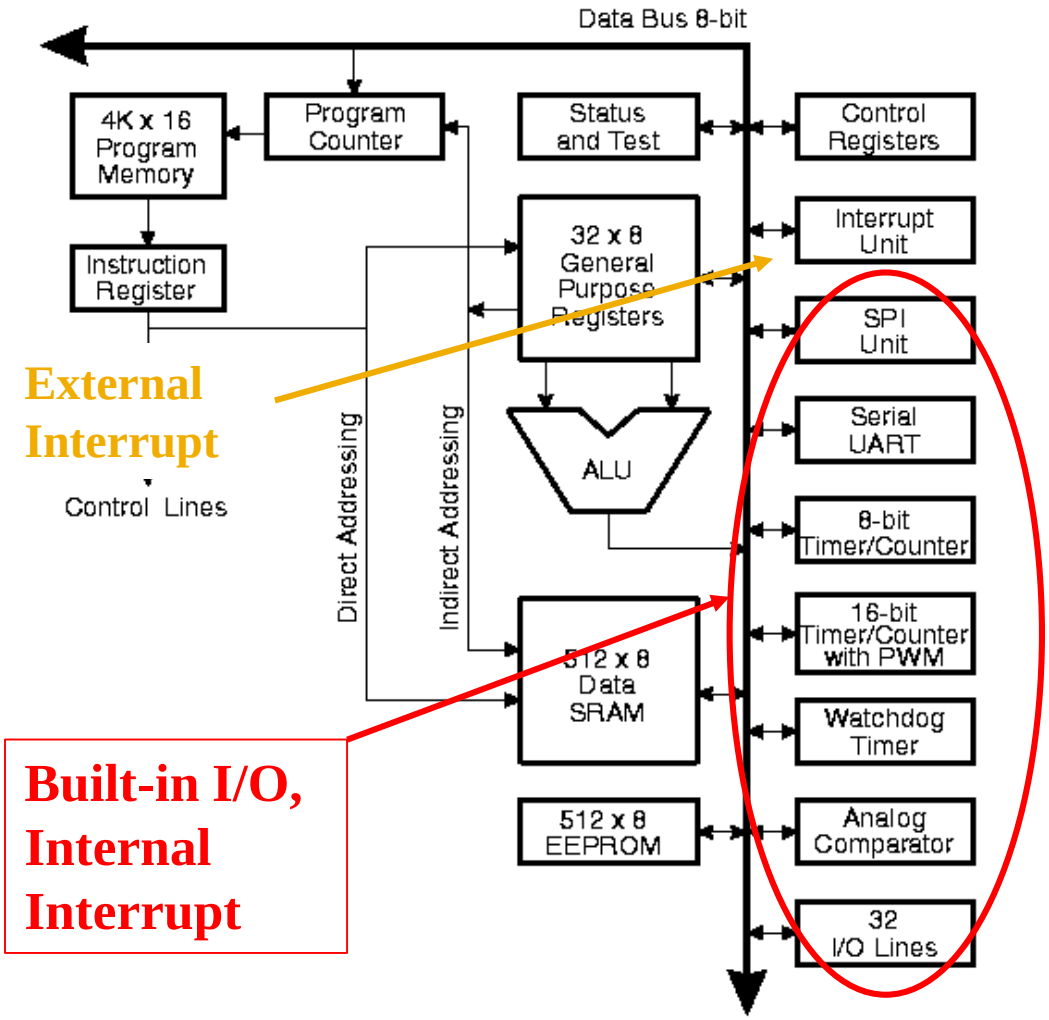
AVR Timer/Counter Interrupt

Erdefi Rakun

Sources of interrupts in AVR

Covered by ICO:

1. **External Interrupt**
2. **Internal Interrupt / Timer Interrupt**



Timer / Counter ?


