AVR - Chapter 4

뉴테크놀로지 컴패니
대표 류 대 우
davidryu@newtc.co.kr

[타이머 / 카운터]

- 입력으로 들어오는 펄스를 셈하는 장치.
- 입력으로 들어오는 펄스가 어디에 존재하느냐에 따라서 타이머와 카운터로 나눔.
- AVR ATmega128 : 범용 타이머/카운터 4개 존재
  - Timer/Counter0(8비트)
  - Timer/Counter1(16비트)
  - Timer/Counter2(8비트)
  - Timer/Counter3(16비트)

[타이머 / 카운터]

- **타이머와 카운터의 차이점**
  - **타이머**
    - 내부클럭(빠름/분주가능: 범위 내에서 클럭 선택가능) – 동기모드
    - 타이머는 MCU의 내부클럭(clkI/O>분주기>clkT)을 이용
    - 일정시간 간격의 펄스를 만들어 내거나 일정시간 경과 후에 인터럽트를 발생시키는 기능
  - **카운터**
    - 외부클럭(느림/분주불가능: 외부클럭 그대로 사용) – 비동기모드
    - 카운터는 외부 핀(TOSC1, TOSC2, T1, T2, T3)을 통해서 들어오는 펄스를 계수(Edge Detector)하여 Event Counter로서 동작 (펄스=사건, 카운터 값=사건의 횟수)

[타이머 / 카운터]

- **타이머 카운터 레지스터**
  - 타이머/카운터 제어 레지스터(TCCRn)
  - 타이머/카운터 레지스터(TCNTn)
  - 출력 비교 레지스터(OCRn)
  - 인터럽트 관련
    - 타이머/카운터 인터럽트 플래그 레지스터(TIFR)
    - 타이머/카운터 인터럽트 마스크 레지스터 (TIMSK)

[타이머 / 카운터]

- 타이머를 사용하기 위해서는 타이머에서 사용하는 클럭에 대해서 설정을 해야 함.
- 프리스케일러(Prescaler) 값으로 조절할 수 있음.
- 프리스케일러 값은 각 타이머의 컨트롤 레지스터(TCCRn)에서 설정
- 각 타이머 레지스터(TCNTn)에 얼마마다 한번 씩 인터럽트를 걸게 할 것인지와 관련된 값을 써주면 됨.
- 인터럽트를 사용해야 하므로 타이머 인터럽트 관련 레지스터들을 설정
- 타이머 인터럽트에서는 TIMSK 레지스터만 설정

[프리스케일러]

- 전치 분주기라는 말로 설명을 자주 하고 있는 이것은 8개의 스케일을 가지고 있다. (시스템 클럭/4)을 프리스케일러가 분주 하고 나온 클럭을 타이머 클럭으로 사용 하는 것이다.
- 분주비가 1:4 일경우 시스템클럭이 4번 들어 올때 스리스케일러를 통과한 클럭은 1번 클럭이 발생하고 타이머는 1번의 클럭으로 인식 하는 것이다.
- 즉, 시스템 클럭이 16MHz 이라고 한다면 타이머 클럭은 4MHz 가 되는 것이다.

CS02	CS01	CS00	기 능 설 명
0	0	0	정지, 타이머/카운터0 기능 정지
0	0	1	CLK
0	1	0	CLK/8
0	1	1	CLK/64
1	0	0	CLK/256
1	0	1	CLK/1024
1	1	0	외부 핀 T0의 하강 에지 동작
1	1	1	외부 핀 T0의 상승 에지 동작

[ATmega128 타이머/카운터 비교표]

	타이머/카운터0	타이머/카운터1	타이머/카운터2	타이머/카운터3
기본구조	8bit	16bit	8bit	16bit
타이머입력	clk I/O	clk I/O	clk I/O	clk I/O
카운터입력	TOSC1와TOSC2 (32.768kHz) TOSC1	T1	T2	T3
타이머 프리스케일러	1,8,32,64, 128,256,1024	1,8,64,256,1024	1,8,64,256,1024	1,8,64,256,1024
레지스터	TCCR0 TCNT0 OCR0 ASSR SFIOR TIMSK TIFR	TCCR1A TCCR1B TCCR1C TCNT1H, TCNT1L OCR1AH, OCR1AL OCR1BH, OCR1BL OCR1CH, OCR1CL ICR1H, ICR1L SFIOR TIMSK, ETIMSK TIFR, ETIFR	TCCR2 TCNT2 OCR2 SFIOR TIMSK TIFR	TCCR3A TCCR3B TCCR3C TCNT3H, TCNT3L OCR3AH, OCR3AL OCR3BH, OCR3BL OCR3CH, OCR3CL ICR3H, ICR3L SFIOR TIMSK, ETIMSK TIFR, ETIFR

[ATmega128 타이머/카운터 비교표]

	타이머/카운터0	타이머/카운터1	타이머/카운터2	타이머/카운터3
동작모드	Normal CTC Fast PWM Phase Correct PWM	Normal, CTC, Fast PWM, Phase Correct PWM, Phase and Frequency Correct PWM	Normal CTC Fast PWM Phase Correct PWM	Normal, CTC, Fast PWM, Phase Correct PWM, Phase and Frequency Correct PWM
입력신호	TOSC1, TOSC2	T1, IC1	T2	T3, IC3
출력신호	OC0	OC1A OC1B OC1C	OC2	OC3A OC3B OC3C
인터럽트	Overflow, Output Compare Match	Overflow, Output Compare Match A/B/C, Input Capture	Overflow, Output Compare Match	Overflow, Output Compare Match A/B/C, Input Capture
기타	RTC기능 타이머카운터모두 프리스케일러 사용	Capture		Capture

[8비트 타이머 / 카운터]

- **특징**
  - 싱글 채널 카운터
  - Clear Timer on Compare Match 기능
    (Auto Reload 기능)
  - 글리치가 없는 Phase Correct PWM 기능
  - 주파수 발생 기능
  - 10비트 프리스케일러 기능
  - Timer0 Overflow Interrupt와 Timer0 Compare Match Interrupt
  - IO 클록과는 별도로 32.768kHz 를 인가 가능한 핀
    (TOSC1, TOSC2 핀)

[8비트 타이머 / 카운터]

- 타이머 0을 제어하기 위한 레지스터
  - 타이머/카운터 0 제어 레지스터 : TCCR0
    - Timer/Counter0 Control Register : TCCR0
  - 타이머/카운터 0 레지스터 : TCNT0
    - Timer/Counter0 Register : TCNT0

Timer/Counter0 Control Register - TCCR0

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	CS02	CS01	CS00	TCCR0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Timer/Counter0 Register - TCNT0

