AVR - HW5

임베디드스쿨1기
Lv1과정
2020. 10. 13
손표훈

1. Timer/Counter0

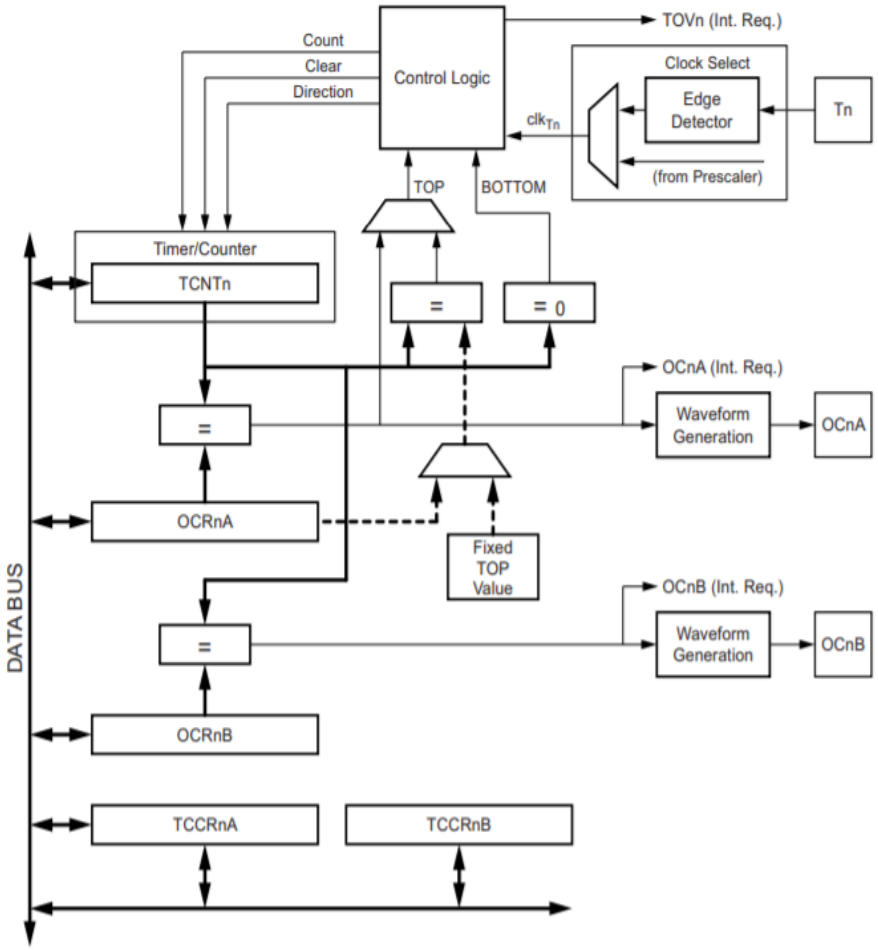
(1) ATmega328P의 Timer/Counter0의 특징

8-bit Timer/Counter0 with PWM

Features

- Two independent output compare units
- Double buffered output compare registers
- Clear timer on compare match (auto reload)
- Glitch free, phase correct pulse width modulator (PWM)
- Variable PWM period
- Frequency generator
- Three independent interrupt sources (TOV0, OCF0A, and OCF0B)

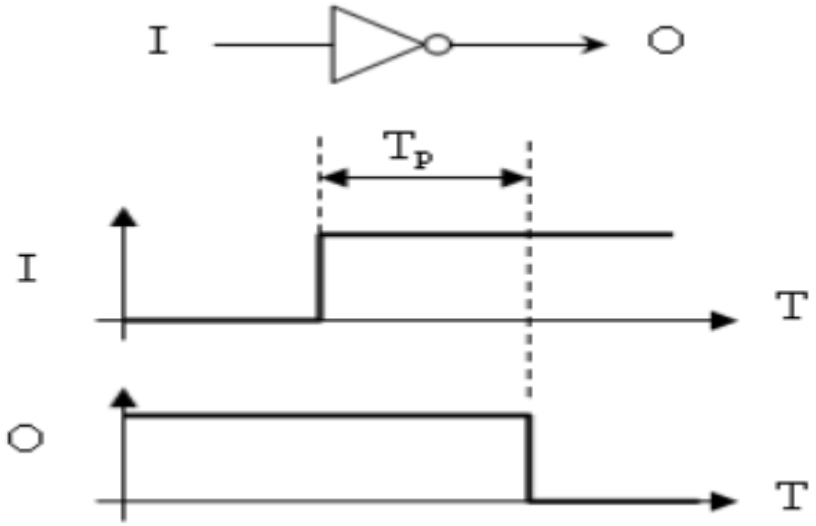
Figure 14-1. 8-bit Timer/Counter Block Diagram



1. Timer/Counter0

(1) ATmega328P의 Timer/Counter0의 특징

- Glitch Free? Glitch?
- 우선 Propgation Delay가 뭔지 알아야한다..