dreamstime.com

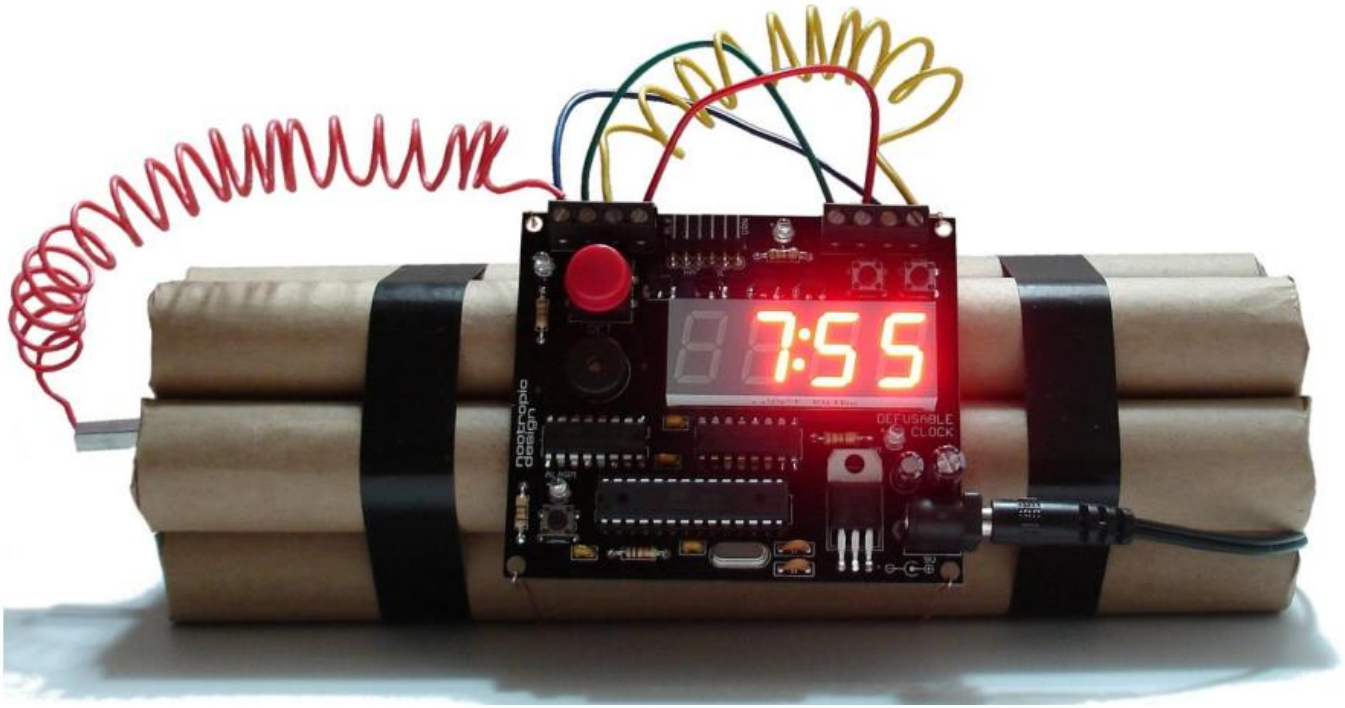
Timer / Counter

- Many applications need to count an event or generate time delays.
- Counter registers in microcontrollers for this purpose.
- Timer is a simple counter.
- Input clock and operation of timer is independent of the program execution.
- AVR has two 8-bit and one 16-bit timer
- AVR can be configured to execute interrupts if a timer event has occurred.

How the Timer Works.

- Increment the counter variable (counter register, TCNT).
- In order to increment the counter register, the timer has to access the clock source.
- The frequency of the clock source determines the increment (rate of change) of the counter register.

Timer Interrupt?