Bit	7	6	5	4	3	2	1	0	
	MSB							LSB	TCNT0
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	

[8비트 타이머 / 카운터]

- **프리스케일러 (Prescaler)**
  - 타이머에 공급하는 입력 클럭의 속도를 조절하는 분주기.
  - 주파수가 상당히 빠르기 때문에 프리스케일러를 클록의 입력으로 사용.
  - 사용 가능한 프리스케일
    - fCLK_IO/8
    - fCLK_IO/32
    - fCLK_IO/64
    - fCLK_IO/128
    - fCLK_IO/256 or fCLK_IO/1024

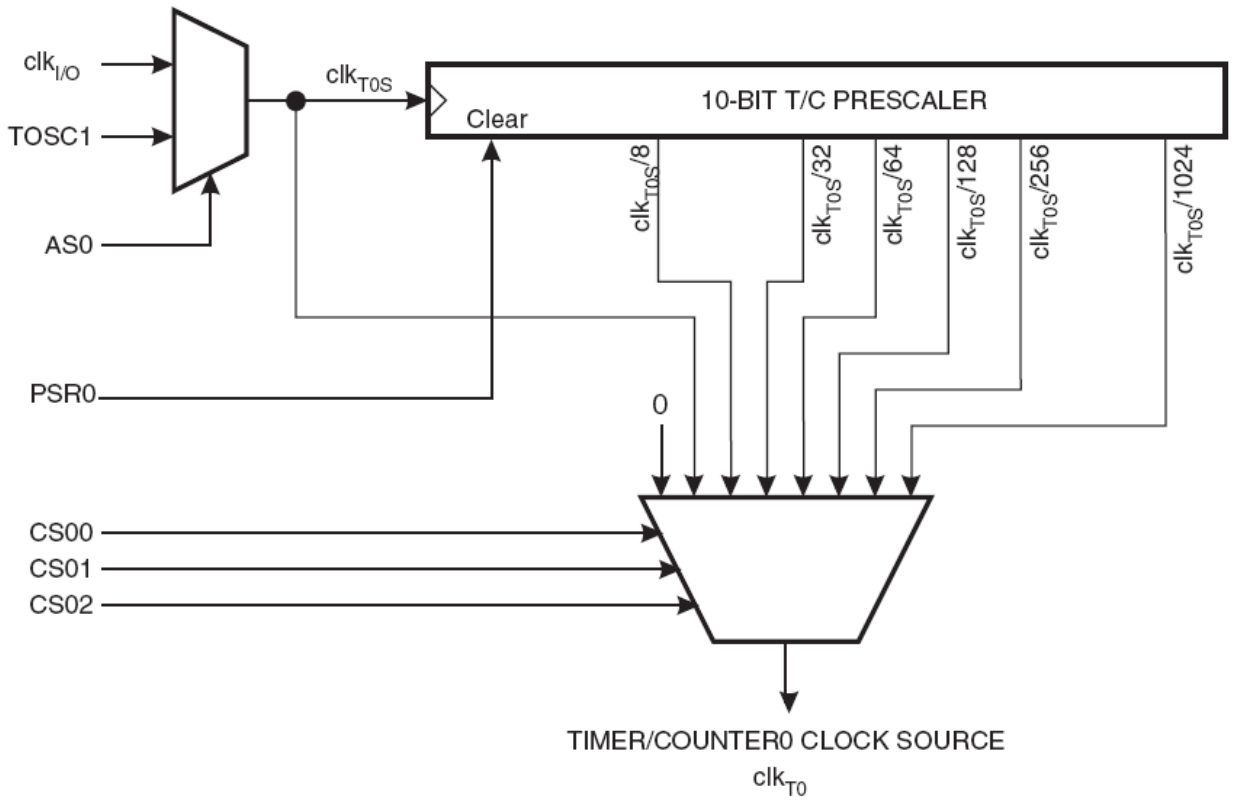
[8비트 타이머 / 카운터]

Bit	7	6	5	4	3	2	1	0	
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	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	

- Timer/Counter0의 클록 소스를 선택하는 레지스터.
- ASSR 레지스터의 AS0(BIT 3) 비트를 “1”로 설정하면 Timer/Counter0을 TOSC1 핀으로 입력되는 비동기 카운터를 사용.
- 즉, Timer/Counter0을 Real Time Counter(RTC)로 설정
- 이 때 TOSC1과 TOSC2는 Timer/Counter0의 클록 소스
- SFIOR 레지스터의 PSR0 비트를 “1”로 설정하면 프리스케일러는 리셋

[8비트 타이머 / 카운터]

- **프리스케일 선택도**



[8비트 타이머 / 카운터]

Special Function IO Register(SFIOR)

Bit	7	6	5	4	3	2	1	0	
	TSM	–	–	–	ACME	PUD	PSR0	PSR321	SFIOR
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **BIT 7 : TSM(Timer/Counter Synchronization Mode)**
  - "1" = Timer/Counter 의 동기 모드가 된다. 이 모드에서는 PSR0와 PSR321에 쓴 값은 유지된다.
  - “0” = PSR0와 PSR321에 쓴 값은 하드웨어에 의해서 클리어된다.

- **BIT 1 : PSR0(Prescaler Reset Timer/Counter0)**
  - "1" = Timer/Counter0의 Prescaler가 리셋된다.
  - “0” = 영향이 없다.

[8비트 타이머 / 카운터]

- **Timer/Counter0의 클록 소스**
  - AS0를 이용하여 외부 또는 내부 클록 소스를 설정.
  - 클록 선택 비트 CS02:0를 설정하여 사용 주파수를 분주함.
  - 카운터 순서는 Timer/counter0 Control Register TCCR0 레지스터의 WGM01과 WGM00을 설정하여 결정.
  - Timer/Counter0의 타이머 값(TCNT0)과 Output Compare Register(OCR0)는 8비트로 구성
  - Timer/Counter0은 내부적으로는 프리스케일러를 통하여 사용할 수 있음
  - 외부적으로는 TOSC1, TOSC2 핀으로 클록을 사용

[8비트 타이머 / 카운터]

Bit	7	6	5	4	3	2	1	0	
	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	TCCR0
Read/Write	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/Counter0 의 동작을 설정하고 프리스케일러를 설정하는 기능을 수행.

- BIT 7 : FOC0 (Force Output Compare)
  - WGM 비트가 non-PWM 모드로 설정되어 있을 때 FOC0 비트는 활성화된다. 그러나 AVR 의 상위 기종과 호
  - 환성을 유지하기 위해서 PWM 모드로 동작할 때는 “0”으로 설정되어야만 한다. 반면에 “1”로 설정하면 Waveform Gerneration Unit 으로 동작한다.

