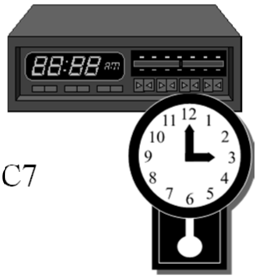
EEL 3744

Menu

- Introduction to OCx
- Output Compare
  - > Special Case for OC
    - – Output Compare: OC0 - OC7
      - ☞OM:OL
    - – OC7
    - – Forced Output Compares
  - > PWM example
  - > Periodic Signal Generation Example
- XMEGA OC/PWM



See docs/examples on web-site:
doc8331 (Sec 14-15), doc8385 (Sec 16-17)

```
68HC12: OC2PWM.asm,
OC2PWM_sim_debug.asm, OC7PWM.asm,
OC5SigGn.asm, OC2_ex_s12.asm
DSC: SPRUG04A
```

EEL 3744

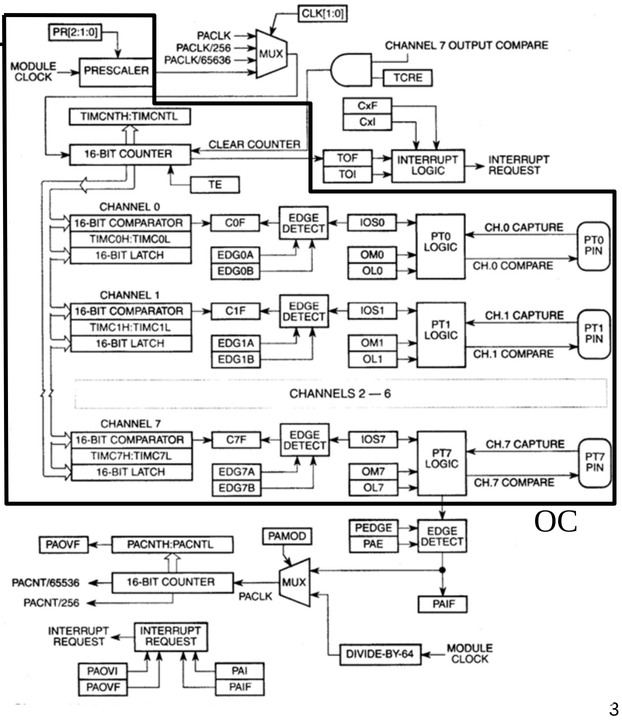
Timer/Counter System

- Microcontrollers are equipped with a precision timing system (much more precise than RTI/RTC systems)
- The Timer/Counter (TC) system is essentially a counter that increments or decrements based on
  - > Regular clock pulses and a timer prescaler (timer)
  - > Irregular event pulses (counter)
- Useful for
  - > Timing
  - > Periodic Interrupts or Event Generation
  - > Pulse Width Modulation
  - > Event counting
  - > Signal Measurements

See doc8385, sec 16

EEL 3744

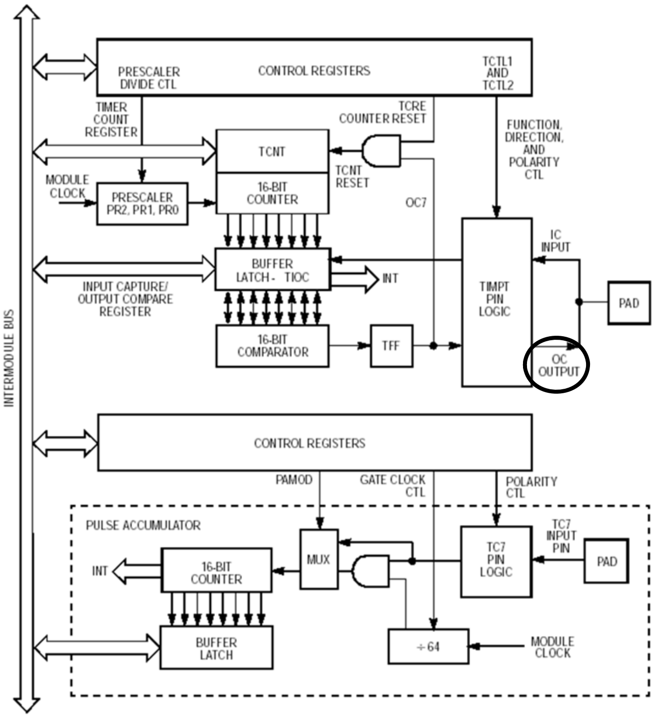
68HC12 Timer System Block Diagram: OC System



S&HE: Fig 10.1

EEL 3

68HC12 Timer Block Diagram: OC System

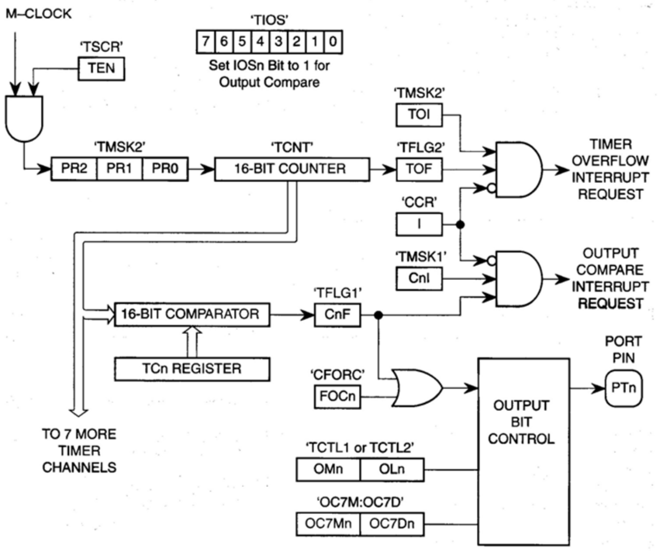


Tech: Fig 12-1

EEL 3744

68HC12 Main Timer System

S&HE: Fig 10.3



EEL 3744

68HC12 TIOS: Timer Input Capture/Output Compare Select Register & DDRT

- TIOS - Timer Input Capture/Output Compare Select Register
  >0 = IC; **1= OC**

	7	6	5	4	3	2	1	0	
$0080	IOS7	IOS6	IOS5	IOS4	IOS3	IOS2	IOS1	IOS0	**TIOS**
RESET	0	0	0	0	0	0	0	0	

- DDRT - Data Direction Register for Port T
  >0 = Input; **1 = Output**

	7	6	5	4	3	2	1	0	
$00AF	DDRT7	DDRT6	DDRT5	DDRT4	DDRT3	DDRT2	DDRT1	DDRT0	**DDRT**
RESET	0	0	0	0	0	0	0	0	

EEL 3744

68HC12 TCNT and TSCR

	7	6	5	4	3	2	1	0	
$0084	Bit 15	-	-	-	-	-	-	Bit 8	**TCNT** High
RESET	0	0	0	0	0	0	0	0	

	7	6	5	4	3	2	1	0	
$0085	Bit 7	-	-	-	-	-	-	Bit 0	**TCNT** Low
RESET	0	0	0	0	0	0	0	0	

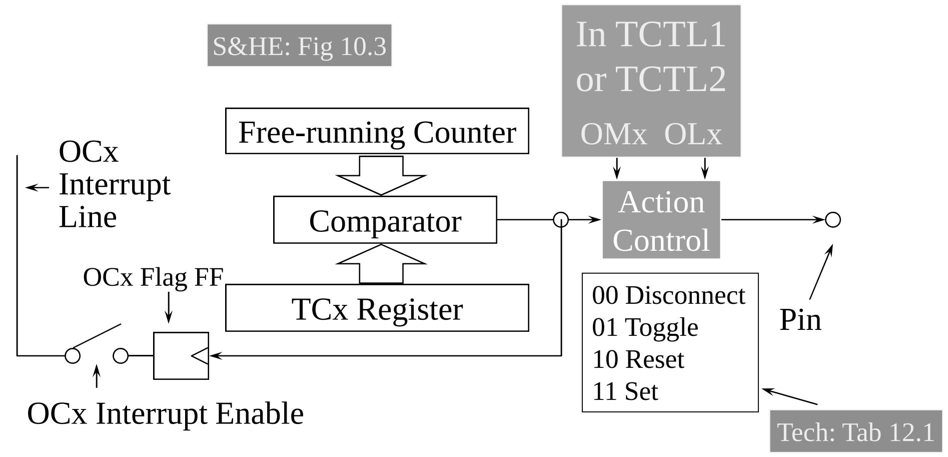
	7	6	5	4	3	2	1	0	
$0086	**TEN**	TSWAI	TSBCK	TFFCA	0	0	0	0	**TSCR**
RESET	0	0	0	0	0	0	0	0	

- TEN (Timer Enable) in TSCR:
  - >0 = disable
  - >**1 = enable**

EEL 3744

68HC12 Output Compare Block Diagram (from OM,OL)

- Block Diagram of Output Compare (from OM,OL)