Tim Dosen POK 2017

Timer Interrupt

- Interrupt is activated by the timer/counter.
- If the timer/counter reaches a certain condition (see chart), the interrupt will activate.
- There are 3 types of timer interrupts:

Addr	Source	Definition
$003	Timer1 Capt	Timer/Counter` Capture Event
$004	Timer1 CompA	Timer/Counter1 Compare Match A
$005	Timer1 CompB	Timer/Counter1 Compare Match B
$006	Timer1 Ovf	Timer/Counter1 Overflow
$007	Timero Ovf	Timer/Countero Overflow
$00E	Timero Comp	Timer/Countero Compare

Timer Event

AVR timer can be specified to monitor several events:

- **Timer overflow** means that the counter has counted up to its **max value** and is reset to zero in the next timer clock cycle. The timer over flow event causes **Timer Overflow Flag (TOVx)** to be set in TIFR.
- **Compare match** means that the timer will be checked against the preloaded **Output Compare Register (OCRx).** When the timer reaches the compare value, the corresponding **Output Compare Flag (OCFx)** in the TIFR register is set.
- **Input capture** means that there is a signal **change in AVR input** pin. This signal change causes timer value to be read and saved in the **Input Capture Register (ICRx),** and the **Input Capture Flag (ICFx)** in the TIFR will be set.

Timer Event Notification

- Timer operates independently of the program execution.
- For each timer event there is a corresponding status flag in the Timer Interrupt Flag register (TIFR).
- The occurrence of timer events require notification of processor to trigger the execution of corresponding action.
- Ways to monitor timer event:
  - polling
  - Interrupt
  - Changing the level of output pin automatically.

Timer Event Notification

Polling

```
loop:
    in      r16, TIFR       ; load TIFR in register 16
    sbrs    r16, TOVO       ; skip next instruction if bit (zero) in
                            ; register (r16) is set
    rjmp    loop            ; jump to loop if no Timero
                            ;overflow occurred
                            ; Event Service Code starts here
```

Timer Event Notification

SBRS – Skip if Bit in Register is Set

Description:

This instruction tests a single bit in a register and skips the next instruction if the bit is set.

Operation:

(i) If Rr(b) = 1 then PC ← PC + 2 (or 3) else PC ← PC + 1

	Syntax:	Operands:	Program Counter:
(i)	SBRS Rr,b	$0 \le r \le 31, 0 \le b \le 7$	PC ← PC + 1, Condition false - no skip PC ← PC + 2, Skip a one word instruction PC ← PC + 3, Skip a two word instruction

16-bit Opcode:

1111	111r	rrrr	0bbb

Timer Event Notification - 2

Interrupt

```
ldi   r16, 1<<OCIE2
out  TIMSK,r16      ; Enable timer output compare interrupt
sei                 ; Enable global interrupts
```

Alamat dan fungsi I/O

Address	Hex Name	Function
$3F ($5F)	SREG	Status Register
$3E ($5E)	SPH	Stack Pointer High
$3D ($5D)	SPL	Stack Pointer Low
$3B ($5B)	GIMSK	General Interrupt MaSK Register
$3A ($5A)	GIFR	General Interrupt Flag Register
$39 ($59)	TIMSK	Timer/Counter Interrupt MaSK Register
$38 ($58)	TIFR	Timer/Counter Interrupt Flag Register
$35 ($55)	MCUCR	MCU general Control Register
$33 ($53)	TCCR0	Timer/Counter 0 Control Register
$32 ($52)	TCNT0	Timer/Counter 0 (8-bit)
$2F ($4F)	TCCR1A	Timer/Counter 1 Control Register A
$2E ($4E)	TCCR1B	Timer/Counter 1 Control Register B
$2D ($4D)	TCNT1H	Timer/Counter 1 High Byte
$2C ($4C)	TCNT1L	Timer/Counter 1 Low Byte
$2B ($4B)	OCR1AH	Output Compare Register A High Byte
$2A ($4A)	OCR1AL	Output Compare Register A Low Byte
$29 ($49)	OCR1AH	Output Compare Register B High Byte
$28 ($48)	OCR1AL	Output Compare Register B Low Byte
$25 ($45)	ICR1H	T/C 1 Input Capture Register High Byte
$24 ($44)	ICR1L	T/C 1 Input Capture Register Low Byte
$21 ($41)	WDTCR	Watchdog Timer Control Register

Basic registers of timers

- For each of the timers, there is a **TCNTn (Timer/Counter)** register
- TCNTn register is **a counter**. It **counts up with each pulse**.
- You can **load** a value into TCNTn register or **read** its value.