[8비트 타이머 / 카운터]

- **BIT 6, 3 : WGM01:0 (Waveform Generation Mode)**
  - 이 비트를 조절하여 Counter의 카운팅 방향, Maxium(TOP) 카운터 값의 소스 및 어떤 Waveform Generation 을 사용할지를 결정한다.
  - Waveform Generation Mode

Mode	WGM01 (CTC0)	WGM00 (PWM0)	Timer/Counter Mode of Operation	TOP	Update of OCR0	TOV0 Flag Set-on
0	0	0	Normal	0xFF	immediate	MAX
1	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	1	0	CTC	OCR0	immedate	MAX
3	1	1	Fast PWM	0xFF	TOP	MAX

[8비트 타이머 / 카운터]

- **BIT 5, 4 : COM01:0 (Compare Match Output Mode)**
  - 이 비트를 조절하여 OC0 핀의 동작을 조정한다.
  - 핀을 출력으로 사용하기 위하여 DDR 레지스터를 출력으로 설정하여야 한다
  - Compare Output Mode, non-PWM Mode

COM01	COM00	내용
0	0	Normal port operation, OC0 disconnected.
0	1	Toggle OC0 on compare Match
1	0	Clear OC0 on compare match
1	1	Set Oc0 on compare match

[8비트 타이머 / 카운터]

○ Compare Output Mode, Fast PWM Mode

COM01	COM00	내용
0	0	Normal port operation, OC0 disconnected.
0	1	예약
1	0	Clear OC0 on compare match, set OC0 at TOP
1	1	set OC0 on compare match, clear OC0 at TOP

○ Compare Output Mode, phase Correct PWM Mode

COM01	COM00	내용
0	0	Normal port operation, OC0 disconnected.
0	1	예약
1	0	Clear OC0 on compare match when up-counting. Set OC0 on compare match when down-counting.
1	1	Set OC0 on compare match when up-counting. Clear OC0 on compare match when down-counting.

[8비트 타이머 / 카운터]

- **BIT 2, 1, 0**
  - CS02, CS01, CS00 (Clock select 0 bit 2, 1, 0)
  - 클록 선택 비트 2, 1, 0은 타이머0의 free-scaling source 에 정의되어 있다.

CS02	CS01	CS00	설명
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	clk_{t0s}(No prescaling)
0	1	0	clk_{t0s}/8 (From prescaler)
0	1	1	clk_{t0s}/32 (From prescaler)
1	0	0	clk_{t0s}/64 (From prescaler)
1	0	1	clk_{t0s}/128 (From prescaler)
1	1	0	clk_{t0s}/256 (From prescaler)
1	1	1	clk_{t0s}/1024 (From prescaler)

[8비트 타이머 / 카운터]

Timer/Counter0 Register(TCNT0)

Bit	7	6	5	4	3	2	1	0	
	TCNT0[7:0]								TCNT0
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	

- 매 클록소스의 변화마다 1씩 증가.
- TCNT0 레지스터에 Overflow 가 발생되면 인터럽트 발생기능.
- Timer/Counter0의 8비트 카운터 값을 저장하고 있는 레지스터이다.
- 읽고 쓰기 동작을 할 때 접근 가능하다.

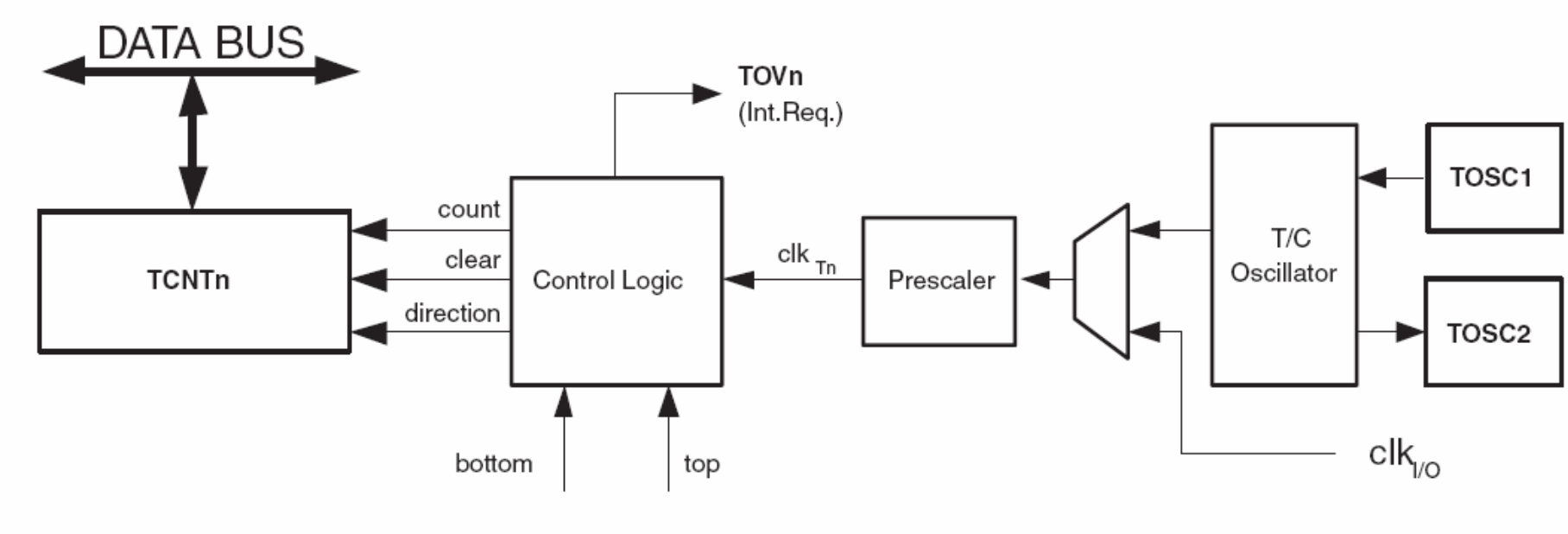
[8비트 타이머 / 카운터]

Timer/Counter0 Output Control Register(OCR0)