EEL 3744 **68HC12** Input Capture / Output Compare Registers

- TCx - Timer Input Capture / Output Compare x

	7	6	5	4	3	2	1	0	
$0090	Bit 15	-	-	-	-	-	-	Bit 8	TC0
$0091	Bit 7	-	-	-	-	-	-	Bit 0	
$0092	Bit 15	-	-	-	-	-	-	Bit 8	TC1
$0093	Bit 7	-	-	-	-	-	-	Bit 0	○
				○ ○ ○					○ ○
$009C	Bit 15	-	-	-	-	-	-	Bit 8	TC6
$009D	Bit 7	-	-	-	-	-	-	Bit 0	
$009E	Bit 15	-	-	-	-	-	-	Bit 8	TC7
$009F	Bit 7	-	-	-	-	-	-	Bit 0	

RESET = $0000

EEL 3744 **68HC12** TCTL1 & TCTL2: Time Control Registers

- TCTL1 & TCTL2- Timer Control Registers
  - >Output Mode (OMx) and Output Level (OLx)

	7	6	5	4	3	2	1	0	
$0088	**OM7**	**OL7**	**OM6**	**OL6**	**OM5**	**OL5**	**OM4**	**OL4**	**TCTL1**
RESET	0	0	0	0	0	0	0	0	

	7	6	5	4	3	2	1	0	
$0089	**OM3**	**OL3**	**OM2**	**OL2**	**OM1**	**OL1**	**OM0**	**OL0**	**TCTL2**
RESET	0	0	0	0	0	0	0	0	

00 Disconnect
01 Toggle
10 Reset
11 Set

Tech: Tab 12.1

EEL 3744 **68HC12** TMSK1: Timer Interrupt Mask 1 TFLG1: Timer Interrupt Flag 1

- TMSK1: Timer Interrupt Mask 1 (for IC/**OC**)
  - >0 = Interrupt disabled; 1 = Interrupt enabled

	7	6	5	4	3	2	1	0	
$008C	**C7I**	**C6I**	**C5I**	**C4I**	**C3I**	**C2I**	**C1I**	**C0I**	**TMSK1**
RESET	0	0	0	0	0	0	0	0	

- TFLG1: Timer Interrupt Flag 1 (for IC/**OC**)
  - > 0 = No interrupt (or interrupt cleared); 1 = Interrupt has occurred

	7	6	5	4	3	2	1	0	
$008E	**C7F**	**C6F**	**C5F**	**C4F**	**C3F**	**C2F**	**C1F**	**C0F**	**TFLG1**
RESET	0	0	0	0	0	0	0	0	

CxF Flag:
- **Automatically** set **each** time TCNT = TCx
- To clear, write a 1 to the flag bit

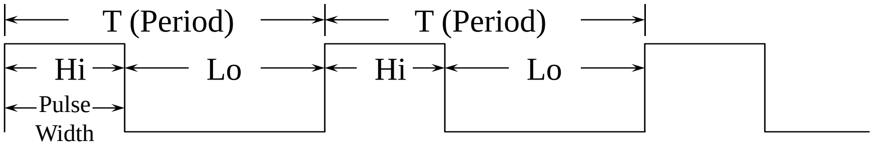
EEL 3744 Pulse Width Modulation (PWM)

- PWM is a method of controlling analog circuits with digital outputs by delivering energy through a sequence of pulses
- A signal (square wave) is generated by controlling in time when to turn a digital signal on or off
- Used for controlling
  - >Servos
  - >Motors
  - >Speakers
  - >Etc.

EEL 3744

PWM Signal

- Pulse Width Modulation (PWM) Signal

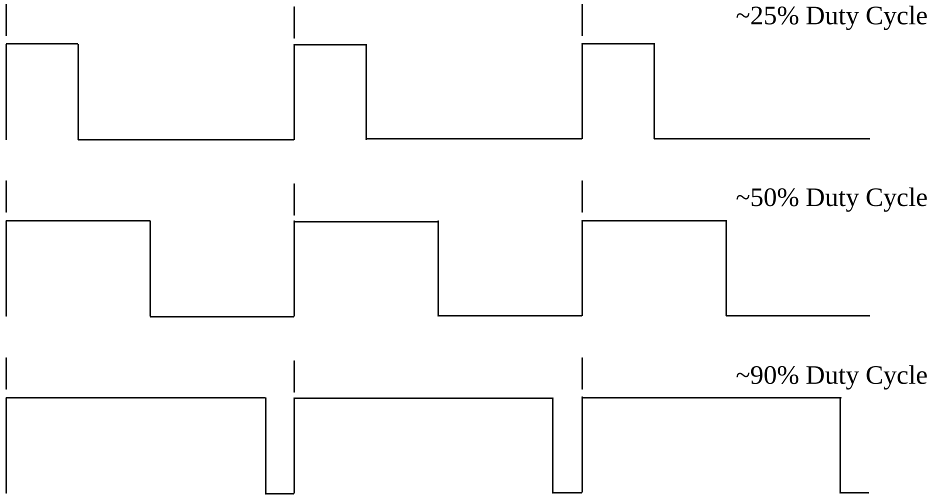


- Duty Cycle is the ratio of “on time” to “off time” during one period
  - >Duty Cycle = (High Time / Period) * 100%

$$\boxed{\textbf{Duty Cycle (\%)} = \frac{\textbf{Hi}}{\textbf{T}} \times \textbf{100}}$$

EEL 3744

PWM Duty Cycle



EEL 3744 **68HC12** Output Compare Example: PWM

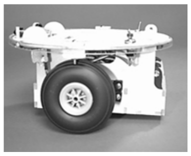
- Example: PWM using OC2 (25% Duty cycle)
  - > T = 32.768ms (default TOF rate)

TCNT:	\$0000-\$FFFF
25%	Hi = \$4000
50%	Hi = \$8000

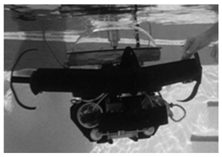
$$\textbf{Duty Cycle (\%)} = \frac{\textbf{Hi}}{\textbf{T}} \times \textbf{100}$$

OC2PWM.asm

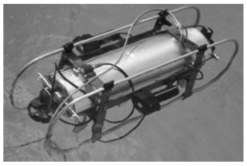
EEL 3744 Period of PWM (PCM for Servos)



- Examples
  - > For ***TJ*** servo motors we use T = 44 ms
  - > For ***Talrik*** servo motors we use T = 32 ms
  - > For ***SubjuGator*** motors we use T = 0.1 ms



- For motors:
  - > To fast ⇒ inefficient
  - > To slow ⇒ herky-jerky



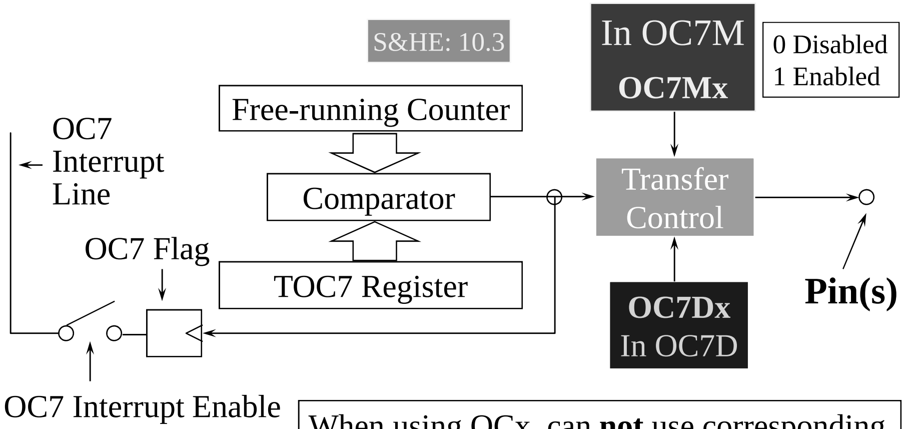
- For servos:
  - > Typically, work only over a small range
    - – For our small servos we use: 30-50 Hz
  - > For our ***Autonomous helicopter*** servos we use T = 20 ms,
    - – Duty cycle between 5 & 10% (1ms ≤ t_{hi} ≤ 2ms)

Show sub web page and sub video

EEL 3744

68HC12 Output Compare: OC7

- Block Diagram of Output Compare (OC7)



When using OCx, can **not** use corresponding Port T pin as a general output

EEL 3744

68HC12 Output Compare: OC7

- TCNT: Timer Counter

	7	6	5	4	3	2	1	0	
$ 0084	Bit 15	-	-	-	-	-	-	Bit 8	TCNT
$ 0085	Bit 7	-	-	-	-	-	-	Bit 0	

RESET = $0000

- TC1: Timer Input Capture / Output Compare Register 7

	7	6	5	4	3	2	1	0	
$009E	Bit 15	-	-	-	-	-	-	Bit 8	TC7
$ 009F	Bit 7	-	-	-	-	-	-	Bit 0	

RESET = $0000

EEL 3744

68HC12 Output Compare: OC7

- OC7M: Output Compare 7 Mask Register

	7	6	5	4	3	2	1	0	
$0082	**OC7M7**	**OC7M6**	**OC7M5**	**OC7M4**	**OC7M3**	**OC7M2**	**OC7M1**	**OC7M0**	**OC7M**
RESET	0	0	0	0	0	0	0	0	

0 = OC7 is disabled
1 = OC7 is enabled **to control the corresponding pin** of Port T

