

Modes of operation (continued..)

- The waveform generated will have a **maximum frequency** of $f_{OC1A} = f_{clk_I/O}/2$ when OCR1A is set to zero(0x0000).
- The waveform frequency is defined by the following equation:

$$f_{OCnA} = \frac{f_{clk_I/O}}{2 \cdot N \cdot (1 + OCRnA)}$$

The N variable represents the prescaler factor (1, 8, 64, 256, or 1024).

Fast Pulse Width Modulation /Fast PWM Mode

- Fast PWM mode (WGM1,3:0 = 5, 6, 7, 14, or 15)
- provides a high frequency PWM waveform generation option.
- Single-slope operation: The operating frequency of the fast PWM mode can be twice as high as the phase correct and phase and frequency correct PWM modes that use dual-slope operation.
- In non-inverting Compare Output mode, the Output Compare (OC1x) is cleared on the Compare Match between TCNT1 and OCR1x, and set at BOTTOM.
- In inverting Compare Output mode output is set on Compare Match and cleared at BOTTOM.

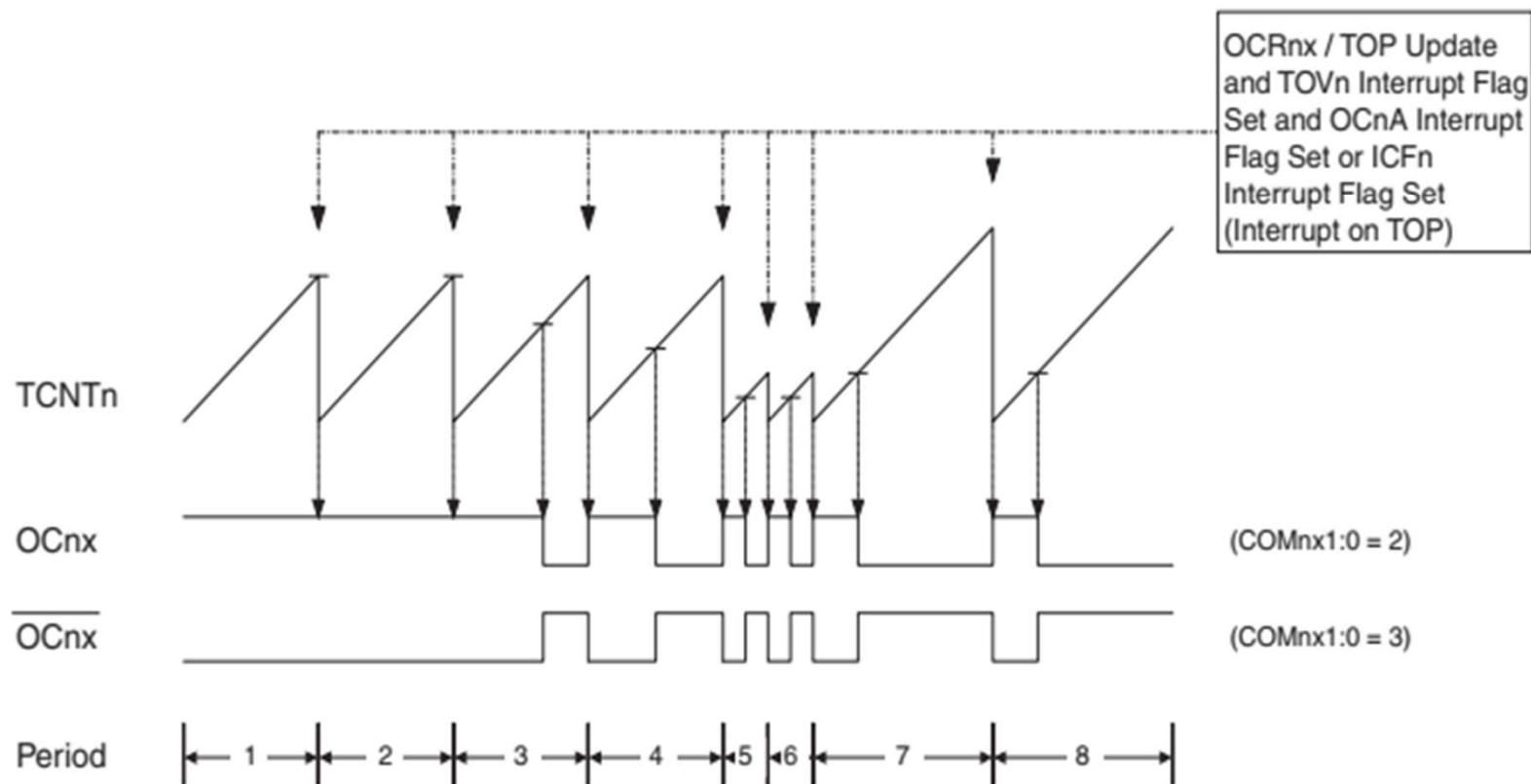
- This high frequency makes fast PWM well suited for
 - ✓ Power regulation,
 - ✓ Rectification
 - ✓ DAC applications.
 - ✓ Allows small component size (coils, capacitors) and hence reduced cost.
- **PWM resolution:**
 - ✓ resolution for fast PWM can be fixed to 8-bit, 9-bit, or 10-bit, or defined by either ICR1 or OCR1A.
 - ✓ The minimum resolution allowed is 2-bit (ICR1 or OCR1A set to 0x0003), and the maximum resolution is 16-bit (ICR1 or OCR1A set to MAX).
 - ✓ The PWM resolution in bits can be calculated by using the following equation:

$$R_{FPWM} = \frac{\log(TOP + 1)}{\log(2)}$$

- In fast PWM mode
 - ✓ The counter is incremented until the counter value matches either one of the fixed values 0x00FF, 0x01FF, or 0x03FF (WGM1,3:0 = 5, 6, or 7),
 - ✓ The value in ICR1 (WGM13:0 = 14)
 - ✓ The value in OCR1A (WGM13:0 = 15).
 - ✓ The counter is then cleared at the following timer clock cycle.

FAST PWM TIMING MODE

Figure 38. Fast PWM Mode, Timing Diagram



Timing diagram description

- In fast PWM mode when OCR1A or ICR1 is used to define TOP.
- The small horizontal line marks on the TCNT1 slopes represent compare matches between OCR1x and TCNT1.
- The OC1x Interrupt Flag will be set when a Compare Match occurs.
- Setting the **COM1x1:0 bits to 2** will produce a **non-inverted PWM**
- An **inverted PWM** output can be generated by setting the **COM1x1:0 to 3**