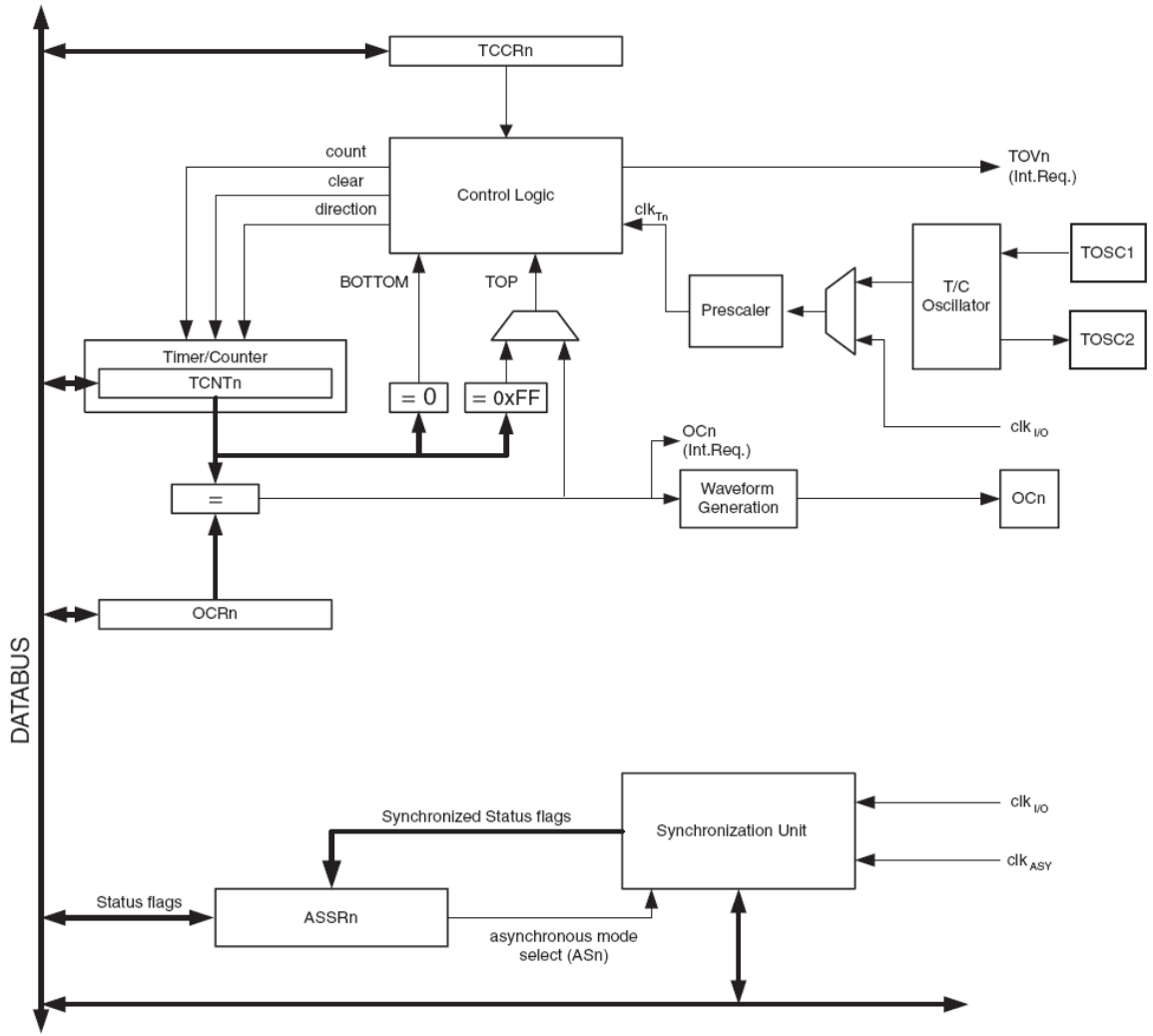
Bit	7	6	5	4	3	2	1	0	
	OCR0[7:0]								OCR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- Timer/Counter0 Register TCNT0 값과 비교하여 OC0 핀에 출력신호를 발생하기 위한 8비트 값을 저장하는 레지스터이다.
- TCNT0 값과 비교하여 매치 동작이 발생하면 OC0 핀에 출력신호를 발생하기 위한 8비트 값을 저장하는 레지스터이다.

Counter Unit Block Diagram



[Block Diagram]

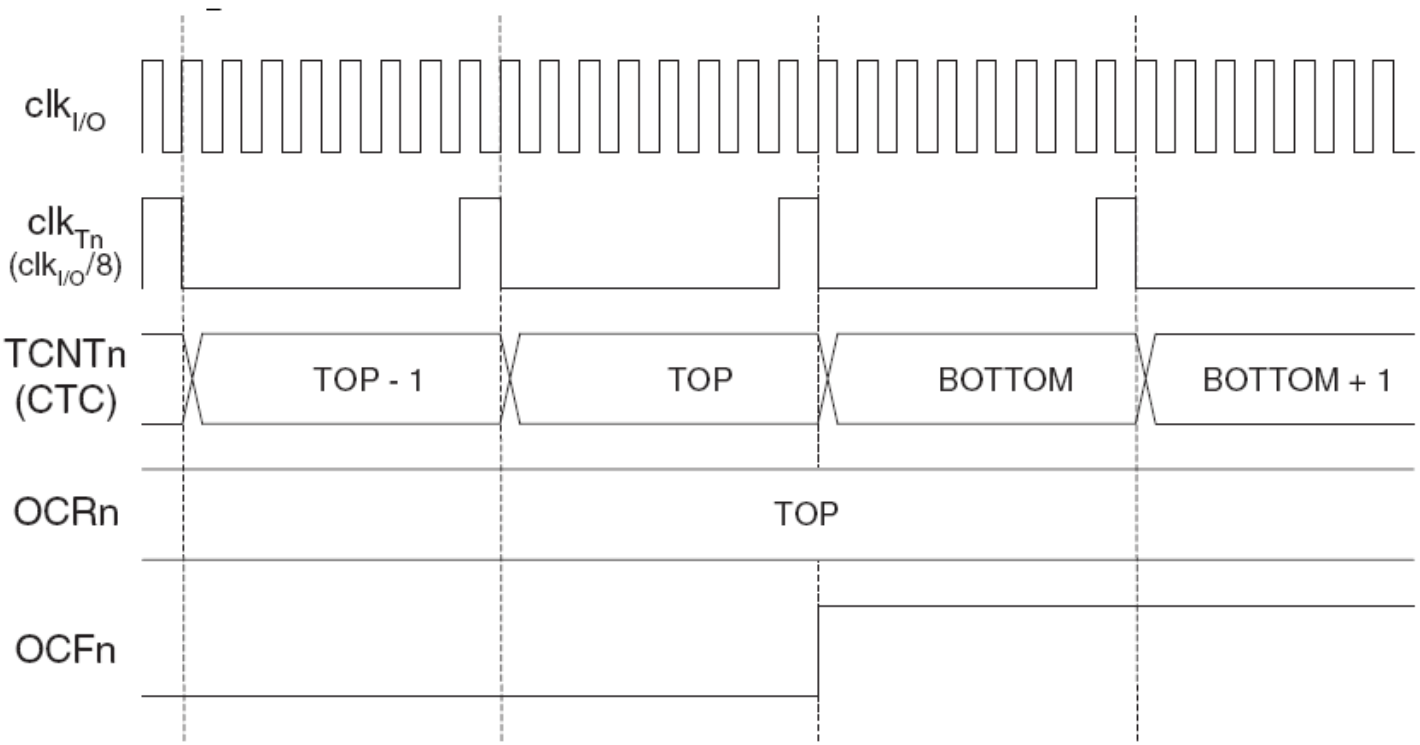


[8비트 타이머 / 카운터]