Address	Hex Name	Function
$3F ($5F)	SREG	Status Register
$3E ($5E)	SPH	Stack Pointer High
$3D ($5D)	SPL	Stack Pointer Low
$3B ($5B)	GIMSK	General Interrupt MaSK Register
$3A ($5A)	GIFR	General Interrupt Flag Register
$39 ($59)	TIMSK	Timer/Counter Interrupt MaSK Register
$38 ($58)	TIFR	Timer/Counter Interrupt Flag Register
$35 ($55)	MCUCR	MCU general Control Register
$33 ($53)	TCCR0	Timer/Counter 0 Control Register
$32 ($52)	TCNT0	Timer/Counter 0 (8-bit)
$2F ($4F)	TCCR1A	Timer/Counter 1 Control Register A
$2E ($4E)	TCCR1B	Timer/Counter 1 Control Register B
$2D ($4D)	TCNT1H	Timer/Counter 1 High Byte
$2C ($4C)	TCNT1L	Timer/Counter 1 Low Byte
$2B ($4B)	OCR1AH	Output Compare Register A High Byte
$2A ($4A)	OCR1AL	Output Compare Register A Low Byte
$29 ($49)	OCR1AH	Output Compare Register B High Byte
$28 ($48)	OCR1AL	Output Compare Register B Low Byte
$25 ($45)	ICR1H	T/C 1 Input Capture Register High Byte
$24 ($44)	ICR1L	T/C 1 Input Capture Register Low Byte
$21 ($41)	WDTCR	Watchdog Timer Control Register

Basic registers of timers

- Each timer has a **TOVn (Timer Overflow) flag**.

Bit	7	6	5	4	3	2	1	0	
	TOV1	OCF1A	OCF1B	–	ICF1	–	TOV0	OCF0	TIFR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

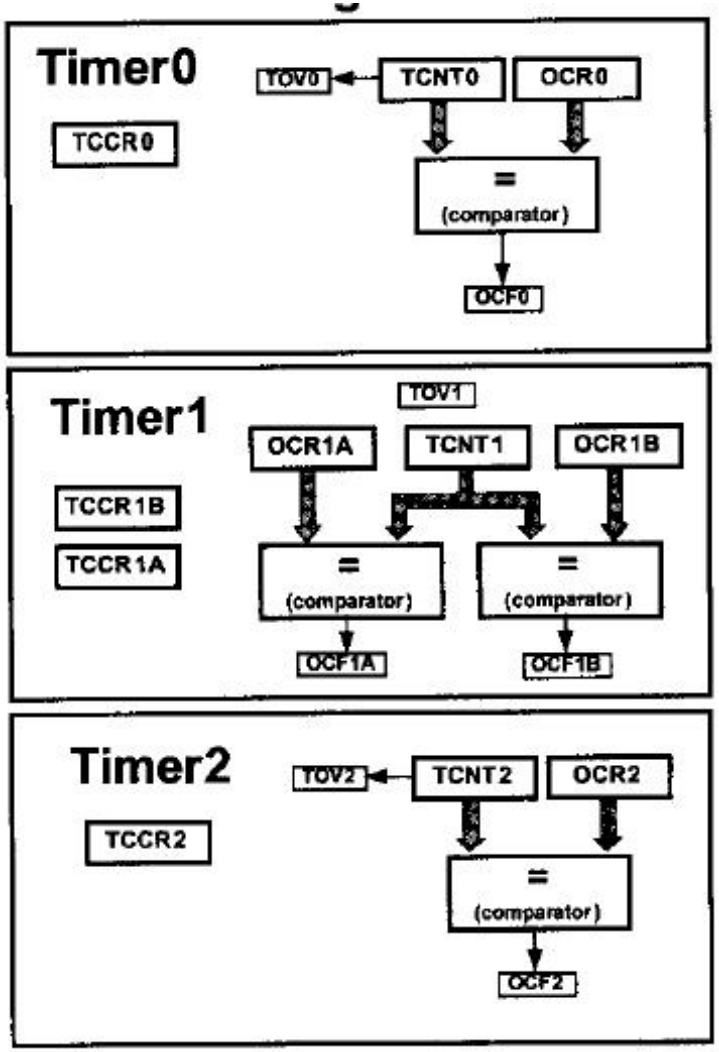
- Read / write from/to timer register using **IN** and **OUT** instructions.