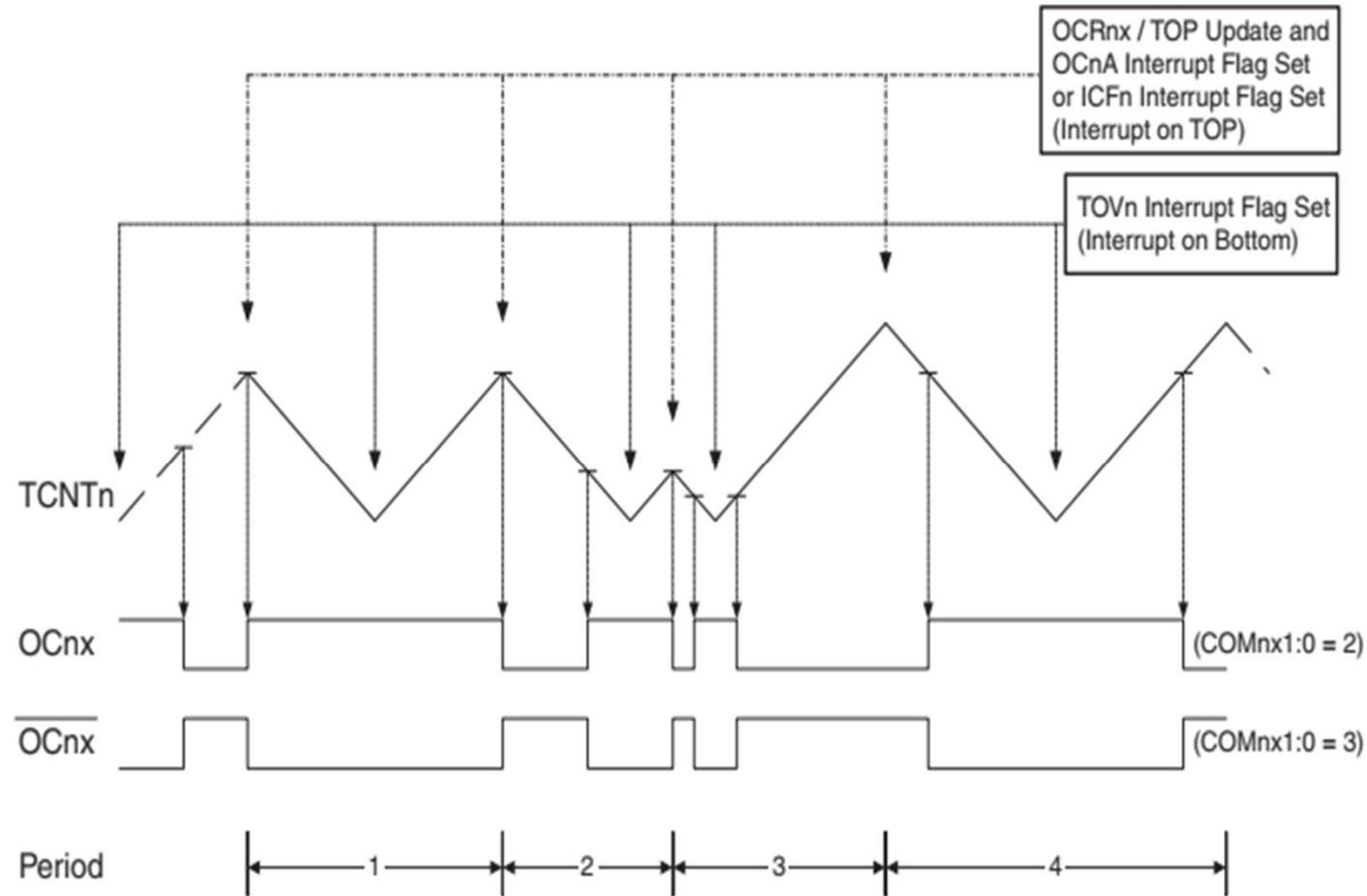
The PWM frequency for the output can be calculated by the following equation:

$$f_{OCnxPWM} = \frac{f_{clk_I/O}}{N \cdot (1 + TOP)}$$

The N variable represents the prescaler divider (1, 8, 64, 256, or 1024).

Phase correct PWM Mode

- Phase correct PWM mode (WGM1,3:0 = 1, 2, 3,10, or 11) provides a high resolution phase correct PWM waveform generation option.
- Based on dual slope operation.
- In non-inverting Compare Output mode, the Output Compare (OC1x) is cleared on the Compare Match between TCNT1 and OCR1x while up-counting and set on the Compare Match while down-counting.
- In inverting Output Compare mode, the operation is inverted i.e OC1x is set while down-counting and clear while up-counting.



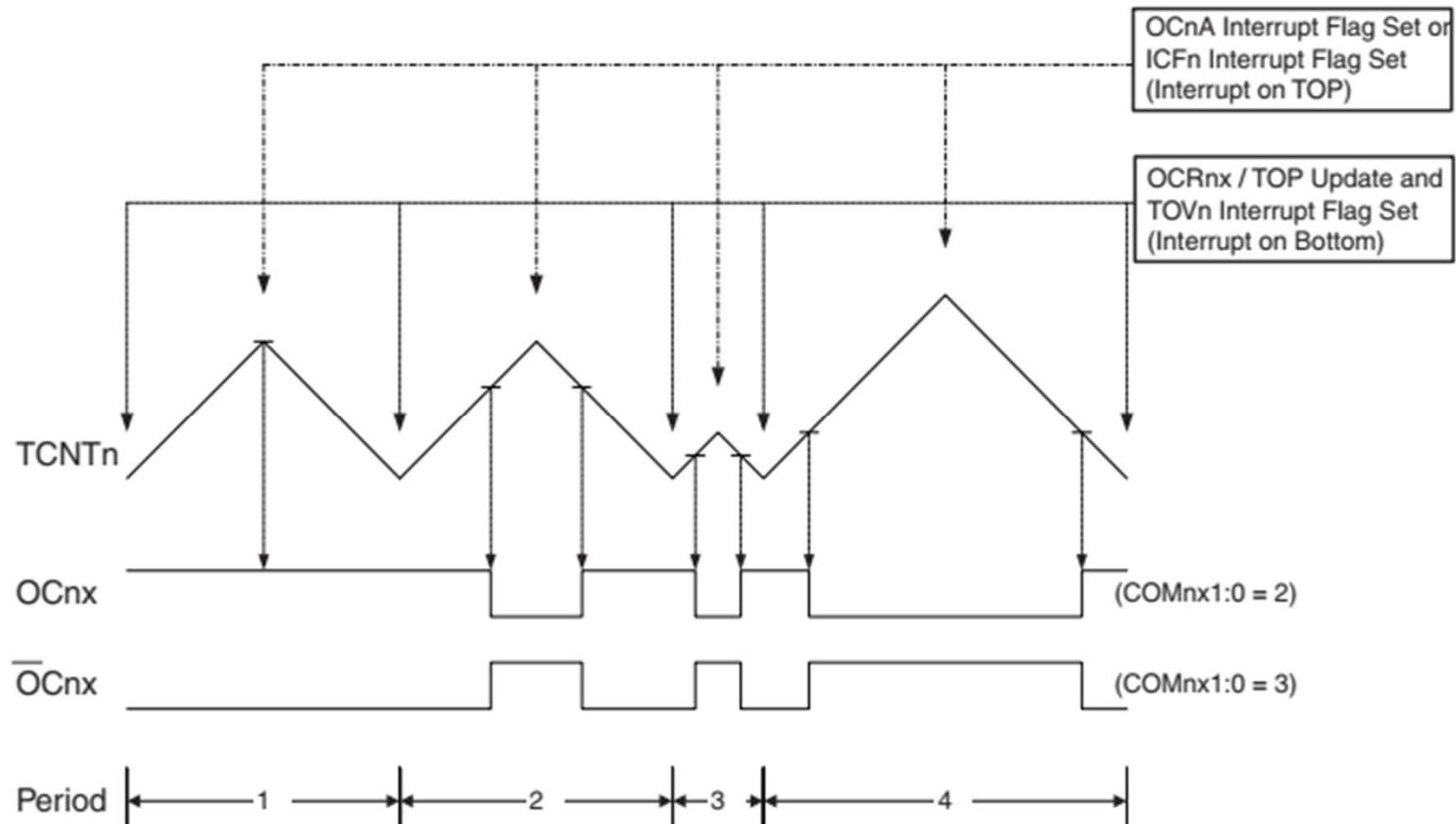
- For inverted PWM the output will have the opposite logic values.
- If OCR1A is used to define the TOP value (WMG13:0 = 11) and COM1A1:0 = 1, the OC1A output will toggle with a 50% duty cycle.

Phase and Frequency Correct PWM Mode

- Phase and frequency correct PWM mode (WGM13:0 = 8 or 9) provides a high resolution phase and frequency correct PWM waveform generation option.
- Based on dual-slope operation.
- The main difference between the phase correct, and the phase and frequency correct PWM mode is the time the OCR1x Register is updated by the OCR1xBufferRegister.