- 모든 이상적인 소자는 입력대비 출력이 동시간대에 나와야 한다..

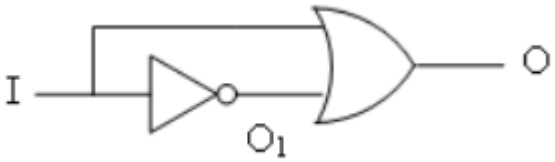
- 하지만 소자 내부 기생 커패시턴스 및 패턴 저항으로 인해 지연이 발생한다.
  (lowpass filter 형성됨..)

- CMOS IC의 경우 FET로 구성되어 있으며, Gtate – Source간 커패시터로 인해 발생

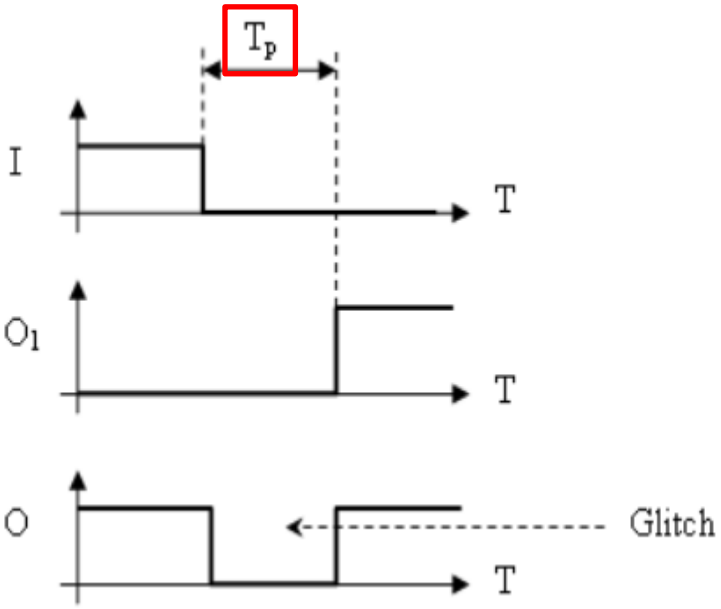
1. Timer/Counter0

(1) ATmega328P의 Timer/Counter0의 특징

- Glitch? 여러 개의 digital IC들로 구성된 회로에서 Propagation Delay로 인해 발생한 입력대비 출력 지연이 의도치 않은 동작을 발생시키는 현상