- OC7D: Output Compare 7 Data Register

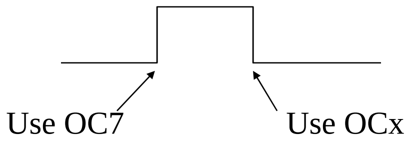
	7	6	5	4	3	2	1	0	
$0083	**OC7D7**	**OC7D6**	**OC7D5**	**OC7D4**	**OC7D3**	**OC7D2**	**OC7D1**	**OC7D0**	**OC7D**
RESET	0	0	0	0	0	0	0	0	

If **OC7Mx = 1**, then data in **OC7Dx is output to Port T bit x** on successful OC7 compares

EEL 3744

68HC12 Using OC7 Along with Another OC Pin

- OC7 can be used along with another OCx feature
- This allows either OC7 or OCx to change the value of a pin
  - >Two edges can be programmed with these two OC features
  - >Pulses **as short as one E-cycle** can be generated using OC7 and OCx together
    - – This is **not** possible when using only a single OC feature



EEL 3744 **68HC12** Using OC7 to Control Multiple OC outputs

- Programming Example:
  Control PA7 and PA6 by PWM using OC7
  (25% Duty cycle)

> Example
`OC7PWM.asm`

TCNT:	$0000-$FFFF
25%	Hi = $4000
33%	Hi = $5555
50%	Hi = $8000
67%	Hi = $AAAA
75%	Hi = $C000

EEL 3744 **68HC12** Forced Output Compares

- A convenient way to change timer output pin states w/o actually setting up and waiting for OCx match
  e.g., spark timing control in an automotive engine
  > **CAUTION** using CFORC if the action OCx is to toggle

- CFORC - Timer Compare Force

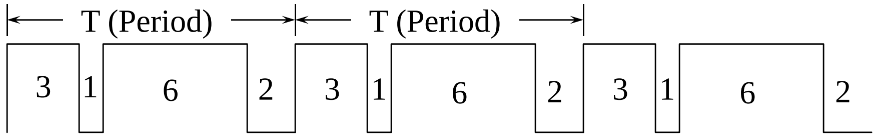
	7	6	5	4	3	2	1	0	
$0081	**FOC7**	**FOC6**	**FOC5**	**FOC4**	**FOC3**	**FOC2**	**FOC1**	**FOC0**	CFORC
RESET	0	0	0	0	0	0	0	0	

0 = Not affected
1 = Automatic pin action programmed for OCx happen as if a match had occurred, but no interrupt is generated (OCxF is not set)

When using OCx, can **not** use corresponding Port T pin as general output [Tech: section 13.4.21]. Use **CFORC**!

EEL 3744 **68HC12** Output Compare Programming Example

- Periodic Signal Generation using Output Compare



Time Unit: 4.096ms = **$2000** (=8192) E-cycles

[Could use: 1 msec = 2000 E-cycles (but hard to see with simulator)]

- Programming Example: Periodic Signal Generation using OC5
  - > Example

OC5SigGn.asm

xmas_lights.wmv

EEL 3744 **68HC12** Using OC for Large Times

- Can the output compare system be used to make very wide pulses accurately?
- Will using TOF help?
- How can you generate 70,000 E-clock high pulse accurately?

EEL 3744

XMEGA 16-bit Timer/Counter Type 0 and Type 1

See doc8331, Sec 14 & doc8385, Sec 16

- XMEGA has a set of eight 16-bit timer/counters (TC)
- Two TCs can be combined to create a 32-bit TC
- A TC consists of a base counter and a set of compare or capture (CC) channels
  - > Waveform generation available
- TC 0 has four CC channels
  - > TC 0 has the split mode feature that split it into two 8-bit Timer/Counters with four compare channels each
- TC 1 has two CC channels

EEL 3744

XMEGA 16-bit Timer/Counter Type 0 and Type 1

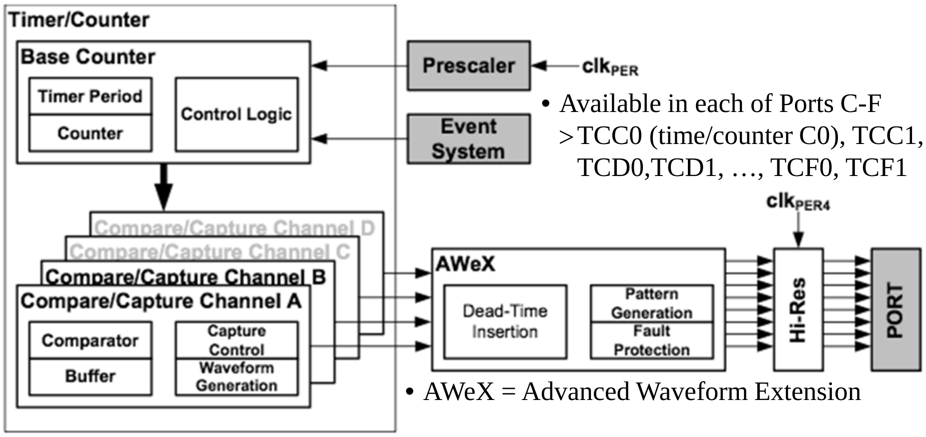
See doc8331, Sec 14 & doc8385, Sec 16

- Input Capture available (see lecture 17)
  - > Input capture with noise cancelling
  - > Frequency capture
  - > Pulse width capture
  - > 32-bit capture
- Timer Overflow and error interrupts/events
- Can be used with event system for:
  - > Quadrature decoding
  - > Count and direction control

EEL 3744

XMEGA Timer/Counter type 0 and type 1

See doc8331, Fig 14-1



- Available in each of Ports C-F
  - >TCC0 (time/counter C0), TCC1, TCD0,TCD1, …, TCF0, TCF1
- AWeX = Advanced Waveform Extension

EEL 3744

XMEGA Prescaler

- Timer is limited by size of counter register and rate at which counter changes
  - >8-bit maxes at count of 256 (~128 μs, if CLK=0.5 μs)
  - >16-bit maxes at count of 65535 (~32,767 μs, CLK=0.5 μs)
- Prescaler modifies the standard timer clock frequency by a chosen value
- Allows timer to be clocked at a desired rate
- Forces a tradeoff between resolution and range
  - >Important with smaller 8-bit and 16-bit counters

EEL 3744

XMEGA Prescaler Example

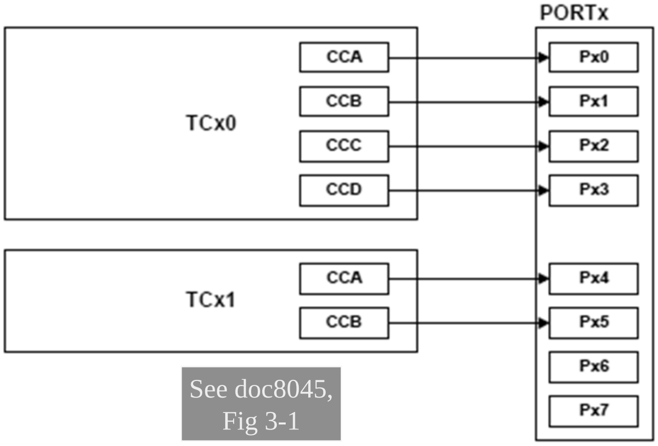
- If clock is at 2 MHz ➔ Period (ticks) = 0.5 μs
  - >An 8-bit counter incrementing at every tick will max out at 256 ticks or 128 μs
  - >If prescaler = 64, the counter updates every 64 ticks (32 μs)
    - – Count changes 4 times during the 256 periods of 0.5 μs clocks
- 256 ticks with prescaler = 64 ➔ 32 μs/tick
  - >256 ticks ➔ 8192 μs = 8.192 ms (=32 μs/tick * 256 ticks)

EEL 3744

XMEGA Timer/Counter Type 0/1 OC I/O Port Pin Mapping

- Timer TCxn, where x indicates the port (C, D, E, or F) and n is the TC number within PORTx.
  - >Example: TCD0 is Timer/Counter 0 connected to PORTD

See doc8045, Sec 3.1

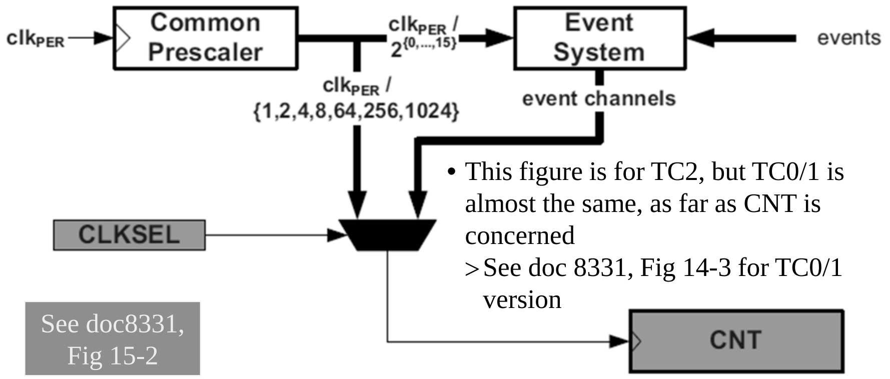


See doc8045, Fig 3-1

EEL 3744

XMEGA TC Clock Sources

- The timer/counter can be clocked from the peripheral clock (clk_{PER}) and from the event system