- In phase and frequency correct PWM mode the counter is incremented until the counter value matches either the value in ICR1 (WGM1,3:0 = 8), or the value in OCR1A (WGM1,3:0 = 9).
- The counter has then reached the TOP and changes the count direction. The TCNT1 value will be equal to TOP for one timer clock cycle.
- The output generated is, in contrast to the Phase Correct mode, symmetrical in all periods. Since the OCR1x Registers are updated at BOTTOM, the length of the rising and the falling slopes will always be equal. This gives symmetrical output pulses and is therefore frequency correct.

Phase and Frequency Correct PWM Timing Diagram



16 bit Timer/Counter Register description

- Timer/Counter 1 Control Register A – TCCR1A :

Bit	7	6	5	4	3	2	1	0	TCCR1A
	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	
Read/Write	R/W	R/W	R/W	R/W	W	W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- Bit 7:6 – COM1A1:0: Compare Output Mode for channel A
- Bit 5:4 – COM1B1:0: Compare Output Mode for channel B

Table 36. Compare Output Mode, Non-PWM

COM1A1/ COM1B1	COM1A0/ COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	Toggle OC1A/OC1B on Compare Match
1	0	Clear OC1A/OC1B on Compare Match (Set output to low level)
1	1	Set OC1A/OC1B on Compare Match (Set output to high level)

Compare output mode (non -PWM)

Note: Non-PWM means WGM1,3:0 bits are set to normal or CTC mode

COM1x1:0 functionality when WGM1,3:0 bits are set to Fast PWM

Table 37. Compare Output Mode, Fast PWM⁽¹⁾

COM1A1/ COM1B1	COM1A0/ COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	WGM13:0 = 15: Toggle OC1A on Compare Match, OC1B disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B disconnected.
1	0	Clear OC1A/OC1B on Compare Match, set OC1A/OC1B at BOTTOM, (non-inverting mode)
1	1	Set OC1A/OC1B on Compare Match, clear OC1A/OC1B at BOTTOM, (inverting mode)

COM1x1:0 bit functionality when the WGM1,3:0 bits are set to the phase correct or the phase and frequency correct, PWM mode.

Table 38. Compare Output Mode, Phase Correct and Phase and Frequency Correct PWM⁽¹⁾

COM1A1/ COM1B1	COM1A0/ COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	WGM13:0 = 9 or 14: Toggle OC1A on Compare Match, OC1B disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B disconnected.
1	0	Clear OC1A/OC1B on Compare Match when up-counting. Set OC1A/OC1B on Compare Match when downcounting.
1	1	Set OC1A/OC1B on Compare Match when up-counting. Clear OC1A/OC1B on Compare Match when downcounting.

- Bit 3 – FOC1A: Force Output Compare for channel A
 - Bit 2 – FOC1B: Force Output Compare for channel B
-
- The FOC1A/FOC1B bits are only active when the WGM13:0 bits specifies a **non-PWM mode**.
 - When writing a logical one to the FOC1A/FOC1B bit, an immediate Compare Match is forced on the waveform generation unit. The OC1A/OC1B output is changed according to its COM1x1:0 bits setting.
 - The FOC1A/FOC1B bits are always read as zero.

- **Bit 1:0 – WGM11:0: Waveform Generation Mode**
- Combined with the WGM13:2 bits found in the TCCR1B Register, these bits control the counting sequence of the counter, the source for maximum (TOP) counter value, and what type of waveform generation to be used.

Waveform generation Mode bit Description

Mode	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	Timer/Counter Mode of Operation ⁽¹⁾	TOP	Update of OCR1x	TOV1 Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCR1A	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP

Mode	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	Timer/Counter Mode of Operation ⁽¹⁾	TOP	Update of OCR1x	TOV1 Flag Set on
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICR1	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCR1A	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICR1	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCR1A	TOP	BOTTOM
12	1	1	0	0	CTC	ICR1	Immediate	MAX
13	1	1	0	1	(Reserved)	-	-	-
14	1	1	1	0	Fast PWM	ICR1	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCR1A	BOTTOM	TOP

TCCR1B

Bit	7	6	5	4	3	2	1	0	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	

1. **ICNC1:** input capture Noise canceler (active when set “1”)
2. **ICES1:** Input capture edge select(When the ICES1 bit is written to zero, a falling (negative) edge is used as trigger, and when the ICES1 bit is written to one, a rising (positive) edge will trigger the capture.
3. **CS12:10:** clock select bits (select prescalers)

Numerical 1

Find the values for TCCR1A and TCCR1B if we want to program Timer1 in mode 4 (CTC, Top = OCR1A), no prescaler. Use AVR's crystal oscillator for the clock source.

Solution:

TCCR1A = 0000 0000 WGM11 = 0, WGM10 = 0

TCCR1B = 0000 1001 WGM13 = 0, WGM12 = 1, oscillator clock source, no prescaler

Bit	7	6	5	4	3	2	1	0	TCCR1A
	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	
Read/Write	R/W	R/W	R/W	R/W	W	W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	TCCR1B
	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	
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	

Numerical 2

Find the value for TCCR0 if we want to program Timer0 as a Normal mode counter.
Use an external clock for the clock source and increment on the positive edge.

Solution:

TCCR0 = 0000 0111 Normal, external clock source, no prescaler

Bit	7	6	5	4	3	2	1	0	TCCR0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Numerical 3

Show the status of the Z flag during the execution of the following program:

LDI	R20, 4	;R20 = 4
DEC	R20	;R20 = R20 - 1
DEC	R20	;R20 = R20 - 1
DEC	R20	;R20 = R20 - 1
DEC	R20	;R20 = R20 - 1

- - -

Numerical 4

Show the status of the C, H, and Z flags after the addition of 0x38 and 0x2F in the following instructions:

```
LDI    R16, 0x38  
LDI    R17, 0x2F  
ADD    R16, R17      ;add R17 to R16
```

Solution 4

R16 = 0x67

C=0

H=1

Z=0

Numerical 5

Write a program to subtract two 16-bit numbers: 2762H – 1296H. Assume R26 = (62) and R27 = (27). Place the difference in R26 and R27; R26 should have the lower byte.

Solution 5

```
;R26 = (62)
;R27 = (27)

LDI R28,0x96 ;load the low byte (R28 = 96H)
LDI R29,0x12 ;load the high byte (R29 = 12H)
SUB R26,R28 ;R26 = R26 - R28 = 62 - 96 = CCH
              ;C = borrow = 1, N = 1
SBC R27,R29 ;R27 = R27 - R29 - C
              ;R27 = 27 - 12 - 1 = 14H
```

After the SUB, R26 has = $62H - 96H = CCH$ and the carry flag is set to 1, indicating there is a borrow (notice, $N = 1$). Because $C = 1$, when SBC is executed R27 has $27H - 12H - 1 = 14H$. Therefore, we have $2762H - 1296H = 14CCH$.

When is V(overflow)Flag set?

If the result of an operation on signed numbers is too large for the register , an overflow occurs and programmer will be notified by setting V=1

When is the V flag set?

In 8-bit signed number operations, V is set to 1 if either of the following two conditions occurs:

1. There is a carry from D6 to D7 but no carry out of D7 (C = 0).
2. There is a carry from D7 out (C = 1) but no carry from D6 to D7.