- **Timer/Counter0의 Output Compare 장치**
  - 8비트 비교기 : 연속해서 TCN0 값과 OCR0 값을 비교해 나간다.
  - TCNT0 값과 OCR0 값이 같을 때마다 비교기는 match 신호를 발생한다.
  - match 신호는 타이머 클록의 다음 사이클에서 Output Compare Flag(OCF0)를 셋함.
  - OCIE0 비트와 SREG 레지스터의 I 비트가 “1”로 셋 되어 있으면 OCF0는 Output Compare Interrupt를 발생한다.
  - OCF0 플래그는 인터럽트가 실행이 되고 나면 자동으로 클리어된다.(OCF0 비트에 “1”을 써서 소프트웨어적으로 클리어 하여도 된다.

[8비트 타이머 / 카운터]

- **Timer/Counter0 타이밍(f_clk IO/8 Prescaler)**



[8비트 타이머 / 카운터]

- **동작 모드**
  - **Normal Mode**
    - 일반적인 타이머 오버플로우 인터럽트가 필요할 때 사용
    - 상향카운터
    - 0x00 ~ 0xFF 계수동작 반복
    - 카운트 도중 Clear 없음
    - 오버플로우(OVF) 인터럽트 (MAX=0xFF값일 때 발생)
    - 비교매치(COMP) 인터럽트
      (파형을 예상하지 못하기 때문에 추천하지 않음)

[8비트 타이머 / 카운터]

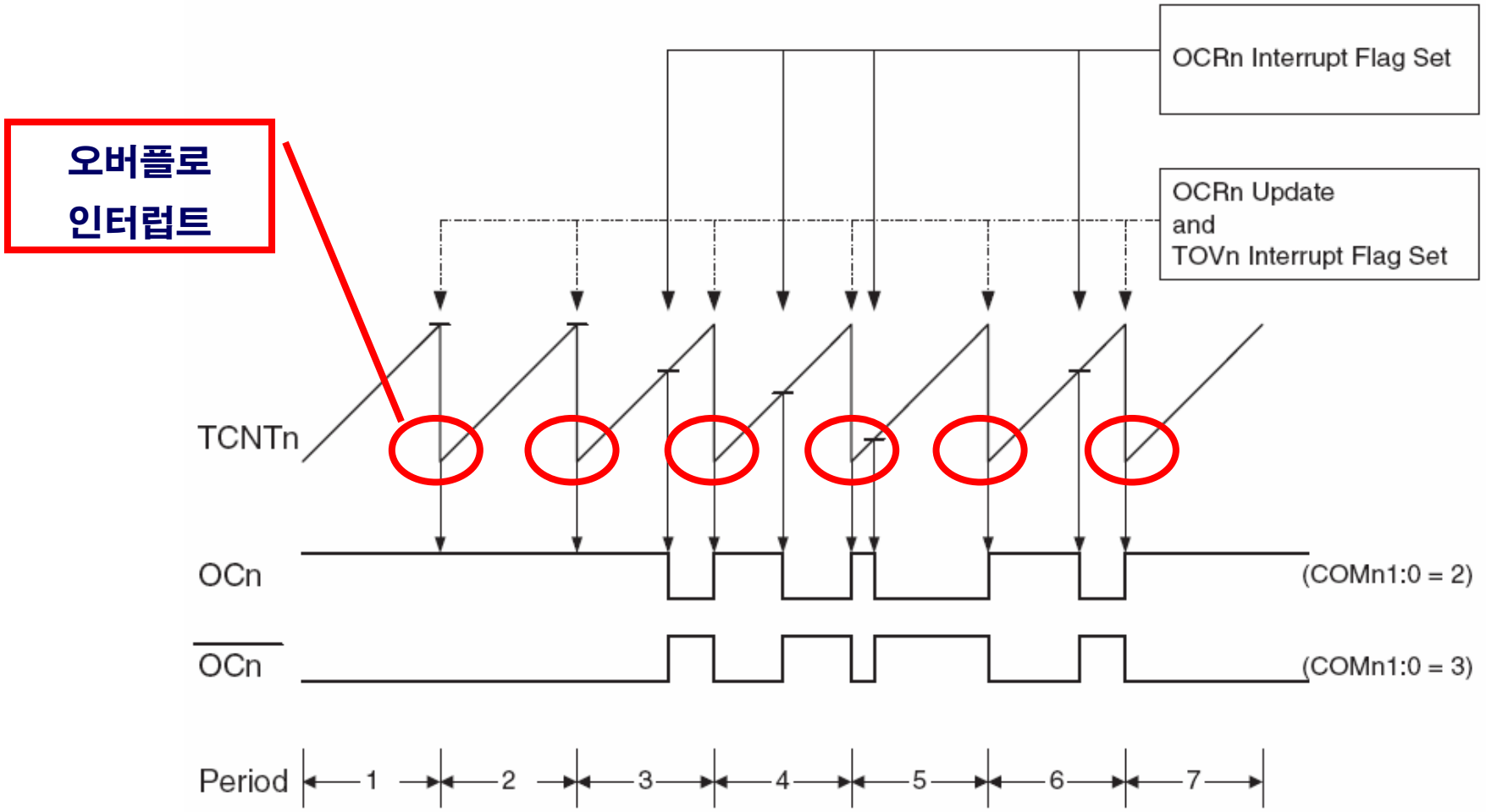
- **CTC Mode(Compare Timer on Compare Match Mode)**
  - 주파수 분주 기능으로 주로 사용
  - 상향카운터
  - 0x00 ~ OCR0 계수 동작 반복
  - OCR0값과 TCNT0값이 같으면 카운트 도중 Clear
  - 오버플로우(OVF) 인터럽트

    (MAX=OCR0값일 때 발생, COMP인터럽트와 동일하게 작동되기 때문에 추천하지 않음)
  - 비교매치(COMP) 인터럽트

Mode	COM01	COM00	설명
PWM모드가 아닌 경우 (Normal / CTC)	0	0	범용 입출력포트 (OC0 출력 차단)
	0	1	비교매치>> OC0 Toggle 출력
	1	0	비교매치>> OC0 = 0 출력
	1	1	비교매치>> OC0 = 1 출력