- This figure is for TC2, but TC0/1 is almost the same, as far as CNT is concerned
  - > See doc 8331, Fig 14-3 for TC0/1 version

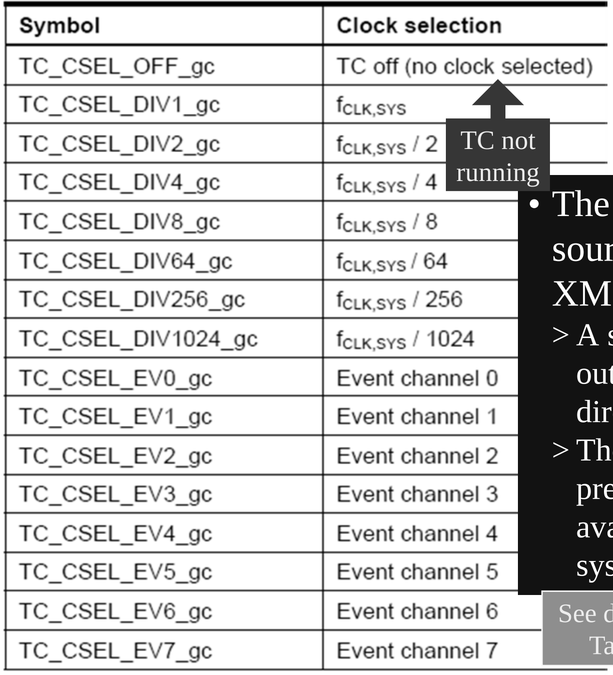
See doc8331, Fig 15-2

University of Florida, EEL 3744 – File **16**
© Drs. Schwartz & Arroyo

32

EEL 3744

XMEGA TC Clock Sources



Symbol	Clock selection
TC_CSEL_OFF_gc	TC off (no clock selected)
TC_CSEL_DIV1_gc	$f_{CLK,SYS}$
TC_CSEL_DIV2_gc	$f_{CLK,SYS}$ / 2
TC_CSEL_DIV4_gc	$f_{CLK,SYS}$ / 4
TC_CSEL_DIV8_gc	$f_{CLK,SYS}$ / 8
TC_CSEL_DIV64_gc	$f_{CLK,SYS}$ / 64
TC_CSEL_DIV256_gc	$f_{CLK,SYS}$ / 256
TC_CSEL_DIV1024_gc	$f_{CLK,SYS}$ / 1024
TC_CSEL_EV0_gc	Event channel 0
TC_CSEL_EV1_gc	Event channel 1
TC_CSEL_EV2_gc	Event channel 2
TC_CSEL_EV3_gc	Event channel 3
TC_CSEL_EV4_gc	Event channel 4
TC_CSEL_EV5_gc	Event channel 5
TC_CSEL_EV6_gc	Event channel 6
TC_CSEL_EV7_gc	Event channel 7

TC not running

- The available clock source selections for the XMEGA TC modules
  - > A selection of prescaler outputs from 1 to 1024 is directly available (**CLKSEL**)
  - > The whole range of time prescalings from 1 to 2^{15} is available through the event system (**HCNT | LCNT**)

See doc8045, Tab 3-1

See doc8331, Sec 15.4

33

EEL 3744 **XMEGA** Timer/Counter type 2

See doc8331, Sec 15 & doc8385, Sec 17

- There are two 8-bit timer/counters (TC) for TC2 for each of the 4 ports (C, D, E, F)
- They are realized when a TC 0 is set in split mode and create a system of two eight-bit timer/counters (from one 16-bit TC0), each with four compare channels
  - > One is the low-byte TC and the other the high-byte TC
  - > Only the low-byte TC can generate compare interrupts
- The two eight-bit TC have a shared clock source and separate period and compare settings
- They can be clocked and timed from the peripheral clock, with optional prescaling, or from the event system
- The counters are always counting down

See doc8331, Fig 15-1 for block diagram

EEL 3744 **XMEGA** Timer/Counter Terminology

See doc8331, Sec 14.2.1

- “**Timer**” is used when the timer/counter clock control is handled by an internal source
- “**Counter**” is used when the clock control is handled externally (e.g., counting external **<u>events</u>**)
- **CC** = **Compare** / **Capture**
  - > When used for compare operations, the **CC** channels are referred to as “**compare channels**”
  - > When used for capture operations, the **CC** channels are referred to as “**capture channels**”

EEL 3744 **XMEGA** Timer/Counter Terminology

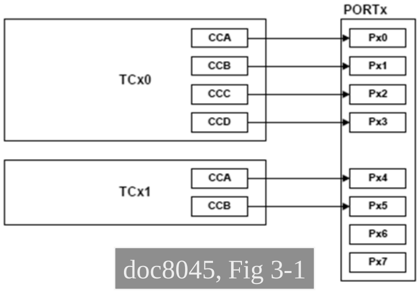
- **CCx –** Compare/Capture registers
  - >In Capture mode, if a capture event is triggered, the current **CNT** value is loaded into the enabled **CCx** register
    - – Used to time intervals between pulses, determine high and low points of input signals, and to define time between two input signals
  - >In Compare mode, the **CNT** register is constantly compared to the **CCx** registers
    - – If **CNT** = **CCx**, then a match event occurs

EEL 3744

XMEGA CCx

- The compare or capture channels consist of a set of 16-bit registers named CCx[H:L], where x indicates the channel (A, B, C, D)
  - >Timer0, with 4 channels, has
    - – CCA[H:L], CCB[H:L], CCC[H:L] and CCD[H:L]
  - >Timer1, with 2 channels has
    - – CCA[H:L] and CCB[H:L].
  - >Each CCx[H:L] register has an associated buffer register CCxBUF[H:L].

See doc8045, Sec 3-5



doc8045, Fig 3-1

EEL 3744 **XMEGA** Timer/Counter Terminology

See doc8331, Sec 14.2.1

- **CNT** – Count register is incremented or decremented every clock cycle (possibly modified by prescaler)
  - >Used by TC module to perform compare/capture operations
  - >May be read or written to as needed
- **PER** – Period register holds the “TOP” value for the TC count

EEL 3744 **XMEGA** Timer/Counter Terminology

See doc8331, Sec 14.2.1

- **BOTTOM**: When the counter reaches zero
- **MAX**: The counter reaches MAXimum (all ones)
- **TOP**: The counter reaches TOP when it becomes equal to the highest value in the count sequence
  - > The TOP value can be equal to the period (PER) or the compare channel A (CCA) register setting
    - – This is selected by the waveform generator mode
- **UPDATE**: The timer/counter signals an update when it reaches BOTTOM or TOP, depending on the waveform generator mode

EEL 3744

XMEGA TC Modes of Operation

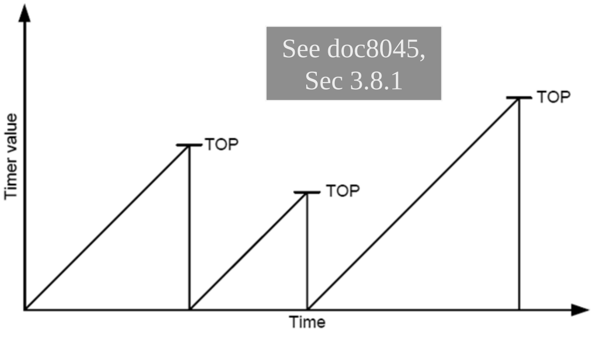
- Normal mode
- Frequency Generation mode
- Single Slope PWM
- Dual Slope PWM, overflow on TOP
- Dual Slope PWM, overflow on TOP and BOTTOM
- Dual Slope PWM, overflow on BOTTOM

See doc8045, Sec 3.8

EEL 3744

XMEGA TC Normal Mode

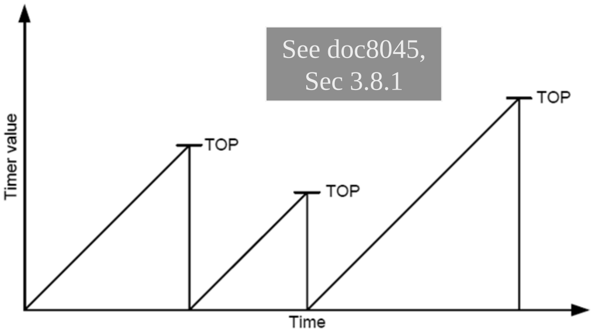
- In Normal Mode, the counter will count in direction set by the DIR bit in CTRLF for each clock until it reaches TOP (when counting up), set by PER[H:L], or BOTTOM (zero, when counting down)
- When TOP is reached when up-counting the counter will be set to zero when the next clock is given
  > If the TC is down-counting the value will wrap around to the value in PER[H:L] after reaching BOTTOM