D7	D6	D5	D4	D3	D2	D1	D0
N=0/1							

Numerical 6

Examine the following code, noting the role of the V and N flags:

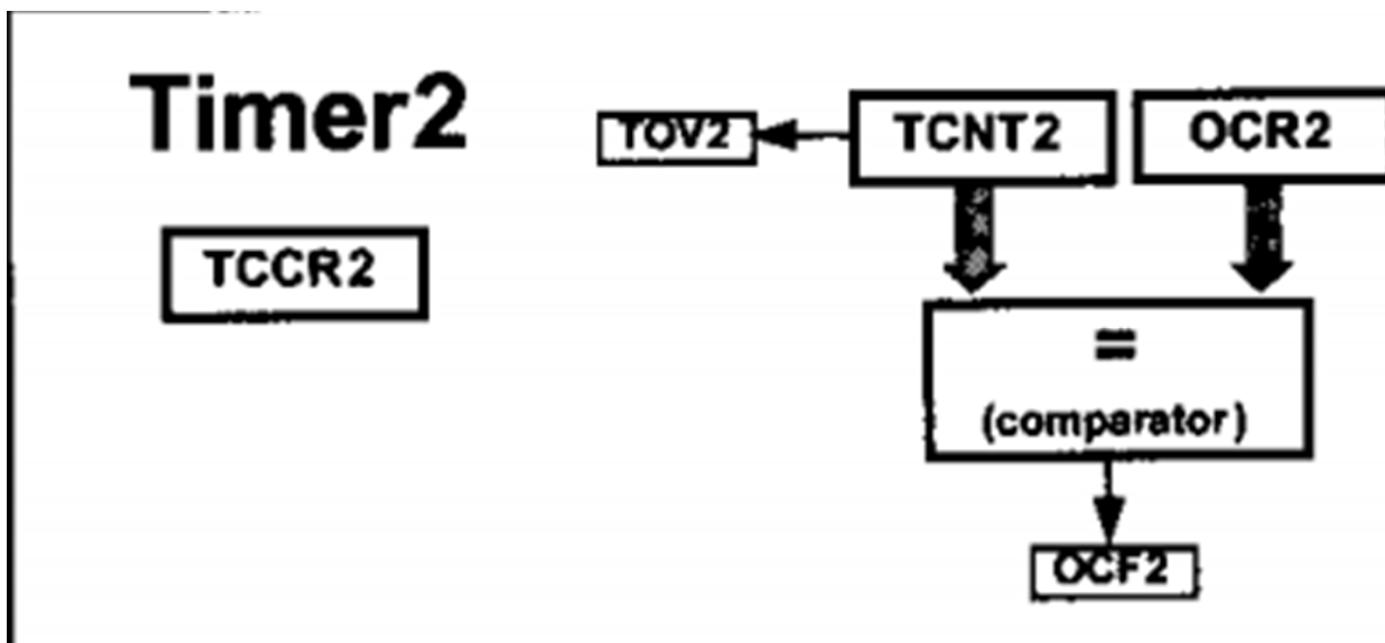
```
LDI R20,0x80 ;R20 = 1000 0000 (80H = -128)
LDI R21,0xFE ;R21 = 1111 1110 (FEH = -2)
ADD R20,R21 ;R20 = (-128) + (-2)
;R20 = 1000000 + 11111110 = 0111 1110,
;N = 0, R0 = 7EH = +126, invalid
```

Solution:

$$\begin{array}{r} \begin{array}{r} -128 \\ + -2 \\ \hline - 130 \end{array} & \begin{array}{l} 1000 \ 0000 \\ \underline{1111 \ 1110} \\ 0111 \ 1110 \end{array} \end{array} \quad \text{N} = 0 \ (\text{positive}) \text{ and } \text{V} = 1$$

According to the CPU, the result is +126, which is wrong, and V = 1 indicates that. Notice that the N flag indicates the sign of the corrupted result, not the sign that the real result should have.

Timer/Counter2

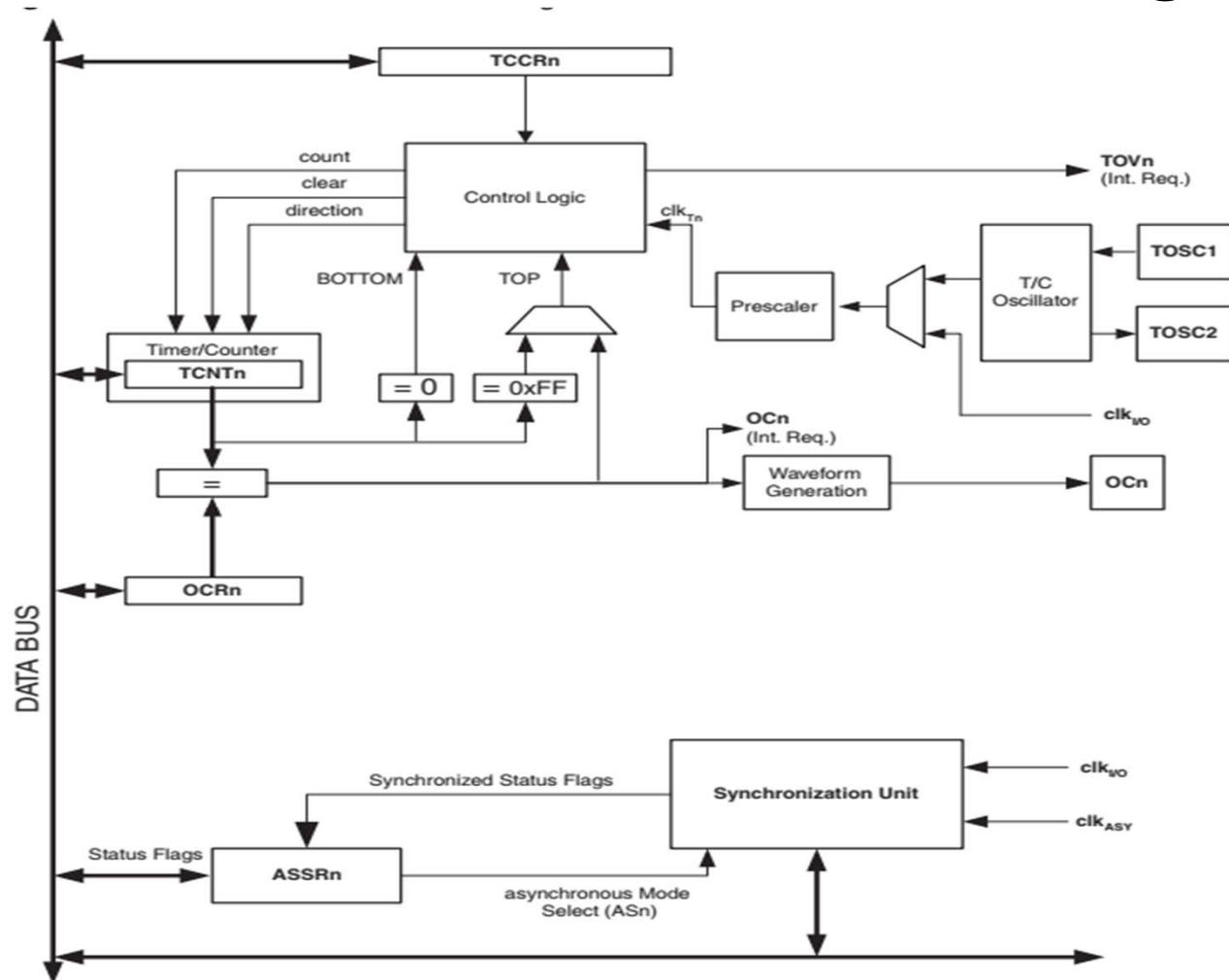


Block Diagram of Timer counter2

Introduction of Timer/counter2

- TIMER2 is an 8bit TIMER and most of the registers of TIMER2 are similar to TIMERO.
- TIMER2 offers a special feature of **Asynchronous operation**. Other Timers don't have this feature.
- Glitch-free, phase Correct Pulse Width Modulator (PWM)** .
- Frequency Generator**
- 10-bit Clock Prescaler**
 - Allows Clocking from External 32kHz Watch Crystal Independent of the I/O Clock
- Single Channel Counter**
- Clear Timer on Compare Match**

8-bit Timer/Counter2 Block Diag.