[8비트 타이머 / 카운터]

- **CTC Mode**



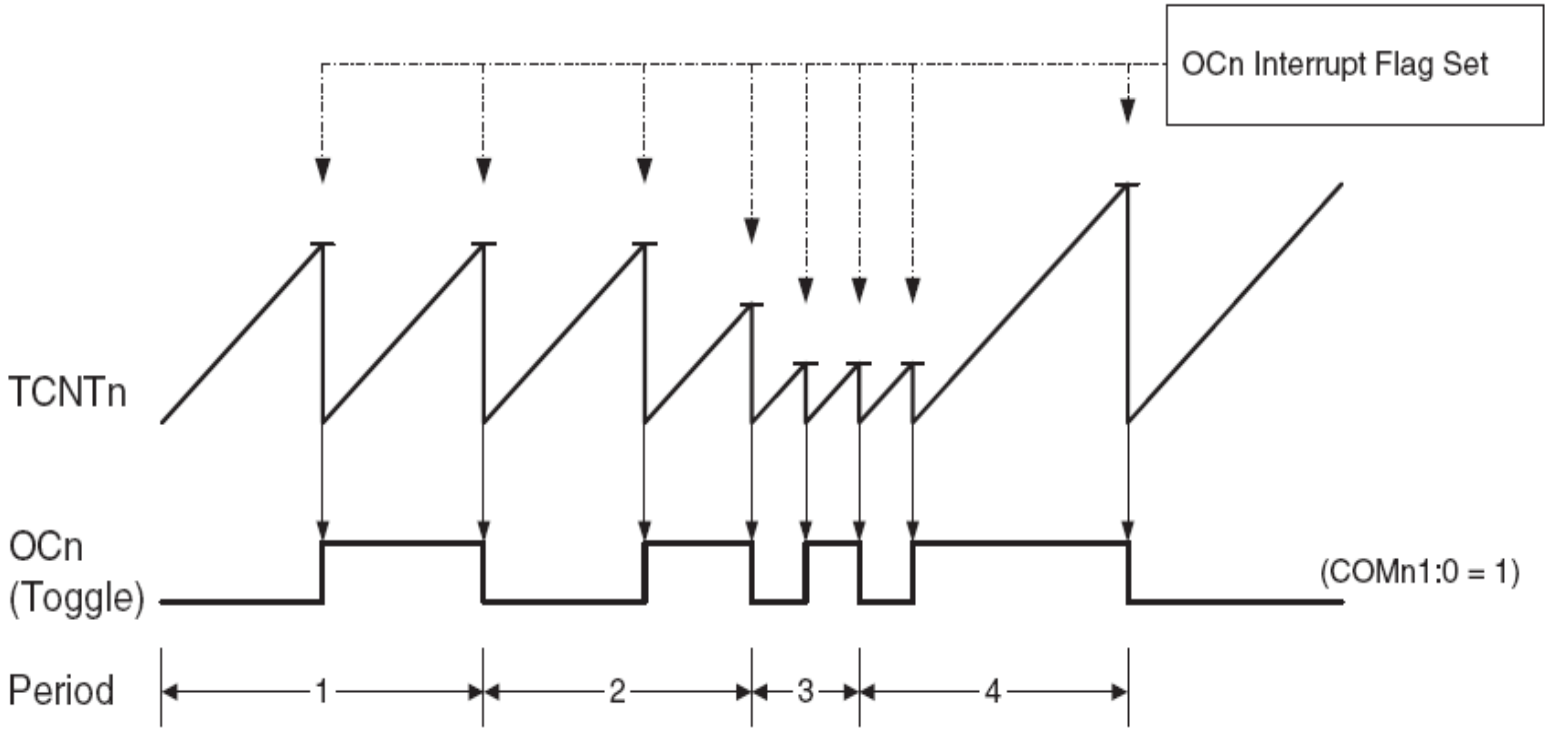
[8비트 타이머 / 카운터]

- **FAST PWM**
  - 높은 주파수 PWM 파형발생이 필요할 때 사용
  - 상향카운터 (Single-Slope Operation)
  - 0x00 ~ 0xFF 계수 동작 반복
  - TCNT0과 OCR0의 Compare Match되면 OC0에 LOW출력(COM0 1:0 = 2)
  - 0xFF → 0x00 오버플로우되면 OC0에 HIGH출력(COM0 1:0 = 2)

Mode	COM01	COM00	설명
FAST PWM	0	0	범용 입출력포트 (OC0 출력 차단)
	0	1	(reserved)
	1	0	비교매치>> OC0 = 0 출력 오버플로우>> OC0 = 1 출력
	1	1	비교매치>> OC0 = 1 출력 오버플오우>> OC0 = 0 출력

[8비트 타이머 / 카운터]