See doc8045, Sec 3.8.1

EEL 3744 **XMEGA** TC Normal Mode

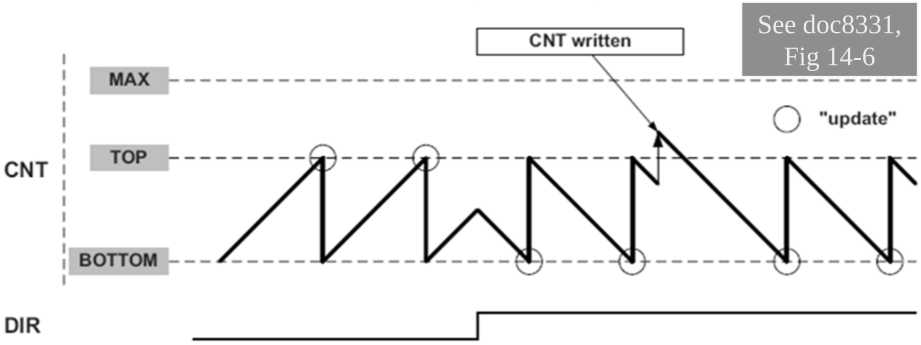
- Changing the counter value while the counter is running is allowed
- The write access has higher priority than count, clear, or reload and will be immediate
  - > However, if the value written is outside the BOTTOM-TOP boundary the counter either has to count down until TOP is reached or count up until wraparound (passing **MAX**) for the timer to re-stabilize to the period time



See doc8045, Sec 3.8.1

EEL 3744 **XMEGA** TC Normal Mode (DIR)

- The counter will count in the direction set by the direction (**DIR**) bit for each clock until it reaches **TOP** or **BOTTOM**.
  - > When up-counting & **TOP** is reached, counter set to 0 when next clock is given
  - > When down-counting, counter is reloaded with period register value when **BOTTOM** is reached
- It is possible to change counter value when the counter is running
- Direction of counter can be changed during normal operation



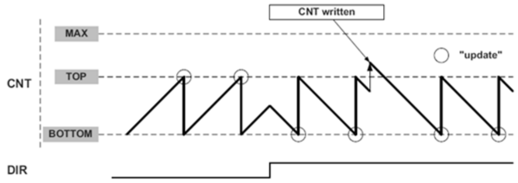
See doc8331, Fig 14-6

EEL 3744

XMEGA Timer Normal Operation

- During normal operation, **CNT** is continuously compared to **PER** or **0**
  - > This checks whether **CNT** has reached **TOP** or **BOTTOM**
- If up-counting, when **CNT** = **PER**, an 'update' condition occurs and CNT register is reset to 0
- If down-counting, when **CNT** = **0**, an 'update' condition occurs and CNT register is reset to **PER**'s value

See doc8331, Fig 14-6

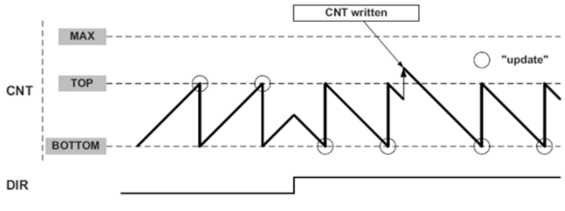


EEL 3744

XMEGA Timer Normal Operation

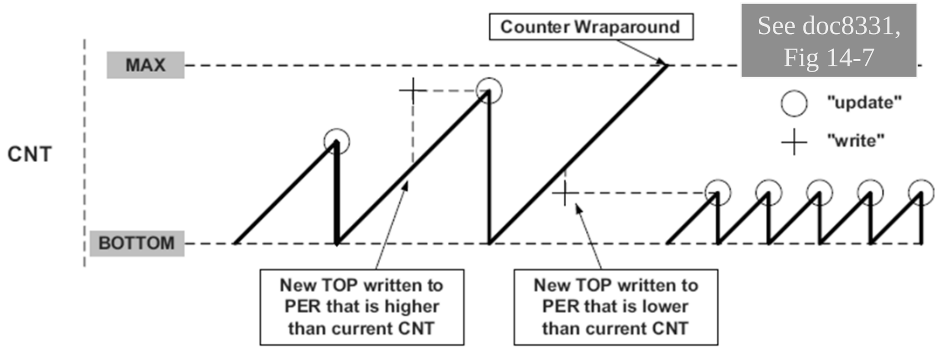
- Changing the counter value during operation is allowed
  - > If the counter value is set above the **TOP** value it will either count down until **TOP** is reached or count up until the value wraps around and starts again at 0
- The **TOP** value may also be changed in **PER** during operation and will result in varying periods for the timer

See doc8331, Fig 14-6



EEL 3744 **XMEGA** TC Normal Mode (PER)

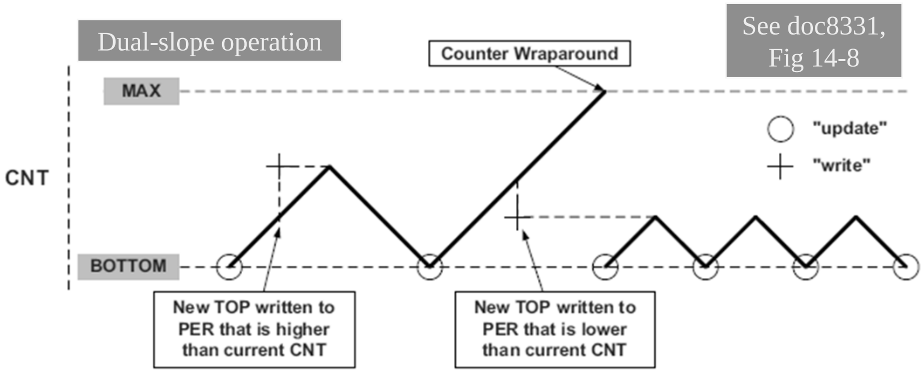
- Counter **period** is changed by writing a new **TOP** value to the period register


MAX
CNT
BOTTOM
Counter Wraparound
"update"
"write"
New TOP written to PER that is higher than current CNT
New TOP written to PER that is lower than current CNT

See doc8331, Fig 14-7

EEL 3744 **XMEGA** TC Dual Slope Mode (Unbuffered)

- A counter wraparound can occur in any mode of operation when up-counting without buffering, as shown below
  - > This due to the fact that the CNT and PER are continuously compared, and if a new **TOP** value that is lower than current CNT is written to PER, it will wrap before a compare match happen


MAX
CNT
BOTTOM
Counter Wraparound
"update"
"write"
New TOP written to PER that is higher than current CNT
New TOP written to PER that is lower than current CNT

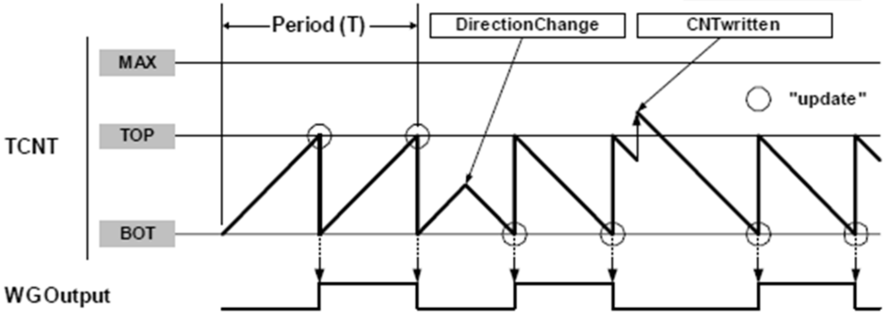
Dual-slope operation

See doc8331, Fig 14-8

EEL 3744 **XMEGA** TC Frequency Generation Mode

- There is little difference between the frequency waveform generation mode (**FRQ**) and the normal mode of operation
- For **FRQ**, the period (T) is controlled by the CCA[H:L] register instead of PER[H:L] (PER[H:L] is not used)
- Waveform Generation (**WG**) output is toggled on each compare match between CNT[H:L] and CCA[H:L]