propagation delay

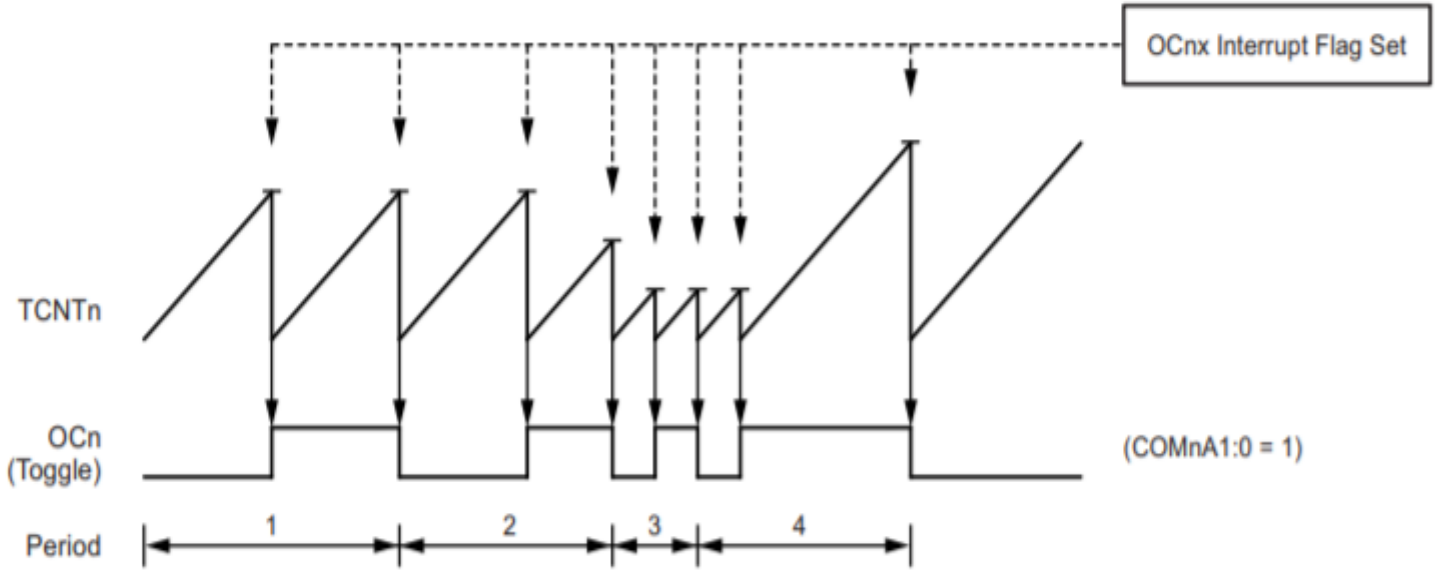


- 입력이 'H'이면 이론대로라면 'H'가 계속 출력되어야 하지만
- Propagation Delay로 인해 의도치 않은 'L'가 출력되는 구간이 발생한다.

1. Timer/Counter0

(1) CTC mode(Clear Timer on Compare Match)의 특징

Figure 14-5. CTC Mode, Timing Diagram

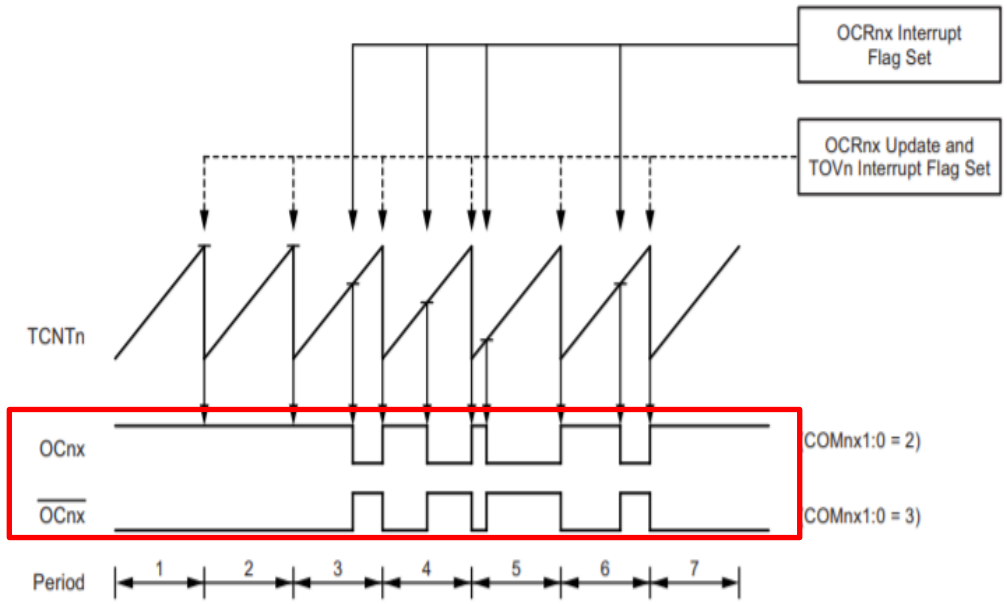


- TCNT 레지스터의 값에 따라 주파수, 듀티비가 가변한다.
- 시계, 멜로디 발생 등에 활용할 수 있다.