- **Fast PWM Mode**



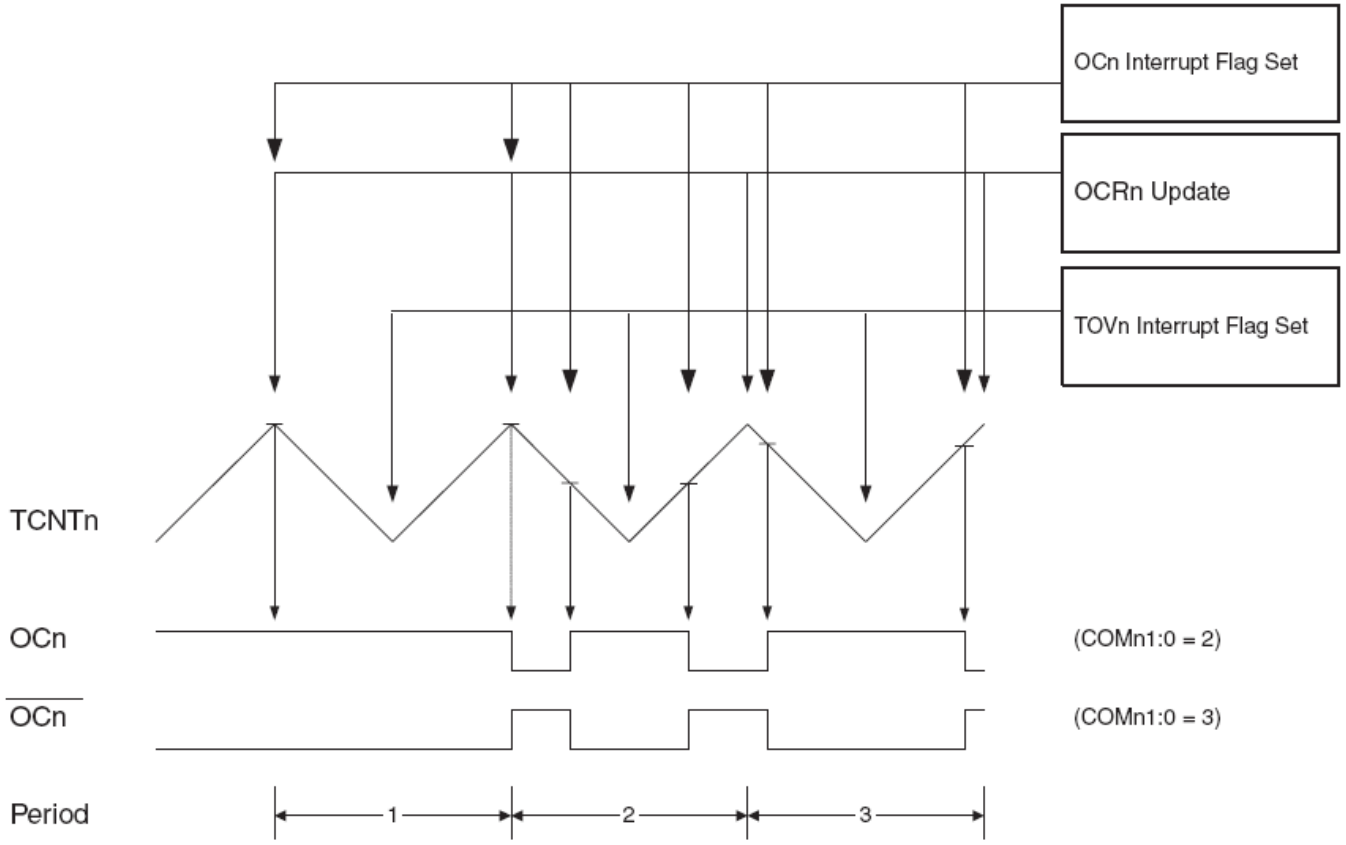
[8비트 타이머 / 카운터]

- **Phase Correct PWM**
  - 높은 분해능의 PWM출력 파형을 발생하는데 사용
  - 상향카운터 0x00 → 0xFF
  - 하향카운터 0xFF → 0x00
  - 0x00 ~ 0xFF ~ 0x00 계수 동작 반복
  - 상향카운터 비교매치>> OC0 = 0 출력(COM0 1:0 = 2)
  - 하향카운터 비교매치>> OC0 = 1 출력(COM0 1:0 = 2)

Mode	COM01	COM00	설명
Phase Correct PWM	0	0	범용 입출력포트 (OC0 출력 차단)
	0	1	(reserved)
	1	0	상향카운터 비교매치>> OC0 = 0 출력 하향카운터 비교매치>> OC0 = 1 출력
	1	1	상향카운터 비교매치>> OC0 = 1 출력 하향카운터 비교매치>> OC0 = 0 출력

[8비트 타이머 / 카운터]

- **Phase Correct PWM Mode**

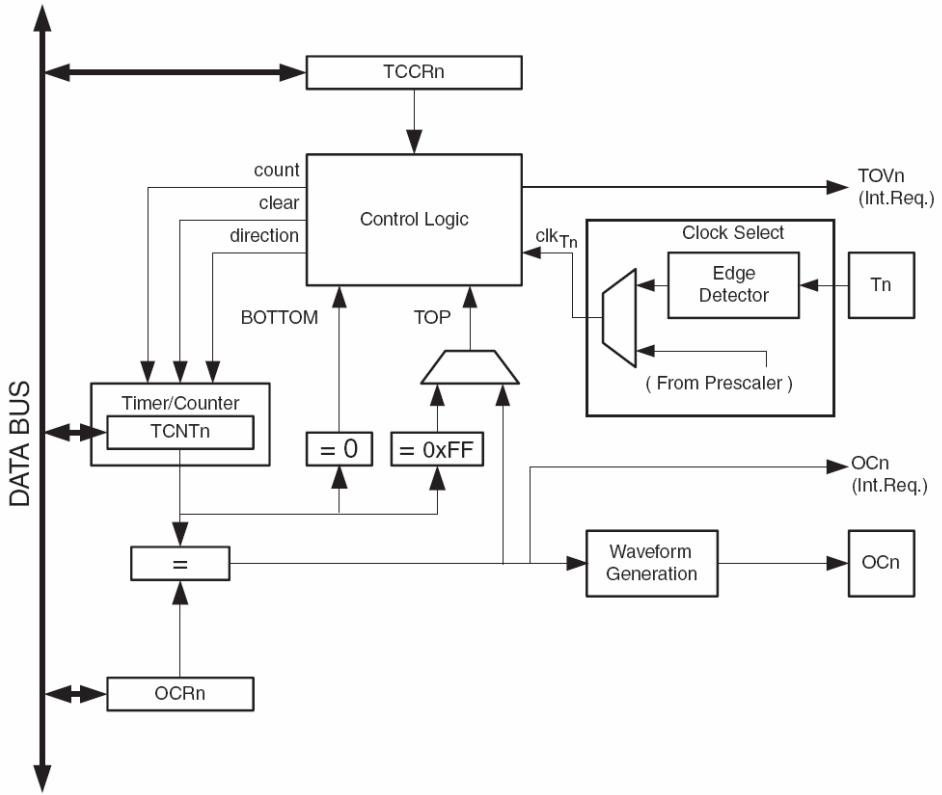


[8비트 타이머 / 카운터]

- **타이머/카운터 0과 2**
  - **같은점**
    - **8비트 구조**
    - **OVERFOLW**
    - **PWM 비슷한 기능**
    - **제어방식도 비슷함.**
  - **차이점**
    - **타이머/카운터 0 은 32.768kHz의 크리스탈을 접속하는 TOSC1 및 TOSC2 단자를 가지고 있음**
    - **RTC의 기능을 갖도록 할 수 있음**
    - **다른 타이머/카운터와는 틀리게 내부 클럭을 사용하든 외부의 클럭을 사용하든 모두 프리스케일러의 분주기능을 사용할 수 있음**

[8비트 타이머 / 카운터]

- **Timer/Counter 2의 Block Diagram**



[8비트 타이머 / 카운터]

- **Normal Mode**

$$f_{OCn} = \frac{f_{clk\_I/O}}{2 \cdot N \cdot (1 + OCRn)}$$

- **PWM Mode**

$$f_{OCnPWM} = \frac{f_{clk\_I/O}}{N \cdot 256}$$

- **N = 프리스케일(1, 8,32,64,128, 256 or1024)**

[8비트 타이머 / 카운터]

CPU에서 사용하는 CLK이 4MHz 일 때 3.2 [ms]마다 주기적으로 인터럽트를 수행하기 위한 레지스터를 설정하여라.