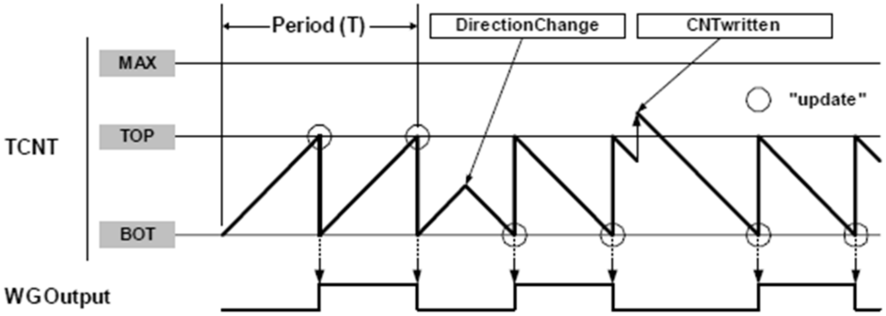
See doc8045, Sec 3.8.2



EEL 3744 **XMEGA** TC Frequency Generation Mode

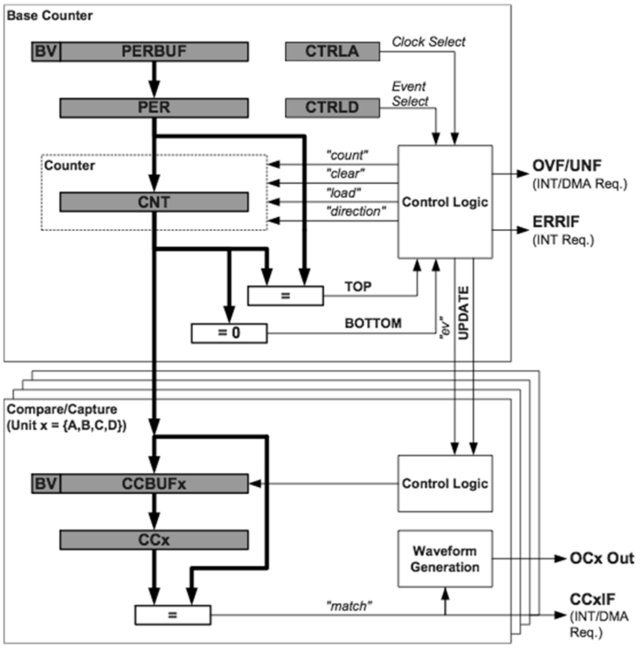
See doc8045, Sec 3.8.2

- The waveform generated will have a maximum frequency of $f_{clk}/2$ when CCA[H:L] is set to zero (0x0000)
- The waveform frequency is defined by eqn: $f_{FRQ} = \frac{f_{CLK}}{2 \cdot N(CCA+1)}$
  > N represents the TC clock prescaler
- The Overflow Status Flag (OVFIF) or Compare A Flag (CCAIF) can be used to generate interrupts
  > If enabled, the interrupt handler routine can be used for updating CCA[H:L] to change the output frequency



EEL 3744

XMEGA Timer System Block Diagram



doc8331: Figure 14-2

University of Florida, EEL 3744 – File **16**
© Drs. Schwartz & Arroyo

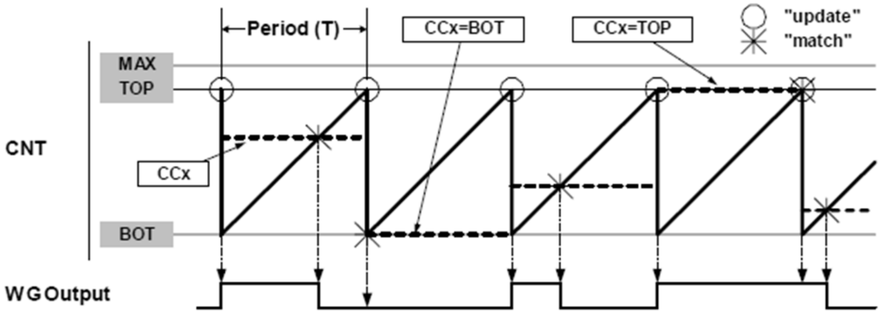
50

EEL 3744

See doc8045, Sec 3.8.3

XMEGA Single Slope PWM Mode

- Counter counts from BOTTOM to TOP then restarts from BOTTOM
- The waveform generator output is set on the compare match between the count and compare registers, and cleared at TOP
- The mode provides twice the PWM frequency than dual-slope PWM

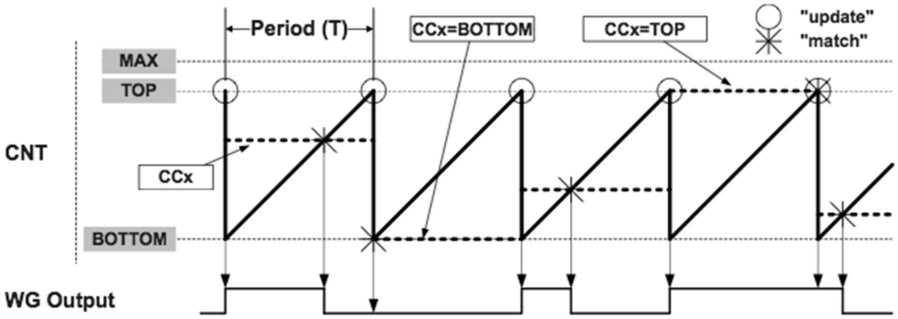


Univer

51

EEL 3744

XMEGA Single-slope Pulse Width Modulation (PWM)



- WG output is
  - > Set on compare match between **CNT** and **CCx** registers
  - > Cleared at **TOP**

doc8331: Figure 14-15

EEL 3744

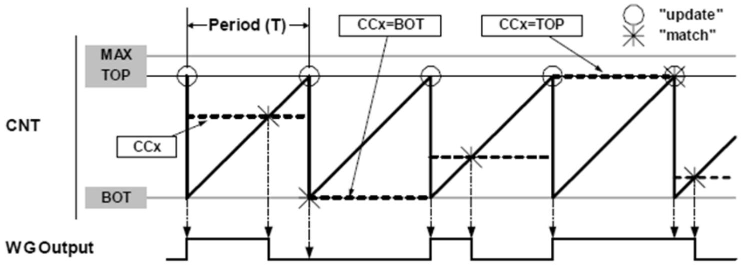
See doc8045, Sec 3.8.3

XMEGA Single Slope PWM Mode

- The PWM base frequency depends on the period setting (PER[H:L]), system clock frequency, and clock prescale.
- The PWM base frequency equation is below, where *N* represents the TC clock prescaler

$$f_{PWM\_SS} = \frac{f_{CLK}}{N(PER+1)}$$

- Use Overflow Status Flag (OVFIF) or Compare Flag (CCxIF) to generate interrupts
- If enabled, the ISR can be used for updating the period and compare buffer values



EEL 3744 **XMEGA** Single-slope Pulse Width Modulation (PWM)

- Both the Overflow Status Flag (OVFIF) and Compare Flag (CCxIF) may be used to generate interrupts
- When enabled, these interrupts may be used to update the period and compare buffer values

EEL 3744 **XMEGA** Single-slope Pulse Width Modulation (PWM)

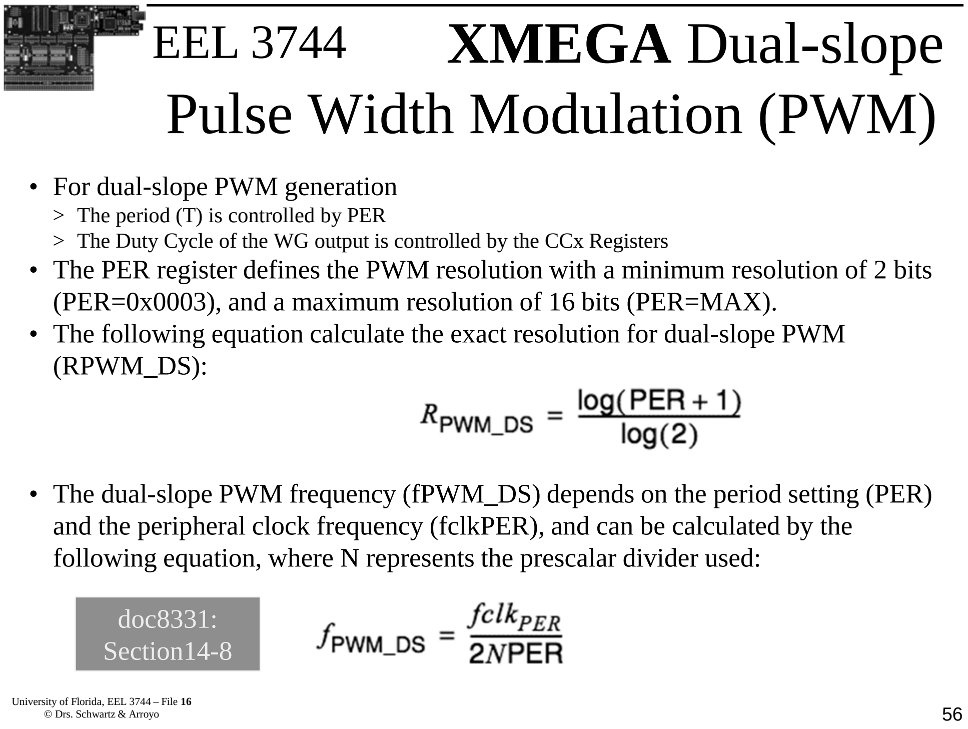
- For single-slope PWM generation
  - > The period (T) is controlled by PER
  - > The Duty Cycle of the WG output is controlled by the CCx Registers
- The PER register defines the PWM resolution with a minimum resolution of 2 bits (PER=0x0003), and a maximum resolution of 16 bits (PER=MAX).
- The following equation calculate the exact resolution for single-slope PWM (RPWM_SS):

$$R_{\mathrm{PWM\_SS}} = \frac{\log(\mathrm{PER}+1)}{\log(2)}$$

- The single-slope PWM frequency (fPWM_SS) depends on the period setting (PER) and the peripheral clock frequency (fclkPER), and can be calculated by the following equation, where N represents the prescalar divider used:

doc8331:
Section 14.8.3

$$f_{\mathrm{PWM\_SS}} = \frac{fclk_{PER}}{N(\mathrm{PER}+1)}$$



EEL 3744 **XMEGA** Dual-slope Pulse Width Modulation (PWM)