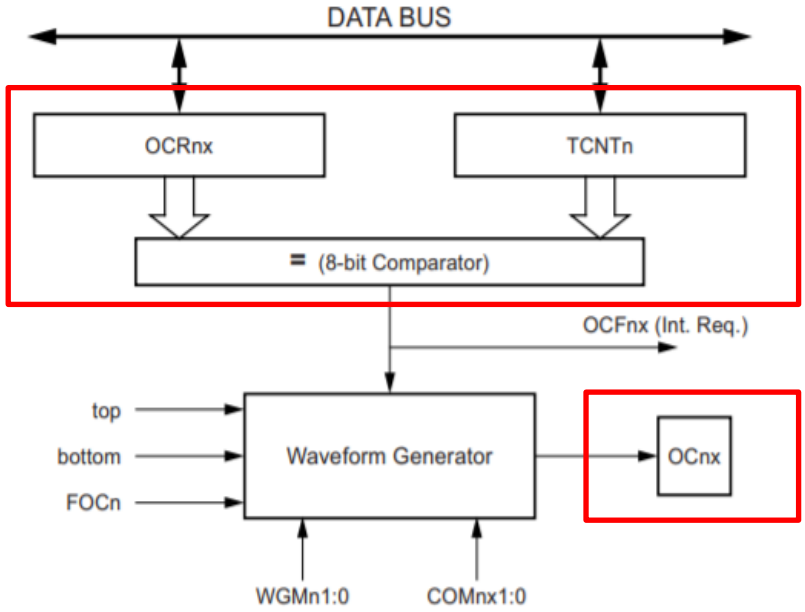
1. Timer/Counter0

(2) FastPWM mode의 특징

Figure 14-6. Fast PWM Mode, Timing Diagram



Output Compare Unit, Block Diagram

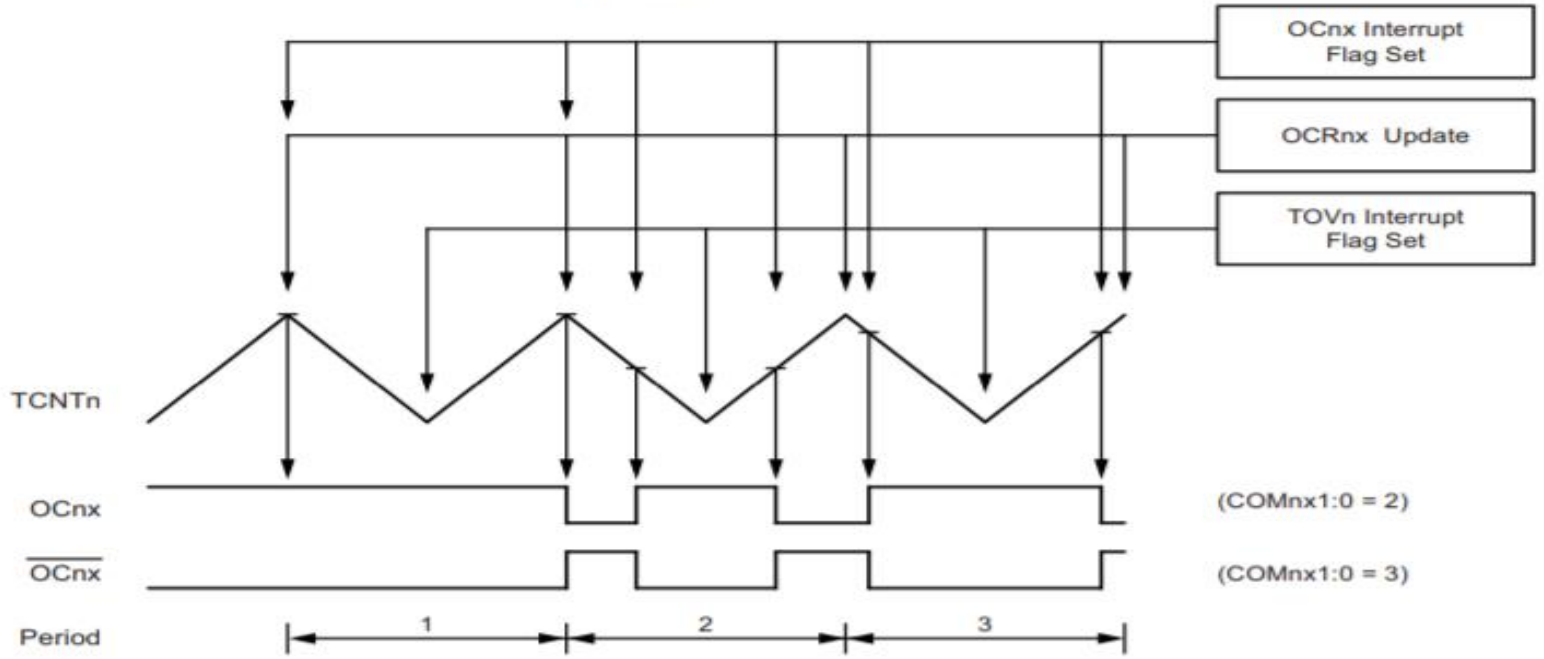


- TCNT레지스터의 값에 따라 주파수만 가변이 된다.
- OCR레지스터에 값을 설정하면, TCNT레지스터의 값과 비교하여 듀티비가 조절된다.

1. Timer/Counter0

(3) Phase Correct PWM mode의 특징

Figure 14-7. Phase Correct PWM Mode, Timing Diagram



- TCNT레지스터의 값에 따라 주파수만 가변이 된다.
- OCR레지스터에 값을 설정하면, TCNT레지스터의 값과 비교하여 듀티비가 조절된다.
  (여기까지 Fast PWM과 같음)
- TCNT값이 증가와 감소 동작을 함
- PWM 파형이 TCNT기준 파형의 중간에서 발생한다.