Timer 2 Registers:

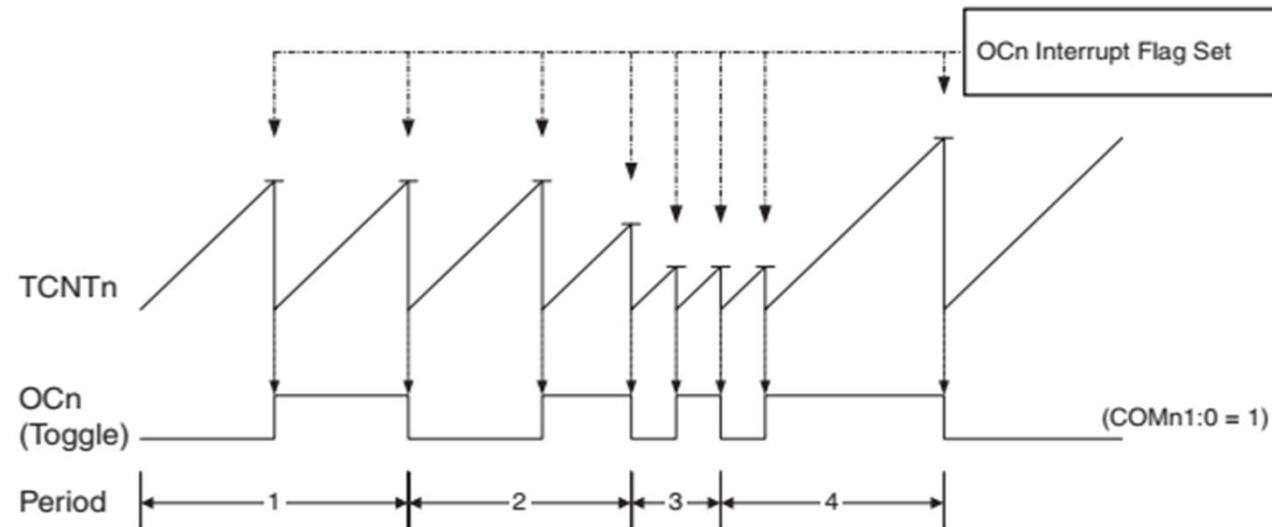
- TCCR2 Register:

Bit	7	6	5	4	3	2	1	0	TCCR2
Read/Write	FOC2 W	WGM20 R/W	COM21 RW	COM20 RW	WGM21 RW	CS22 RW	CS21 RW	CS20 RW	
Initial Value	0	0	0	0	0	0	0	0	

Waveform generation Mode description

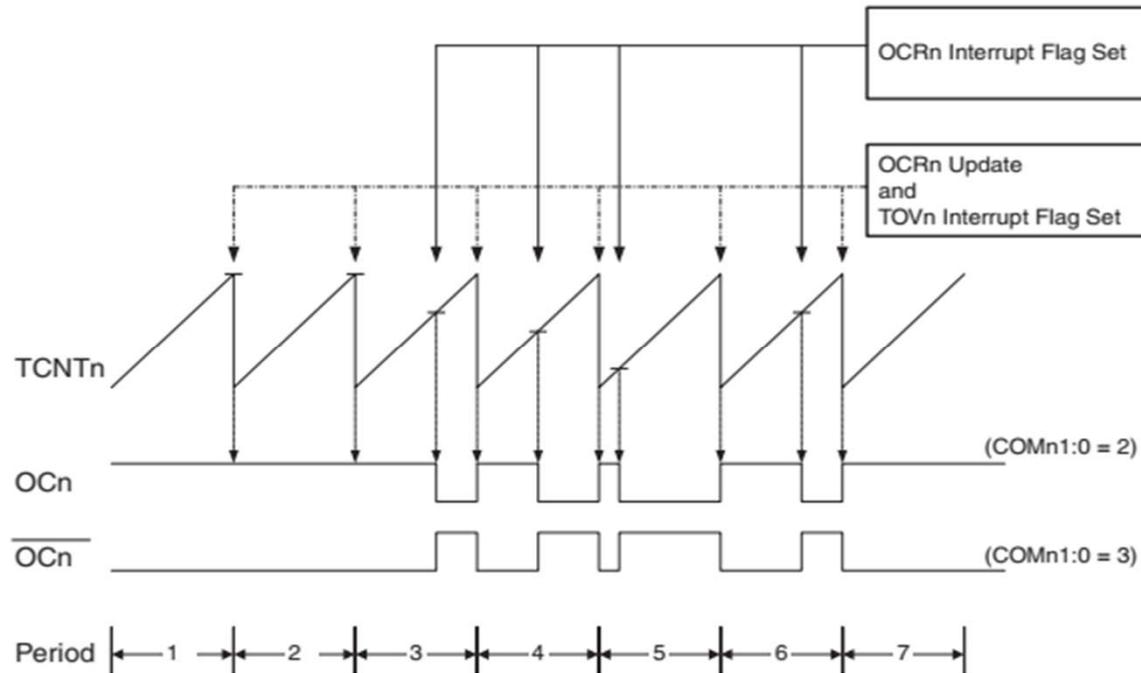
Mode	WGM21 (CTC2)	WGM20 (PWM2)	Timer/Counter Mode of Operation ⁽¹⁾	TOP	Update of OCR2	TOV2 Flag Set
0	0	0	Normal	0xFF	Immediate	MAX
1	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	1	0	CTC	OCR2	Immediate	MAX
3	1	1	Fast PWM	0xFF	BOTTOM	MAX

Compare Output Mode, non PWM (normal or CTC mode)



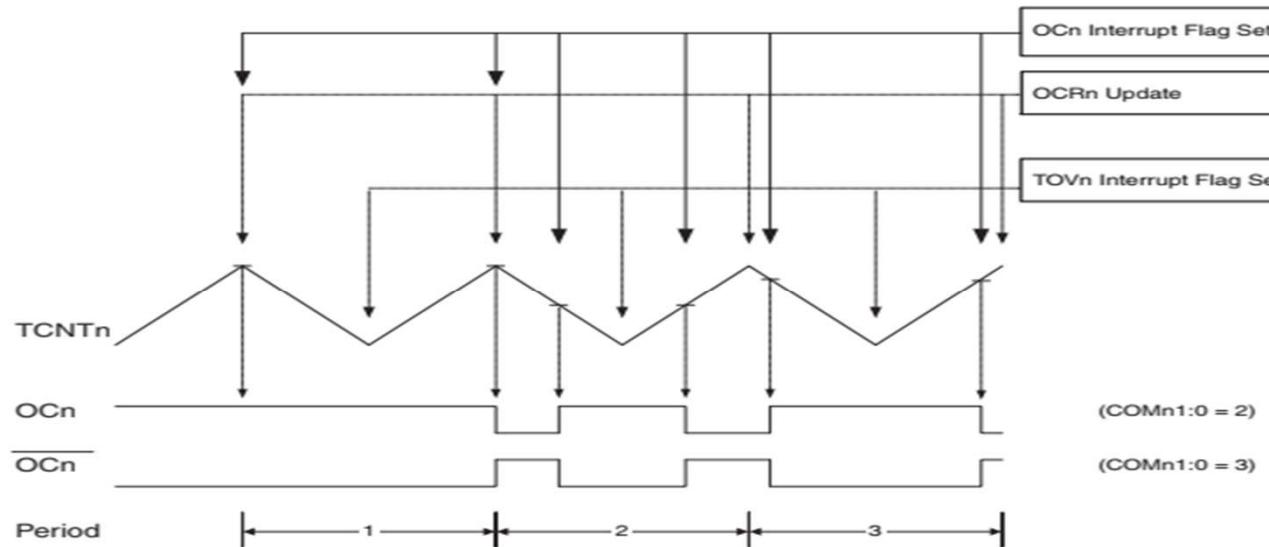
COM21	COM20	Description
0	0	Normal port operation, OC2 disconnected
0	1	Toggle OC2 on Compare Match
1	0	Clear OC2 on Compare Match
1	1	Set OC2 on Compare Match