먼저 3.2[ms]가 되기 위해서 TCLK0과 TCNT의 값을 구하면 다음과 같다.

$$3.2\times10^{-3}\,[\mathrm{s}] = \frac{\mathrm{CS}}{4\times10^{6}} \times \mathrm{TCNT0} = \frac{64}{4\times10^{6}} \times (256-200)$$

즉, TCLK0 = CLK/64 이고 TCNT0 = (0xFF - 200) 이다. 따라서 TCCR0 = 0x03, TCNT0 = 0x38이 된다.

[8비트 타이머 / 카운터]

CPU에서 사용하는 CLK이 4MHz이고 TCCR0 = 0x05, TCNT0 = 0x20이 설정되어 있을 때 타이머/카운터0 오버플로 인터럽트가 걸리는 시간은 얼마인가?

TCCR = 0x05 = 00000101이므로 CS02 CS01 CS00 = 1 0 1 이다. 따라서 TCLK0 = CLK / 1024 가 되어 한 클럭의 주기가 1024 / 4,000,000 [sec]이다.

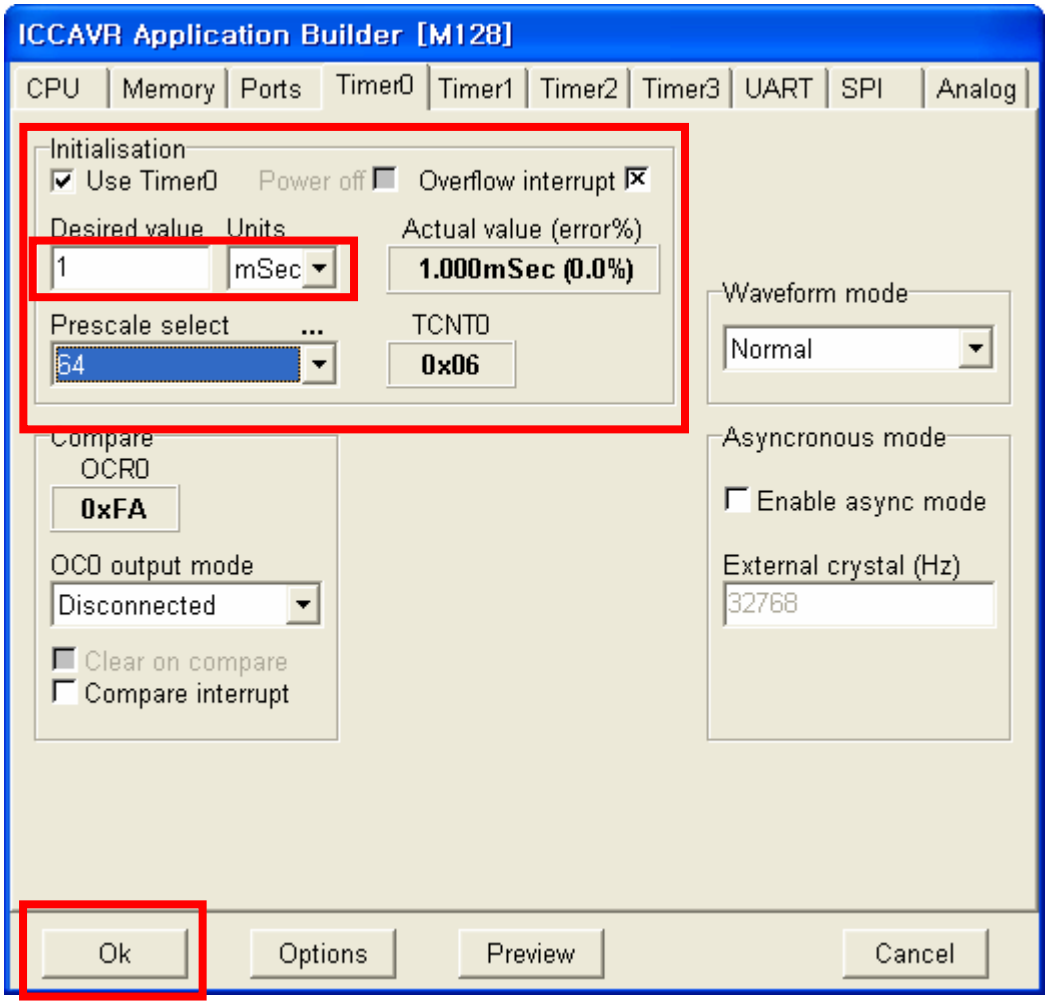
또한 TCNT0 = 0x20에서 시작하여 0xFF까지 카운팅을 하므로 실제 일어나는 계수값은 0xFF - 0x20 = 0xDF 즉 223번 카운팅을 한다.

그러므로 전체 카운팅 되는 시간은

$$\text{Interrupt Period} = \frac{1048}{4\times10^{6}} \times 223 = 0.058426[\text{s}]$$

따라서, 0.058426 [s]마다 인터럽트가 걸린다.

[8비트 타이머 / 카운터 예제]



[8비트 타이머 / 카운터 예제]

- volatile int cnt = 0;

```
#pragma interrupt_handler timer0_ovf_isr:17
void timer0_ovf_isr(void)
{
 TCNT0 = 0x06; //reload counter value
 if(!cnt)
     EX_LED = ~EX_LED;
 cnt++;
 cnt %= 100;
}
```

[8비트 타이머 / 카운터 문제1]

- 타이머를 이용하여 FND에 분과 초를 나타내어라.
- 4개의 FND 중 왼쪽 2개는 분을 오른쪽 2개는 초를 나타낸다.
- 분 및 초의 범위는 0~59 이며, 60이 되면 초는 분으로 분은 시로 넘어가게 된다.
- 시는 표현할 공간이 없으므로 표시하지 않는다.

[8비트 타이머 / 카운터 문제2]

- 다음 LED가 순서대로 점등되도록 만드시오.
  - 순서는 빨강 -> 노랑 -> 파랑 -> 보라 -> 녹색
  - 빨강이 점등 된 후에 노랑이 점등 될 때 빨강은 계속 켜져 있도록 합니다.
  - 최종 녹색이 들어오면 빨강부터 다시 한다.
  - 오른쪽은 녹색까지 전부 점등된 화면입니다.
  - delay가 아닌 Timer를 이용합니다.