1. Timer/Counter0

(4) Phase Correct PWM vs Fast PWM mode

Figure 14-7. Phase Correct PWM Mode, Timing Diagram

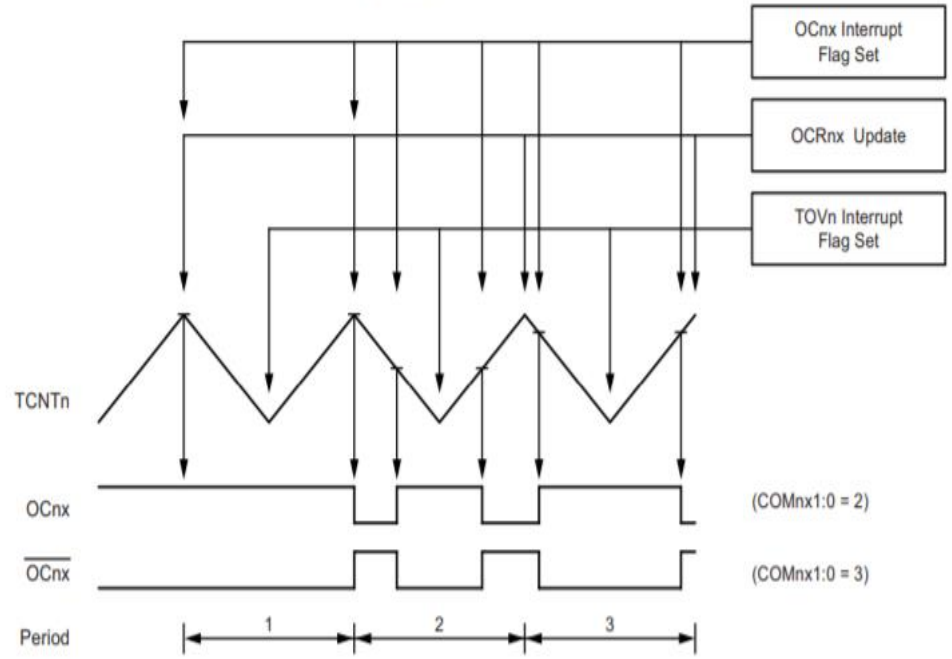
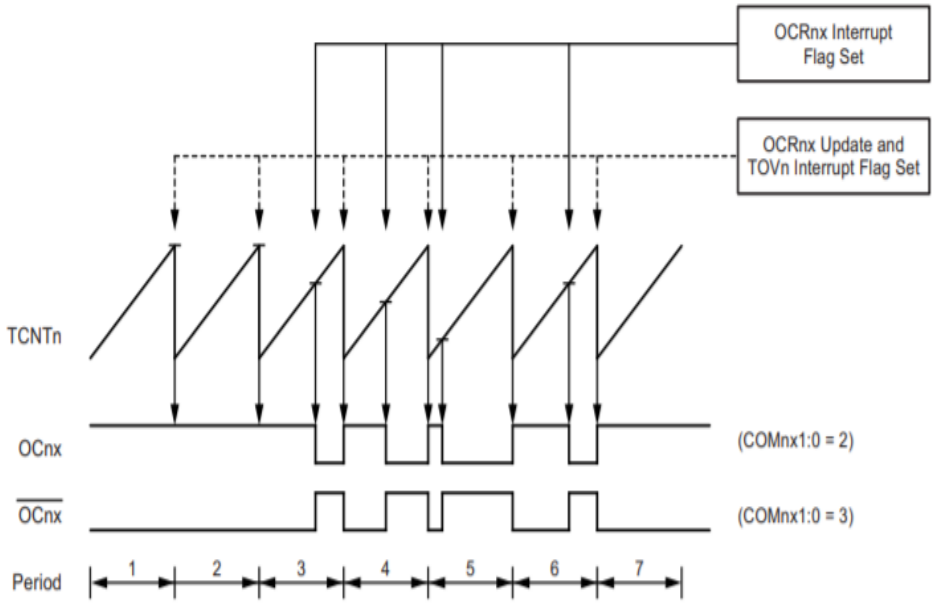


Figure 14-6. Fast PWM Mode, Timing Diagram



- Phase Correct PWM의 경우 분해능이 Fast PWM보다 2배 높지만 동일 클럭, 동일한 분주비일 경우 Phase Correct PWM의 주파수가 Fast PWM에 비해 2배 느리다.
  (Phase Correct의 경우 삼각파형태로 기준주파수를 가지고, Fast PWM은 톱니파형태)