Compare Output Mode , Fast PWM



COM21	COM20	Description
0	0	Normal port operation, OC2 disconnected
0	1	Reserved
1	0	Clear OC2 on Compare Match, set OC2 at BOTTOM, (non-inverting mode)
1	1	Set OC2 on Compare Match, clear OC2 at BOTTOM, (inverting mode)

Compare Output Mode (Phase correct PWM mode)



COM21	COM20	Description
0	0	Normal port operation, OC2 disconnected
0	1	Reserved
1	0	Clear OC2 on Compare Match when up-counting. Set OC2 on Compare Match when downcounting
1	1	Set OC2 on Compare Match when up-counting. Clear OC2 on Compare Match when downcounting

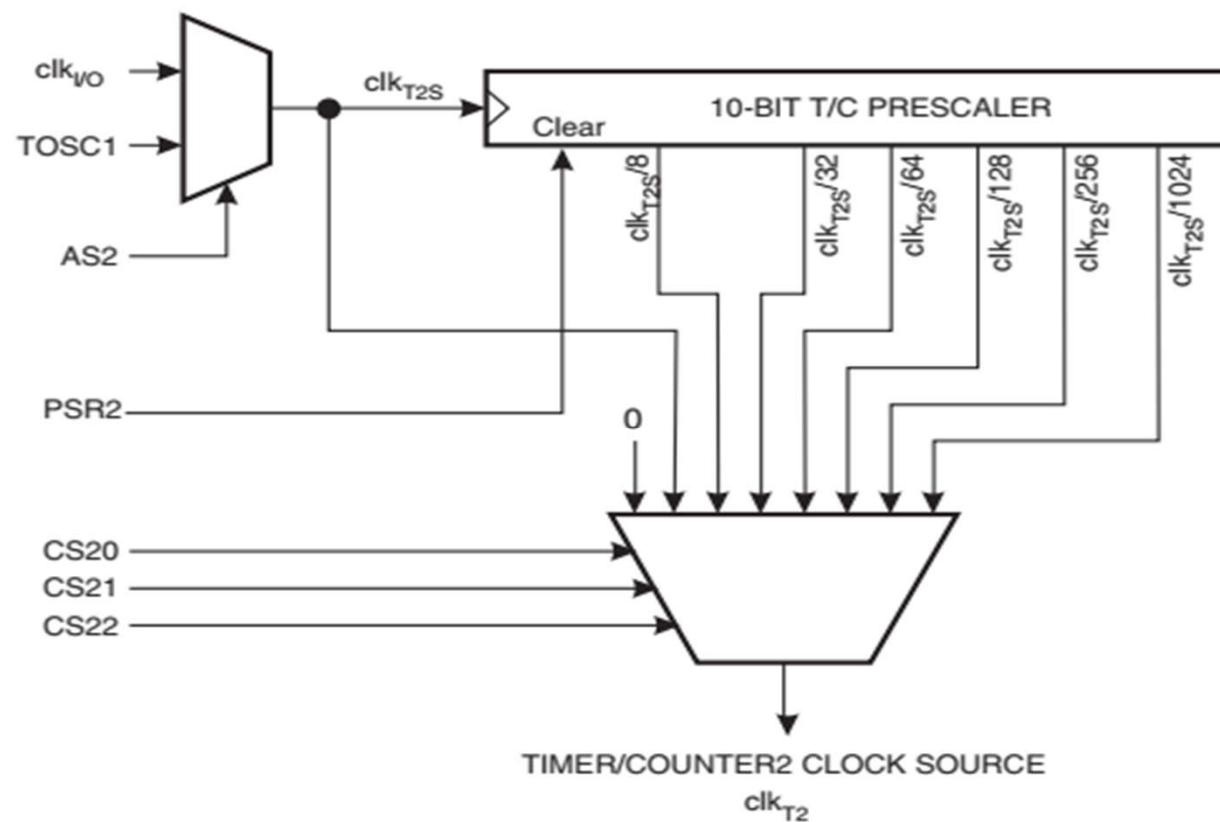
Timer 2 Registers

- Timer2 offers a wide range of prescalers to choose from.
- In TIMERO/1, the prescalers available are 8, 64, 256 and 1024, whereas in **TIMER2, we have 8, 32, 64, 128, 256 and 1024.**

- **Clock select bits : CS22:20**

CS22	CS21	CS20	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	$\text{clk}_{\text{T2S}}/(\text{No prescaling})$
0	1	0	$\text{clk}_{\text{T2S}}/8$ (From prescaler)
0	1	1	$\text{clk}_{\text{T2S}}/32$ (From prescaler)
1	0	0	$\text{clk}_{\text{T2S}}/64$ (From prescaler)
1	0	1	$\text{clk}_{\text{T2S}}/128$ (From prescaler)
1	1	0	$\text{clk}_{\text{T2S}}/256$ (From prescaler)
1	1	1	$\text{clk}_{\text{T2S}}/1024$ (From prescaler)

Timer/Counter2 Prescaler



TCNT2 Register

In the **Timer/Counter Register** – TCNT2, the value of the timer is stored. Since TIMER2 is an 8-bit timer, this register is 8 bits wide.

Bit	7	6	5	4	3	2	1	0	TCNT2
Read/Write	RW								
Initial Value	0	0	0	0	0	0	0	0	

TIMSK Register

- **Timer/Counter Interrupt Mask:**
- It is a register common to all the timers.

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	-	TOIE0	TIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

TIFR Register

The **Timer/Counter Interrupt Flag Register** – TIFR is as follows. It is a register common to all the timers.

Bit	7	6	5	4	3	2	1	0	TIFR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Asynchronous Operation of the Timer/Counter2

- Asynchronous Status Register – ASSR

Bit	7	6	5	4	3	2	1	0	ASSR
Read/Write	-	-	-	-	AS2	TCN2UB	OCR2UB	TCR2UB	
Initial Value	0	0	0	0	0	0	0	0	

Bit 3 – AS2: Asynchronous Timer/Counter2

When AS2 is written to zero, Timer/Counter 2 is clocked from the I/O clock, $\text{clk}_{\text{I/O}}$. When AS2 is written to one, Timer/Counter 2 is clocked from a crystal Oscillator connected to the **Timer Oscillator 1 (TOSC1)** pin.

- **Bit 2 – TCN2UB: Timer/Counter2 Update Busy**

When Timer/Counter2 operates asynchronously and TCNT2 is written, this bit becomes set. When TCNT2 has been updated from the temporary storage register, this bit is cleared by hardware. A logical zero in this bit indicates that TCNT2 is ready to be updated with a new value

- **Bit 1 – OCR2UB: Output Compare Register2 Update Busy**

When Timer/Counter2 operates asynchronously and OCR2 is written, this bit becomes set. When OCR2 has been updated from the temporary storage register, this bit is cleared by hardware. A logical zero in this bit indicates that OCR2 is ready to be updated with a new value.

- **Bit 0 – TCR2UB: Timer/Counter Control Register2 Update Busy**

When Timer/Counter2 operates asynchronously and TCCR2 is written, this bit becomes set. When TCCR2 has been updated from the temporary storage register, this bit is cleared by hardware. A logical zero in this bit indicates that TCCR2 is ready to be updated with a new value.