Basic registers of timers

- Each timer also has the **TCCRn (Timer/Counter control register)** for setting modes of operation.
- Each timer also has an **OCRn (Output Compare Register).** The content of OCRn is compared with the content of TCNTn. When they are equal the **OCFn** (Output compare Flag) flag will be set.

Address	Hex Name	Function
$3F ($5F)	SREG	Status Register
$3E ($5E)	SPH	Stack Pointer High
$3D ($5D)	SPL	Stack Pointer Low
$3B ($5B)	GIMSK	General Interrupt MaSK Register
$3A ($5A)	GIFR	General Interrupt Flag Register
$39 ($59)	TIMSK	Timer/Counter Interrupt MaSK Register
$38 ($58)	TIFR	Timer/Counter Interrupt Flag Register
$35 ($55)	MCUCR	MCU general Control Register
$33 ($53)	TCCR0	Timer/Counter 0 Control Register
$32 ($52)	TCNT0	Timer/Counter 0 (8-bit)
$2F ($4F)	TCCR1A	Timer/Counter 1 Control Register A
$2E ($4E)	TCCR1B	Timer/Counter 1 Control Register B
$2D ($4D)	TCNT1H	Timer/Counter 1 High Byte
$2C ($4C)	TCNT1L	Timer/Counter 1 Low Byte
$2B ($4B)	OCR1AH	Output Compare Register A High Byte
$2A ($4A)	OCR1AL	Output Compare Register A Low Byte
$29 ($49)	OCR1AH	Output Compare Register B High Byte
$28 ($48)	OCR1AL	Output Compare Register B Low Byte
$25 ($45)	ICR1H	T/C 1 Input Capture Register High Byte
$24 ($44)	ICR1L	T/C 1 Input Capture Register Low Byte
$21 ($41)	WDTCR	Watchdog Timer Control Register

Basic registers of timers (cont.)



Basic registers of timers (cont.)
TCCRx

Bit	7	6	5	4	3	2	1	0
	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00
Read/Write	W	RW	RW	RW	RW	RW	RW	RW
Initial Value	0	0	0	0	0	0	0	0

FOC0 D7 Force compare match: This is a write-only bit, which can be used while generating a wave. Writing 1 to it causes the wave generator to act as if a compare match had occurred.