1. Timer/Counter0

(4) Phase Correct PWM vs Fast PWM mode

Figure 14-7. Phase Correct PWM Mode, Timing Diagram

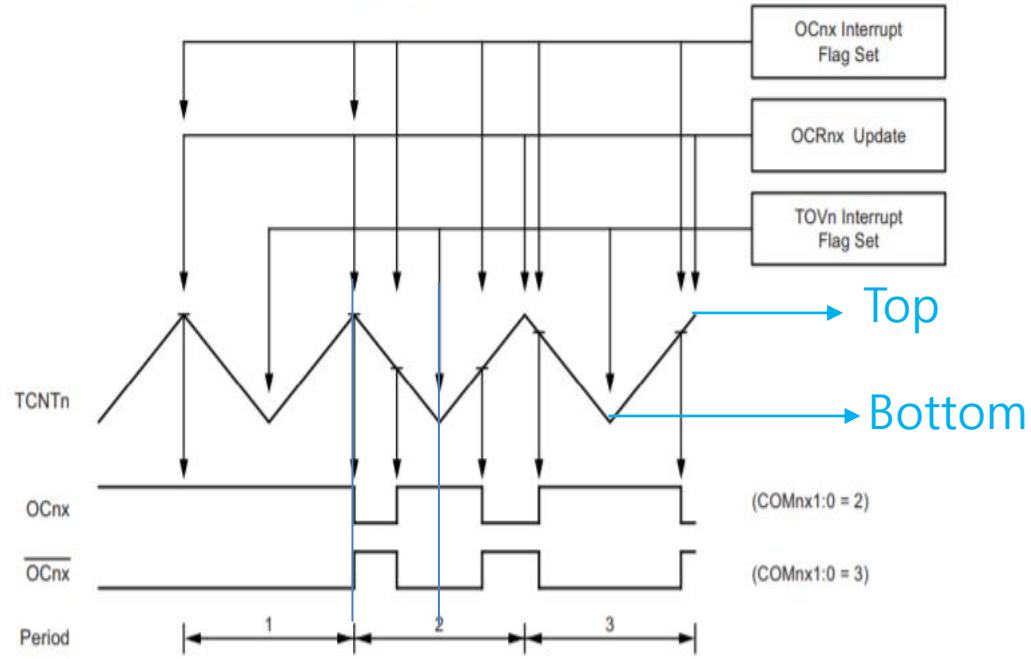
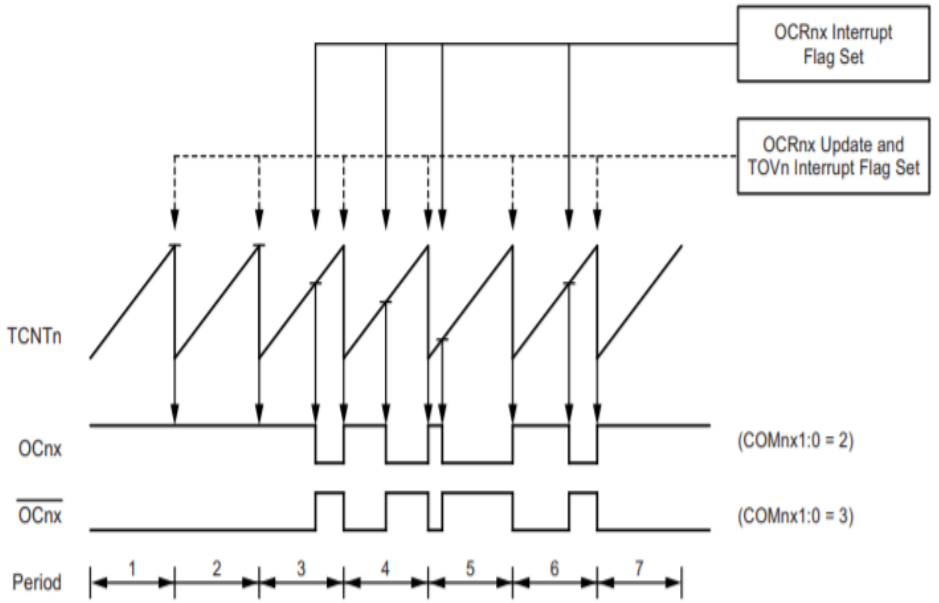


Figure 14-6. Fast PWM Mode, Timing Diagram



- Phase Correct의 경우 OCR값을 Bottom(Duty = 0%)로 설정하면 Ocn핀의 PWM 출력이 'L'로 되는 반면

- Fast PWM의 경우 Bottom값 설정시 narrow pulse가 발생한다.
  Fast PWM의 경우 Counter값이 Top이 되면 Ocn핀의 출력이 'H'가 되니까..
  따라서 0% 설정시 타이머 카운터를 disable하거나 분주를 0으로 설정해야한다.