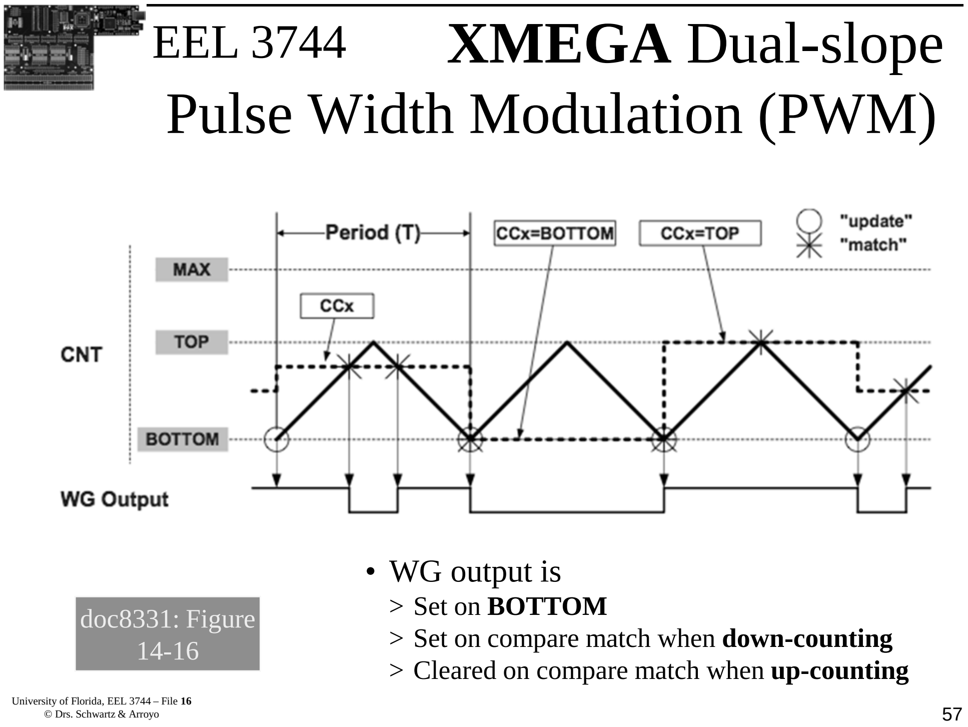
- For dual-slope PWM generation
  - > The period (T) is controlled by PER
  - > The Duty Cycle of the WG output is controlled by the CCx Registers
- The PER register defines the PWM resolution with a minimum resolution of 2 bits (PER=0x0003), and a maximum resolution of 16 bits (PER=MAX).
- The following equation calculate the exact resolution for dual-slope PWM (RPWM_DS):

$$R_{\text{PWM\_DS}} = \frac{\log(\text{PER}+1)}{\log(2)}$$

- The dual-slope PWM frequency (fPWM_DS) depends on the period setting (PER) and the peripheral clock frequency (fclkPER), and can be calculated by the following equation, where N represents the prescalar divider used:

doc8331: Section14-8

$$f_{\text{PWM\_DS}} = \frac{fclk_{PER}}{2N\text{PER}}$$



EEL 3744 **XMEGA** Dual-slope Pulse Width Modulation (PWM)

- WG output is
  - > Set on **BOTTOM**
  - > Set on compare match when **down-counting**
  - > Cleared on compare match when **up-counting**

doc8331: Figure 14-16

EEL 3744

XMEGA CTRLA & CTRLB Register

- CTRLA – Controls the clock source for timers

	7	6	5	4	3	2	1	0	
+0x00	-	-	-	-	CLKSEL3	CLKSEL2	CLKSEL1	CLKSEL0	**CTRLA**
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- CTRLB – Compare or Capture Enables and Waveform Generation Mode

	7	6	5	4	3	2	1	0	
+0x01	CCDEN	CCDEN	CCBEN	CCAEN	-	WGMode2	WGMode1	WGMode0	**CTRLB**
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

EEL 3744

XMEGA CTRLA & CTRLB Register

- CTRLB WGMode Settings

WGMode [2..0]	Group Config	Mode of Operation	Top	Update	OVIF/EVENT
000	Normal	Normal	PER	TOP	TOP
001	FRQ	Frequency	CCA	TOP	TOP
010		Reserved	-	-	-
011	SingleSlope	Single-Slope PWM	PER	BOTTOM	BOTTOM
100		Reserved	-	-	-
101	DSTOP	Dual-Slope PWM	PER	BOTTOM	TOP
110	DSBOTH	Dual-Slope PWM	PER	BOTTOM	TOP and BOTTOM
111	DSBOTTOM	Dual-Slope PWM	PER	BOTTOM	BOTTOM

- Example:
  TCC0_CTRLB = 0xF0;

EEL 3744

XMEGA CTRLC & CTRLD Register

- CTRLC – Allows direct access to the waveform generators output compare values

	7	6	5	4	3	2	1	0	
+0x02	-	-	-	-	CMPD	CMPC	CMPB	CMPA	**CTRLC**
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- CTRLD – Handles Event Control

	7	6	5	4	3	2	1	0	
+0x03	EVACT2	EVACT1	EVACT0	EVDLY	EVSEL3	EVSEL2	EVSEL1	EVSEL0	**CTRLD**
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	

EEL 3744

See doc8331:
Sec 14.12.8

XMEGA CTRLE Register

- CTRLE – Control Register E
  - > Controls byte mode of Timer/Counter Type 0 (allows switch to Type 2)

	7	6	5	4	3	2	1	0	
+0x04	-	-	-	-	-	-	BYTEM1	BYTEM0	**CTRLE**
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

EEL 3744

See doc8331: Sec 15.10.7

XMEGA CTRLF Register

- CTRLF – Control Register F
  - > CMD: TC Command
    - – Used for software control of timer/counter update, restart, and reset (see table)
    - – Must be used together with CMDEN
  - > CMDEN: Command Enable
    - – Indicate for which TC the command (CMD) is valid

CMD	Group Config	Mode of Operation
00	None	None
01	-	Reserved
10	Restart	Force restart
11	Reset	For hard reset

	7	6	5	4	3	2	1	0	
+0x08	-	-	-	-	CMD1	CMD0	CMDEN1	CMDEN0	**CTRLF**
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

EEL 3744

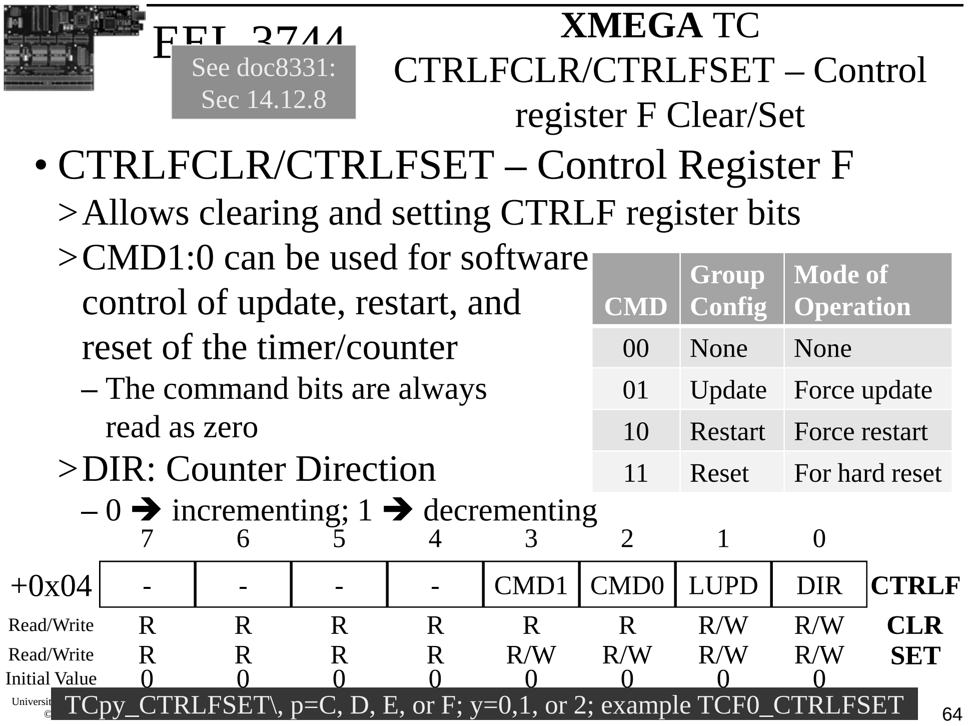
See doc8331: Sec 14.12.8

XMEGA TC CTRLFCLR/CTRLFSET – Control register F Clear/Set

- CTRLFCLR/CTRLFSET – Control Register F
- This register is mapped into two I/O memory locations,
  - > For clearing the register bits (CTRLxCLR)
    - – The individual status is cleared by writing a one to its bit location in CTRLxCLR
  - > For setting the register bits (CTRLxSET)
    - – The individual status is set by writing a one to its bit location in CTRLxSET
- This allows each bit to be set or cleared without use of a read-modify-write operation on a single register

	7	6	5	4	3	2	1	0	
+0x04	-	-	-	-	CMD1	CMD0	LUPD	DIR	**CTRLF**
Read/Write	R	R	R	R	R	R	R/W	R/W	**CLR**
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	**SET**
Initial Value	0	0	0	0	0	0	0	0	