WGM00, WGM01

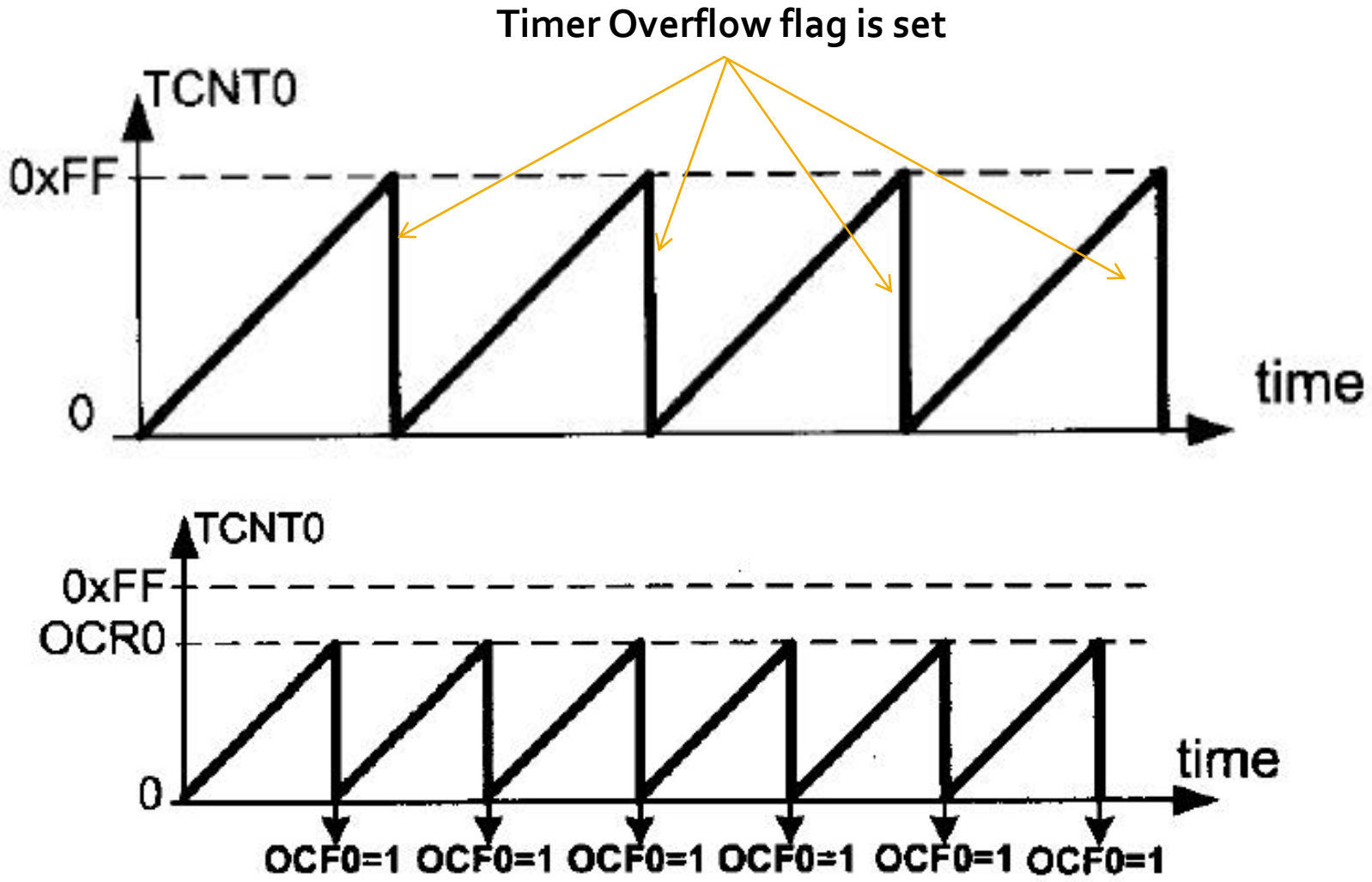
D6	D3	Timer0 mode selector bits
0	0	Normal
0	1	CTC (Clear Timer on Compare Match)
1	0	PWM, phase correct
1	1	Fast PWM

COM01:00 D5 D4 Compare Output Mode:
These bits control the waveform generator (see Chapter 15).

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.

Normal Mode


Timer Overflow flag is set
TCNT0
0xFF
0
time
TCNT0
0xFF
OCR0
0
time
OCF0=1 OCF0=1 OCF0=1 OCF0=1 OCF0=1 OCF0=1

Basic registers of timers (cont.)