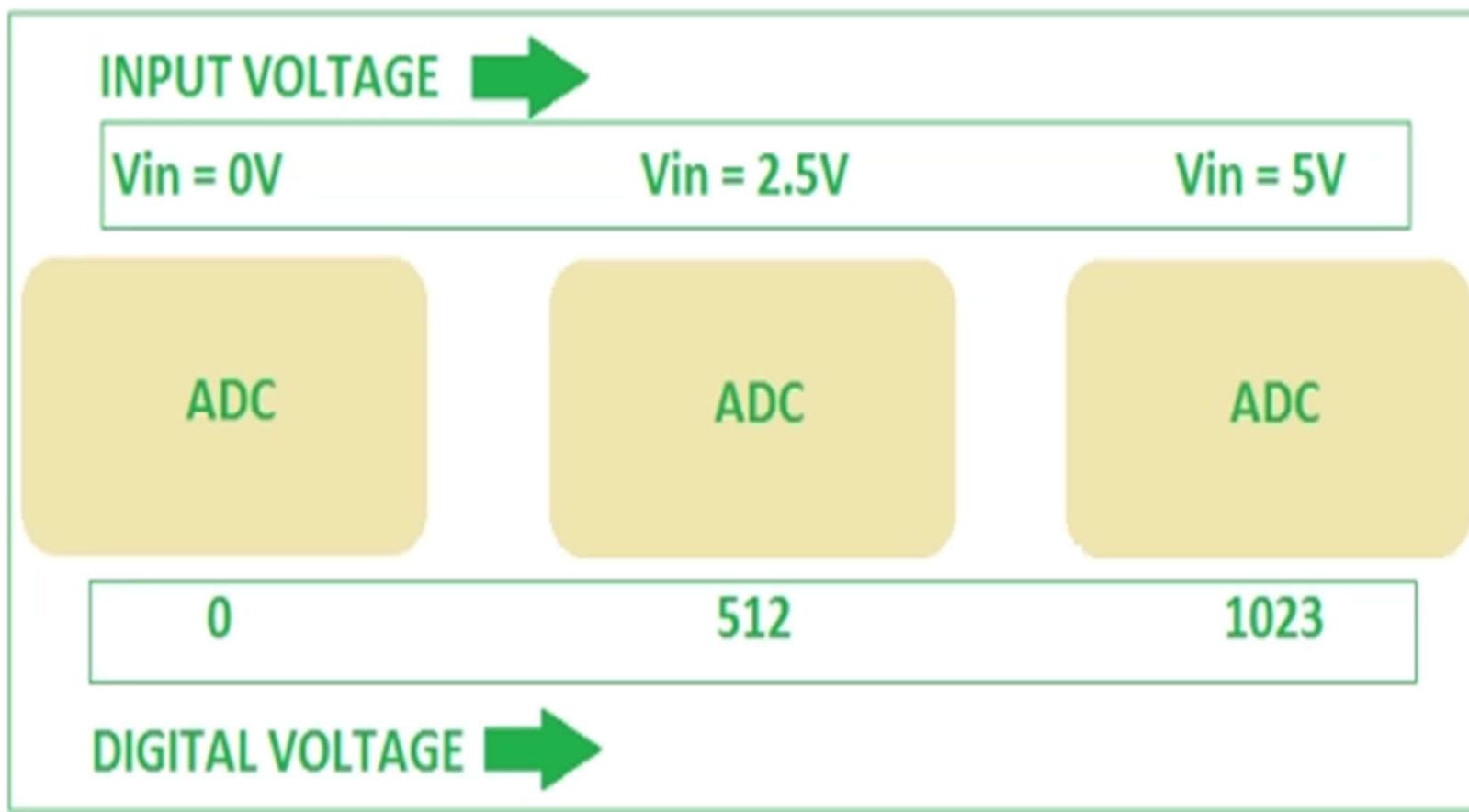
Atmega 8, Analog to Digital Converter (ADC)

- Conversion of external continuous information (analog information) into integer (digital) values. This type of conversion is carried out by Analog to Digital Converter (ADC).
- AVR microcontrollers has inbuilt ADC facility to convert analog voltage into an integer. AVR convert it into 10-bit number of range 0 to 1023.

ADC (continued)...

- There is **Analog Reference (Aref) Voltage** also, which will be considered equivalent to 1023.
- Any voltage value less than this Aref will have less number than 1023.
- The input range is **0-Aref** and **digital output is 0-1023**. Here **0V** will be equal to **0**, and **Aref/2** will be equal to **512** and so on.

ADC (continued)...

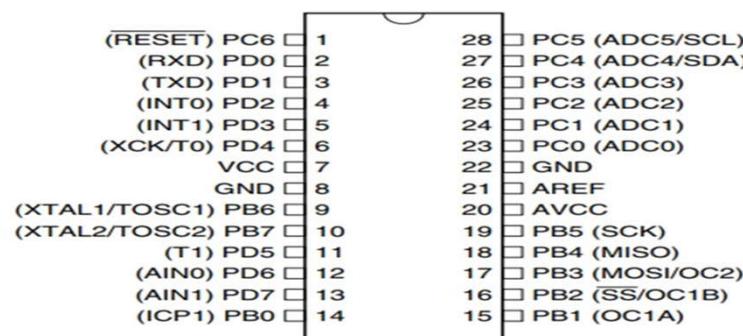


ADC in Atmega8

- The ADC is multiplexed with Port C
- ADC can be operated in **single conversion mode** and **free running mode**.
- In **single conversion mode**: The ADC does a single conversion and stops.
- In **free running mode**: The ADC is continuously converting, i.e it does a conversion and then start the next conversion instantly after that.

ADC Prescaler

- The ADC needs a clock pulse for the job, and for this the system clock is divided by a number (2, 4, 16, 32, 64 and 128) to get the lesser frequency (ADC requires a frequency between **50KHz** to **200KHz**)
- **ADC Channels**: The ADC in Atmega8 has **6 channels**, allows you to take samples from 6 different pins



ADC Registers

- **ADC Registers:** Register provides the communication link between CPU and the ADC. You can configure the ADC according to your need using these registers. The ADC has 3 registers only:
 1. **ADC Multiplexer Selection Register – ADMUX:** For selecting the reference voltage and the input channel

Bit	7	6	5	4	3	2	1	0	ADMUX
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Table 74. Voltage Reference Selections for ADC

REFS1	REFS0	Voltage Reference Selection
0	0	AREF, Internal V_{ref} turned off
0	1	AV_{CC} with external capacitor at AREF pin
1	0	Reserved
1	1	Internal 2.56V Voltage Reference with external capacitor at AREF pin

Table 75. Input Channel Selections

MUX3..0	Single Ended Input
0000	ADC0
0001	ADC1
0010	ADC2
0011	ADC3
0100	ADC4
0101	ADC5

MUX3..0	Single Ended Input
0110	ADC6
0111	ADC7
1000	
1001	
1010	
1011	
1100	
1101	
1110	1.30V (V_{BG})
1111	0V (GND)

2. ADC Control and Status Register A – ADCSRA: It has the status of ADC and is also used to control it

Bit	7	6	5	4	3	2	1	0	ADCSRA
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

➤ **Bit 7 – ADEN: ADC Enable**

Writing this bit to one enables the ADC. By writing it to zero, the ADC is turned off.

➤ **Bit 6 – ADSC: ADC Start Conversion**

In Single Conversion mode, write this bit to one to start each conversion.

In Free Running mode, write this bit to one to start the first conversion. The first conversion after ADSC has been written after the ADC has been enabled, When the conversion is complete, it returns to zero. Writing zero to this bit has no effect.

➤ **Bit 5 – ADFR: ADC Free Running Select**

When this bit is set (one) ,the ADC operates in Free Running mode. In this mode, the ADC samples and updates the Data Registers continuously. Clearing this bit (zero) will terminate Free Running mode.