1. Timer/Counter0

(4) Phase Correct PWM vs Fast PWM mode

Figure 14-7. Phase Correct PWM Mode, Timing Diagram

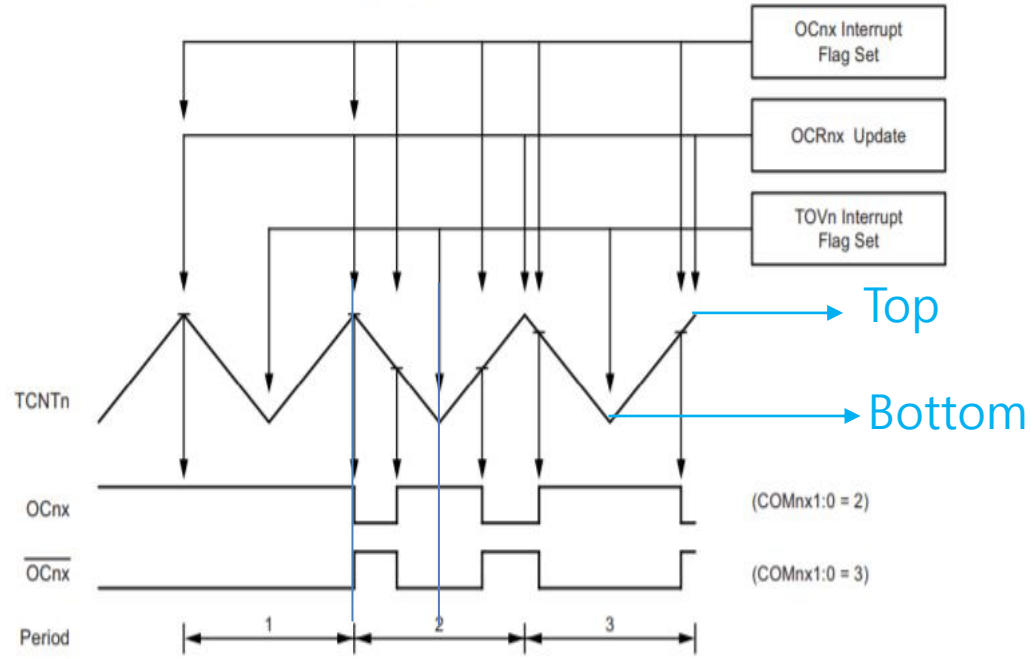
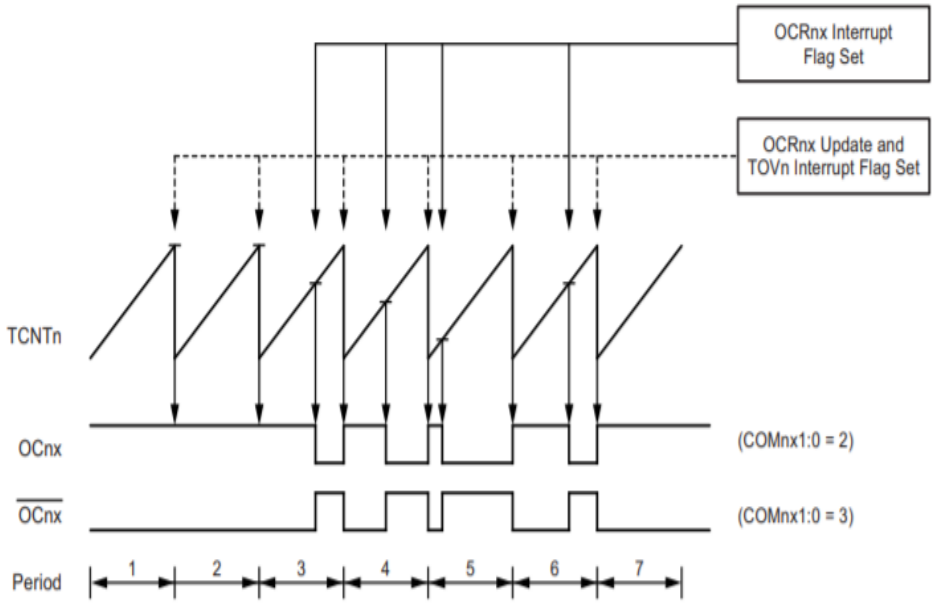


Figure 14-6. Fast PWM Mode, Timing Diagram



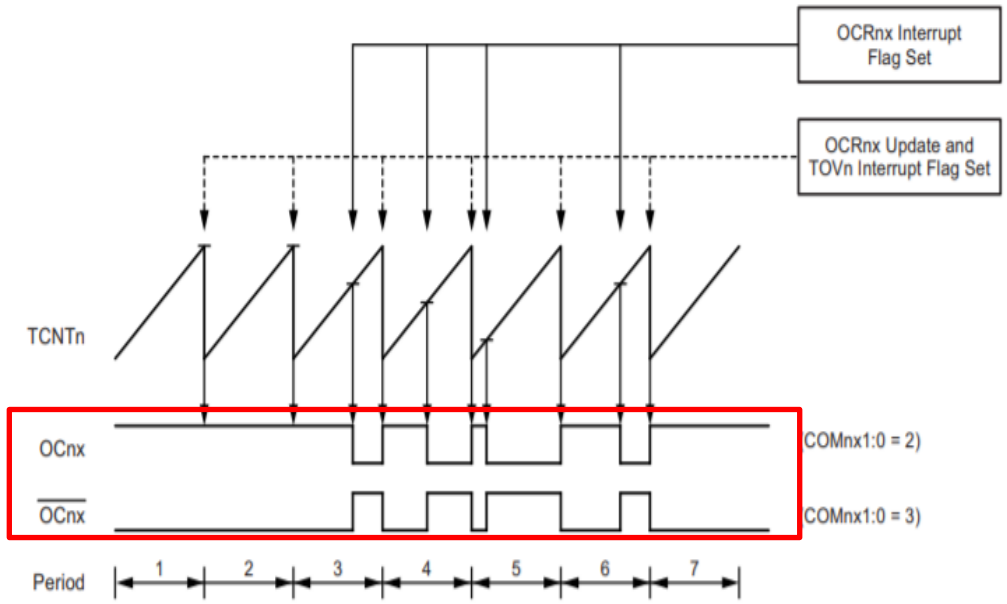
- Phase Correct의 경우 OCR값을 Bottom(Duty = 0%)로 설정하면 Ocn핀의 PWM 출력이 'L'로 되는 반면