- Timer/Counter Interrupt Mask Register (TIMSK) → penentu jenis timer

Bit	7	6	5	4	3	2	1	0	
	TOIE1	OCIE1A	OCIE1B	–	TICIE1	–	TOIE0	OCIE0	TIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- Timer/Counter Interrupt Flag Register (TIFR)→ penanda timer even

Bit	7	6	5	4	3	2	1	0	
	TOV1	OCF1A	OCF1B	–	ICF1	–	TOV0	OCF0	TIFR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

TIMSK

- Used to control which **interrupts are "valid"**
- **TOIE1: Timer Overflow Interrupt Enable (Timer 1);** If this bit is set and if global interrupts are enabled, the micro will jump to the Timer Overflow 1 interrupt vector upon Timer 1 Overflow.
- **OCIE1A**: **Output Compare Interrupt Enable 1 A**; If set and if global Interrupts are enabled, the micro will jump to the Output Compare A Interrupt vetor upon compare match.
- **TICIE1: Timer 1 Input Capture Interrupt Enable**; If set and if global Interrupts are enabled, the micro will jump to the Input Capture Interrupt vector upon an Input Capture event.
- **TOIE0: Timer Overflow Interrupt Enable (Timer 0**); Same as TOIE1, but for the 8-bit Timer 0.

TIFR

It **holds the Timer Interrupt Flags** corresponding to their enable bits in TIMSK

Bit	7	6	5	4	3	2	1	0
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

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

OCF0 D1 Timer0 output compare flag bit
0 = compare match did not occur.
1 = compare match occurred.

TOV1 D2 Timer1 overflow flag bit

OCF1B D3 Timer1 output compare B match flag

OCF1A D4 Timer1 output compare A match flag

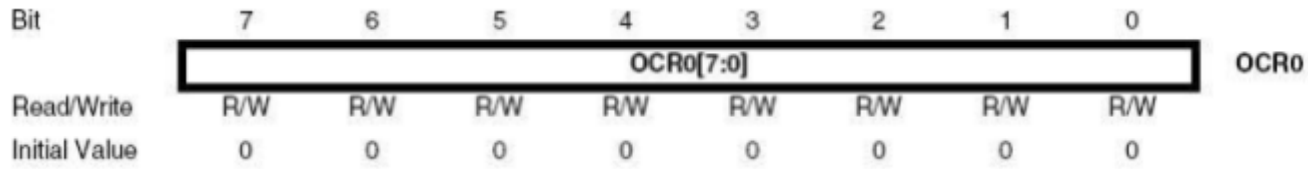
ICF1 D5 Input Capture flag

TOV2 D6 Timer2 overflow flag

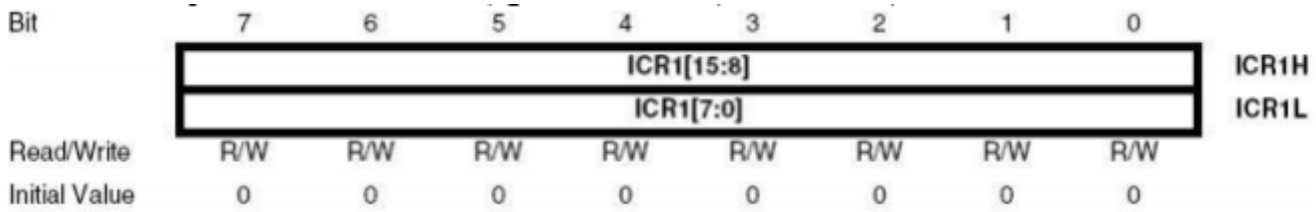
OCF2 D7 Timer2 output compare match flag

Timer Counter Register Cont.