➤ **Bit 4 – ADIF: ADC Interrupt Flag**

This bit is set when an ADC conversion completes and the Data Registers are updated. The ADC Conversion Complete Interrupt is executed if the ADIE bit and the I-bit in SREG are set.

ADIF is cleared by hardware when executing the corresponding interrupt Handling Vector.

➤ **Bit 3 – ADIE: ADC Interrupt Enable**

When this bit is written to one and the I-bit in SREG is set, the ADC Conversion Complete Interrupt is activated

- **Bits 2:0 – ADPS2:0: ADC Prescaler Select Bits**

These bits determine the division factor between the **XTAL frequency** and the **input clock** to the ADC.

Table 76. ADC Prescaler Selections

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

3. The ADC Data Register – ADCL and ADCH

Final result of the conversion is stored here.

ADLAR = 0

Bit	15	14	13	12	11	10	9	8	ADCH	ADCL
	-	-	-	-	-	-	ADC9	ADC8		
	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0		
	7	6	5	4	3	2	1	0		
ReadWrite	R	R	R	R	R	R	R	R		
	R	R	R	R	R	R	R	R		
Initial Value	0	0	0	0	0	0	0	0		
	0	0	0	0	0	0	0	0		

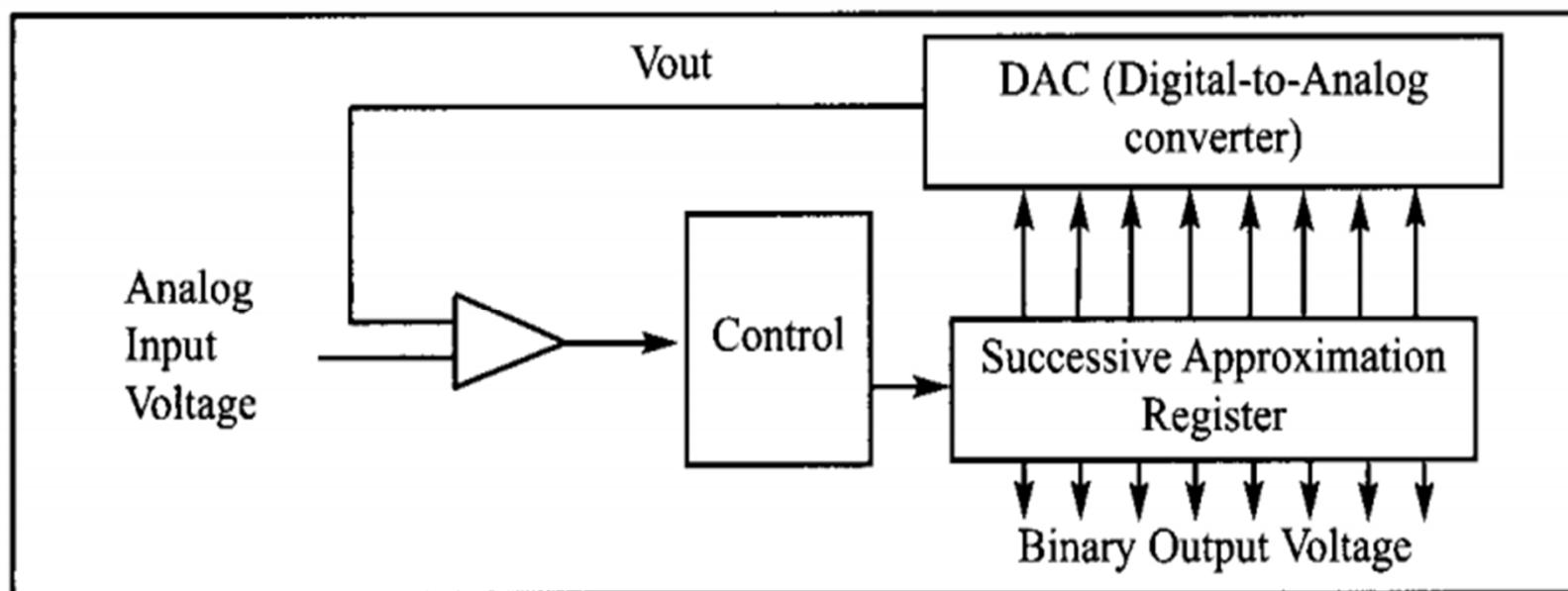
ADLAR = 1

Bit	15	14	13	12	11	10	9	8	ADCH	ADCL
	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2		
	ADC1	ADC0	-	-	-	-	-	-		
	7	6	5	4	3	2	1	0		
ReadWrite	R	R	R	R	R	R	R	R		
	R	R	R	R	R	R	R	R		
Initial Value	0	0	0	0	0	0	0	0		
	0	0	0	0	0	0	0	0		

- **A_{VCC}**: This pin supplies power to ADC. AVCC must not differ more than $\pm 0.3V$ from Vcc
- **A_{REF}**: Another pin which can optionally be used as an external voltage reference pin.
- **Internal 2.56 V** for selecting Vref
- **Voltage Resolution**: This is the smallest voltage increment which can be measured. For 10 bit ADC, there can be 1024 different voltages (for an 8 bit ADC, there can be 256 different voltages)

Successive approximation ADC Technique

- The ADC converts an analog input voltage to a 10-bit digital value through **successive approximation**.



- It works by comparing input voltage with half of the reference voltage generated internally. The comparison continues by dividing the voltage further down and updating each bit in ADC register by 1 if input voltage is high, 0 otherwise. This process lasts 10 times (for 10 bit ADC) and generates resulting binary output.

Major Characteristics of ADC

- **Resolution:** The ADC has n-bit resolution. n= 8, 10,12, 16, 24 bit. Higher resolution ADCs provide smaller step size.

Step size: Smallest change that can be detected by an ADC.

Table 13-1: Resolution versus Step Size for ADC ($V_{ref} = 5 \text{ V}$)

n-bit	Number of steps	Step size (mV)
8	256	$5/256 = 19.53$
10	1024	$5/1024 = 4.88$
12	4096	$5/4096 = 1.2$
16	65,536	$5/65,536 = 0.076$

Notes: $V_{CC} = 5 \text{ V}$

- **Conversion Time:** It is defined as time it takes for the ADC to convert analog input to digital number.
- **It depends upon :**
 - Clock source
 - Technology used for fabrication of ADC.
 - Method of conversion

V_{ref}: Input voltage used for reference voltage. If analog input range needs to be 0-4V, V_{ref} is connected to 4V.

Step size = 4/1024 = 3.90mV (n=10)

Table 13-3: V_{ref} Relation to V_{in} Range for an 10-bit ADC

V _{ref} (V)	V _{in} (V)	Step Size (mV)
5.00	0 to 5	5/1024 = 4.88
4.096	0 to 4.096	4.096/1024 = 4
3.0	0 to 3	3/1024 = 2.93
2.56	0 to 2.56	2.56/1024 = 2.5
2.048	0 to 2.048	2.048/1024 = 2
1.28	0 to 1.28	1/1024 = 1.25
1.024	0 to 1.024	1.024/1024 = 1

- **Digital data output:** For 10 bit ADC , digital data output is D0-D9.
- Data can be brought out of ADC in serial or parallel form.

$$D_{out} = \frac{V_{in}}{\text{step size}}$$

Numerical on ADC

For an 8-bit ADC, we have $V_{\text{ref}} = 2.56$ V. Calculate the D0–D7 output if the analog input is: (a) 1.7 V, and (b) 2.1 V.

Solution

Because the step size is $2.56/256 = 10 \text{ mV}$, we have the following:

- (a) $D_{\text{out}} = 1.7 \text{ V}/10 \text{ mV} = 170$ in decimal, which gives us 10101010 in binary for D7-D0.
- (b) $D_{\text{out}} = 2.1 \text{ V}/10 \text{ mV} = 210$ in decimal, which gives us 11010010 in binary for D7-D0.