- Fast PWM의 경우 Bottom값 설정시 narrow pulse가 발생한다.
  Fast PWM의 경우 Counter값이 Top이 되면 Ocn핀의 출력이 'H'가 되니까..
  따라서 0% 설정시 타이머 카운터를 disable하거나 분주를 0으로 설정해야한다.

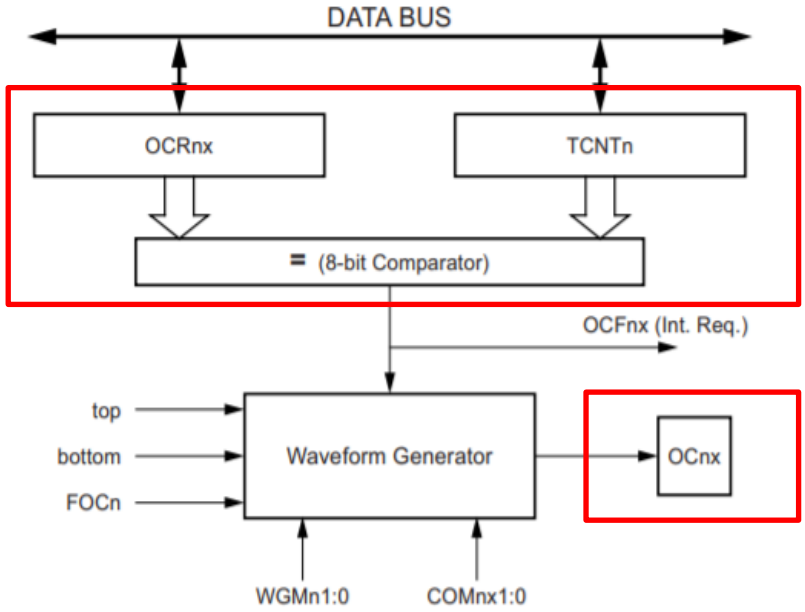
1. Timer/Counter0

(2) FastPWM mode의 특징

Figure 14-6. Fast PWM Mode, Timing Diagram



Output Compare Unit, Block Diagram



- TCNT레지스터의 값에 따라 주파수만 가변이 된다.
- OCR레지스터에 값을 설정하면, TCNT레지스터의 값과 비교하여 듀티비가 조절된다.

2. Duty를 UART로 전송

```
void FastPwmInit(void)
{
    sbi(TCCR0A, WGM01);
    sbi(TCCR0A, WGM00);
    sbi(TCCR0A, COM0A1);
    //sbi(TCCR0A, COM0A0);
    TCCR0B = (1 << CS02) | (1 << CS00); // Fsysclk/1024
    sbi(TIMSK0, TOIE0);
    //TCNT0 = 0x00; //TCNT는 시작값임!
    OCR0A = 127;
    PORTD = 0x00;
    DDRD = 0xFF;

}
```

FAST PWM
NON-Inverting Mode

8bit 0~255의 중간 값인 127로 OCR설정
Duty 50%

```
void UART_INIT(void)
{
    sbi(UCSR0A, U2X0);

    UBRR0H = 0x00;
    UBRR0L = 207;

    UCSR0C |= 0x06;

    sbi(UCSR0B, RXEN0);
    sbi(UCSR0B, TXEN0);
}
```

```
int getDutyRatio(int data)
{
    float duty;
    duty = data;
    duty = (duty+1) / 256;
    duty = duty*100;
    return (int)duty;
}
```

```
Duty = 50%
Duty = 50%
Duty = 50%
Duty = 50%
Duty = 50%
Duty = 50%
Duty = 50%
Duty = 50%
Duty = 50%
Duty = 50%
Duty = 50%
Duty = 50%
Duty = 50%
Duty = 50%
```

설정된 OCR값을 Duty로 변환
127+1 = 128
128 / 256 = 0.5
scale값으로 100을 곱하면 듀티가됨!