- Output Compare Register (OCR1A/B)



- Input Capture Register (ICR1)



Clock settings in TCCR

TCCRx			Synchronous Timer0 & Timer1 P_{CK} = CK	Synchronous/Asynchronous Timer2 P_{CK2} = f (AS2)
Bit 2	Bit 1	Bit 0		
CSx2	CSx1	CSx0	$T_{CK0,1}$	T_{CK2}
0	0	0	0 (Timer Stopped)	0 (Timer Stopped)
0	0	1	P_{CK} (System Clock)	P_{CK2} (System Clock/Asynchronous Clock)
0	1	0	$P_{CK}/8$	$P_{CK2}/8$
0	1	1	$P_{CK}/64$	$P_{CK2}/32$
1	0	0	$P_{C}/256$	$P_{CK2}/64$
1	0	1	$P_{CK}/1024$	$P_{CK2}/128$
1	1	0	External Pin Tx falling edge	$P_{CK2}/256$
1	1	1	External Pin Tx rising edge	$P_{CK2}/1024$

$$TOV_{CK} = \frac{f_{CK}}{MaxVal}$$

TOV_{ck} = frekuensi timer overflow dalam satu detik
f_{ck} = frekuensi clock (P_{ck}/N)
MaxVal = nilai maksimal TCNT ($2^{\text{jumlah bit pada TCNT}}$)

Sample computing time

- Clock = 4MHz
- Required delay 184 ms
- Without precaler, need 736.000 cycles
- **timer0**, counting from 0 to 255
- **timer1**, counting from 0 to 65,535

Prescaler	Clock Frequency	Timer Count
8	500 KHz	92.000
64	62.5 KHz	11.500
256	15.625 KHz	2875
1024	3906.25 Hz	718.75

Example

Assume that the CPU is running with f_{CPU} = 3.69 MHz and the resolution of the timer is 8 bit (MaxVal = 256). UsingTimer/Counter Overflow 0 withTCCR activate CS01.

1. How many times will timer overflow appear in one second?
2. If the TCCR setting is changed by activating CS01 and CS00, what will the change of setting's effect be on the timer/counter overflow interrupt?

Example

- MaxVal = 256
- f_{CPU} = 3.69 MHz
- TCCR0 [2:0] = 010 $\rightarrow P_{CK}/8$
- TCCR0 [2:0] = 011 $\rightarrow P_{CK}/64$

$$TOV_{CK} = \frac{f_{CK}}{MaxVal} = \frac{(\frac{P_{CK}}{P_{Val}})}{MaxVal} = \frac{P_{CK}}{(P_{Val} * MaxVal)}$$

$$TOV_{CK} = \frac{3.69Mhz}{(8*256)} = \sim 1802 \rightarrow \text{every 0.55 ms an overflow occurs}$$

$$TOV_{CK} = \frac{3.69Mhz}{(64*256)} = \sim 225 \rightarrow \text{every 4.4 ms an overflow occurs}$$

- Range timer overflow every 69µs – 71 ms

Steps to activate Timer/Counter Overflow Interrupt

1. Setting the values of TCCR0/TCCR1A/TCCR1B
2. Setting the values of TIFR
3. Setting the clock in TIMSK
4. Activate global Interrupt using SEI instruction

Steps to activate Timer/Counter Compare Interrupt

1. Setting the values of TCCR0/TCCR1A/TCCR1B
2. Setting the values of TIFR
3. Setting the clock in TIMSK
4. Load OCR0/OCR1A/OCR1B with a value [0..MaxVal] which the timer will be checked against every timer cycle.
5. Activate global Interrupt using SEI instruction

Sample programs

Int_OV.asm	: Timer0 overflow
Counter_Compare.asm	: Timer0 compare
T0vComp.asm	: Timer0 overflow & compare
T1_OV	: Timer1 overflow
T1_COMP	: Timer1 compare
T1_OVCOMP	: Timer1 overflow & compare
OVCompExint	: Overflow, Compare, ExtInt

Sample programs

- No Pre scaling clock (using system clock)
- Uses Timer 0 to generate 2 types of timer interrupt: overflow and compare
- The interrupt will appear twice: it will complement the data at port B at the 128^{th} tick, and complement the data at port A when overflow happens.

Sample Programs

- Uses system clock / 8
- Uses Timer 1 to generate 2 types of timer interrupt: overflow and compare
- The interrupt will appear twice: it will complement the data at port B at the 65280^{th} tick, and complement the data at port A when overflow happens.