TCpy_CTRLFSET\, p=C, D, E, or F; y=0,1, or 2; example TCF0_CTRLFSET



EEL 3744

See doc8331: Sec 14.12.8

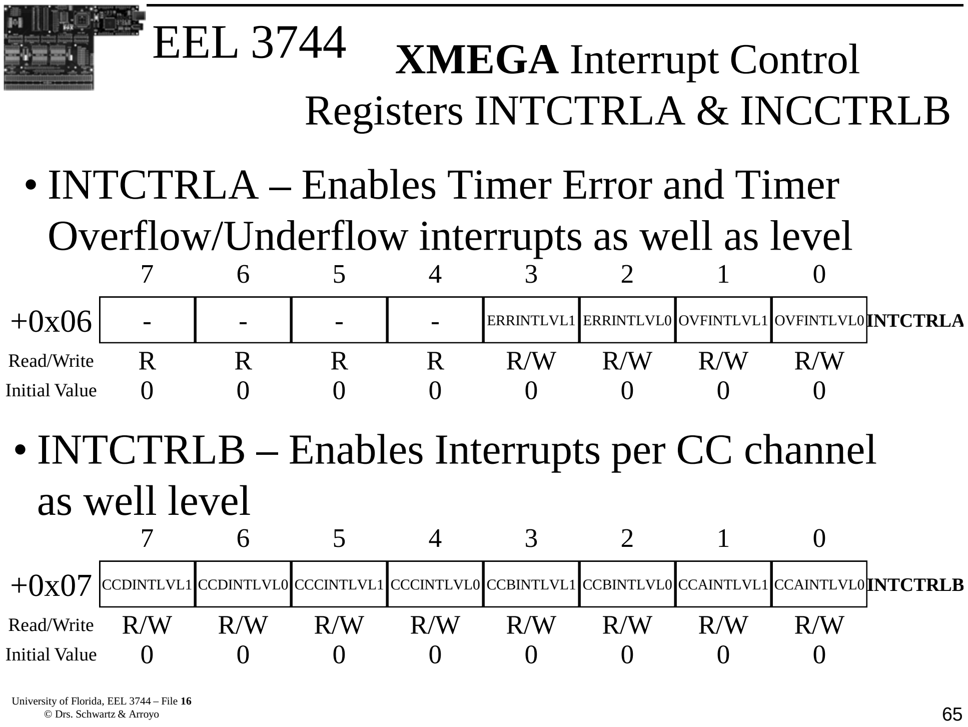
XMEGA TC CTRLFCLR/CTRLFSET – Control register F Clear/Set

- CTRLFCLR/CTRLFSET – Control Register F
  - >Allows clearing and setting CTRLF register bits
  - >CMD1:0 can be used for software control of update, restart, and reset of the timer/counter
    - – The command bits are always read as zero
  - >DIR: Counter Direction
    - – 0 ➔ incrementing; 1 ➔ decrementing

CMD	Group Config	Mode of Operation
00	None	None
01	Update	Force update
10	Restart	Force restart
11	Reset	For hard reset

	7	6	5	4	3	2	1	0	
+0x04	-	-	-	-	CMD1	CMD0	LUPD	DIR	**CTRLF**
Read/Write	R	R	R	R	R	R	R/W	R/W	**CLR**
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	**SET**
Initial Value	0	0	0	0	0	0	0	0	

TCpy_CTRLFSET\, p=C, D, E, or F; y=0,1, or 2; example TCF0_CTRLFSET



EEL 3744

XMEGA Interrupt Control Registers INTCTRLA & INCCTRLB

- INTCTRLA – Enables Timer Error and Timer Overflow/Underflow interrupts as well as level

	7	6	5	4	3	2	1	0	
+0x06	-	-	-	-	ERRINTLVL1	ERRINTLVL0	OVFINTLVL1	OVFINTLVL0	**INTCTRLA**
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- INTCTRLB – Enables Interrupts per CC channel as well level

	7	6	5	4	3	2	1	0	
+0x07	CCDINTLVL1	CCDINTLVL0	CCCINTLVL1	CCCINTLVL0	CCBINTLVL1	CCBINTLVL0	CCAINTLVL1	CCAINTLVL0	**INTCTRLB**
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	

EEL 3744 **XMEGA** Interrupt Flags Register INTFLAGS

- INTFLAGS – Register to hold CC, Error, and Overflow/Underflow flags

	7	6	5	4	3	2	1	0	
+0x0C	CCDIF	CCCIF	CCBIF	CCAIF	-	-	ERRIF	OVFIF	**INTFLAGS**
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

EEL 3744 **XMEGA** CNT Register

- CNTL – LSB of Register Pair for CNT

	7	6	5	4	3	2	1	0	
+0x20	CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0	**CNTL**
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	

- CNTH – MSB of Register Pair for CNT

	7	6	5	4	3	2	1	0	
+0x21	CNT15	CNT14	CNT13	CNT12	CNT11	CNT10	CNT9	CNT8	**CNTH**
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	

EEL 3744

XMEGA PER Register

- PERL – LSB of Register Pair for PER

	7	6	5	4	3	2	1	0	
+0x26	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0	**PERL**
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	

- PERH – MSB of Register Pair for PER

	7	6	5	4	3	2	1	0	
+0x27	PER15	PER14	PER13	PER12	PER11	PER10	PER9	PER8	**PERH**
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	

EEL 3744

XMEGA CCx Register

- CCxL – LSB of Register Pair for CCx

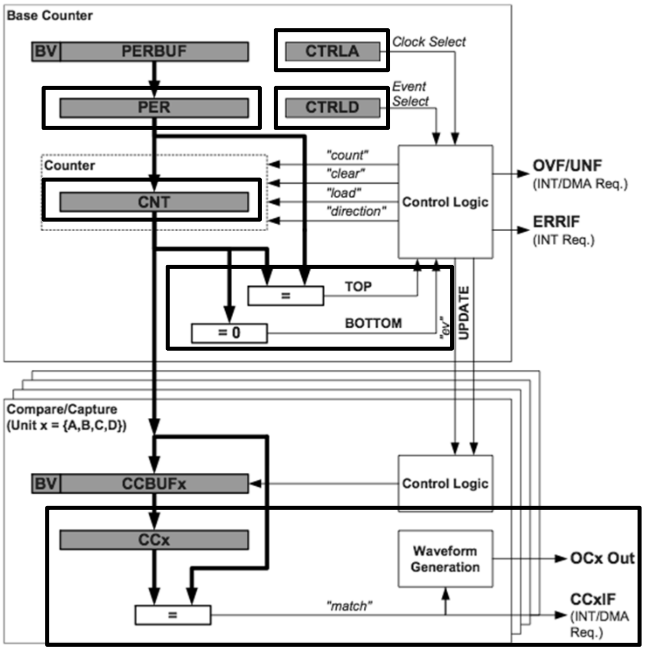
	7	6	5	4	3	2	1	0	
	CCx7	CCx6	CCx5	CCx4	CCx3	CCx2	CCx1	CCx0	**CCxL**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- CCxH – MSB of Register Pair for CCx

	7	6	5	4	3	2	1	0	
	CCx15	CCx14	CCx13	CCx12	CCx11	CCx10	CCx9	CCx8	**CCxH**
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	

EEL 3744

XMEGA Timer System Block Diagram



doc8331, Figure 14-2

EEL 3744

XMEGA Timer0/1 Summary for Lab 6

- For PWM (use Frequency Generation Mode to toggle output pin)
  - > Set pin as output
  - > No interrupt needed if 50% duty cycle
  - > *TC freq* (Hz) = f_{clk} / 2^x (for x=0,1,..3,6,8,10)
    - – Use **TCF0_CTRLA** to set 2^x above; f_{clk} = 2MHz (for our board config)
  - > *TCF1_CCA_VAL* (cycles) = *Toggle period* (s) * *TC Freq* (Hz)
  - > **TCF1_CTRLB**
    - – Enable CCA (or CCB, CCC, or CCD)
    - – Set WGMode to frequency
  - > Set **TCF1_CCA** to *TCF1_CCA_VAL*
    - – Remember that you must store the least significant byte of 2 bytes first
  - > No interrupt necessary

EEL 3744

XMEGA Timer0/1 Summary for Lab 6

- For general timer (more accurate version of RTC)
  - > If necessary, set pin as output
  - > Overflow interrupt vector: **TCF0_OVF_VECT**
  - > *TC freq* (Hz) = f_{clk} / 2^x (for x=0,1,..3,6,8,10)
    - – Use **TCF0_CTRLA** to set 2^x above
    - – f_{clk} = 2MHz (for our board in default configuration)
  - > *TCF0_PERIOD* (cycles) = *Period* (s) * *TC freq* (Hz)
  - > **TCF0_INTCTRLA**, **PMIC_CTRL**
    - – Interrupt to do something every time the timer runs out, e.g.,
      - ☞Toggle pin
      - ☞Run ADC
      - ☞Check UART or LCD
  - > Set **TCF0_PER** to *TCF0_PERIOD*
    - – Remember that you must store the least significant byte of 2 bytes first

Step through Timer_OC.asm

EEL 3744

The End!