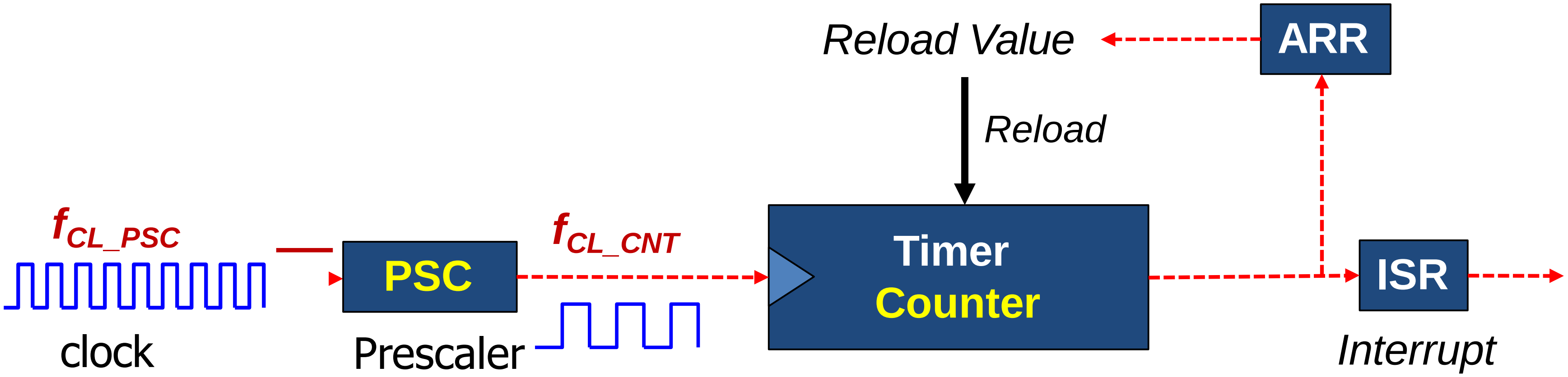
Timer

- Free-run counter (independent of processor)
- Functions
  - **Input capture**
  - Output compare
  - Pulse-width modulation (PWM) generation
  - One-pulse mode output
- STM has many application notes (on all aspects of the STM32)
  - App note AN4776 – General Purpose Timer Cookbook

http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

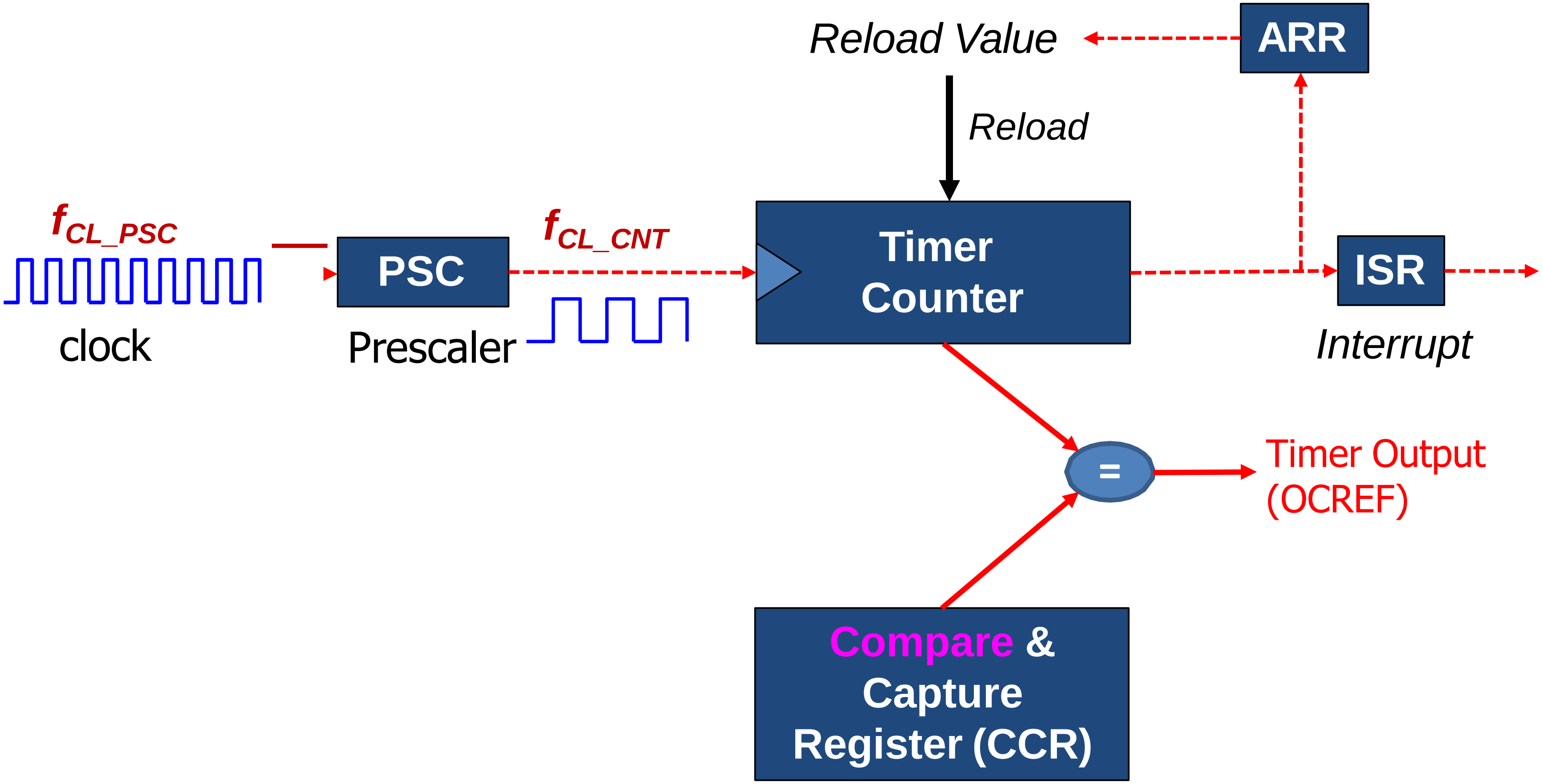
Timer: Clock



$$f_{CK\_CNT} = \frac{f_{CL\_PSC}}{PSC + 1}$$

http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

Timer: Output



f_{CL_PSC}

clock

PSC

Prescaler

f_{CL_CNT}

Timer Counter

Reload Value

Reload

ARR

ISR

Interrupt

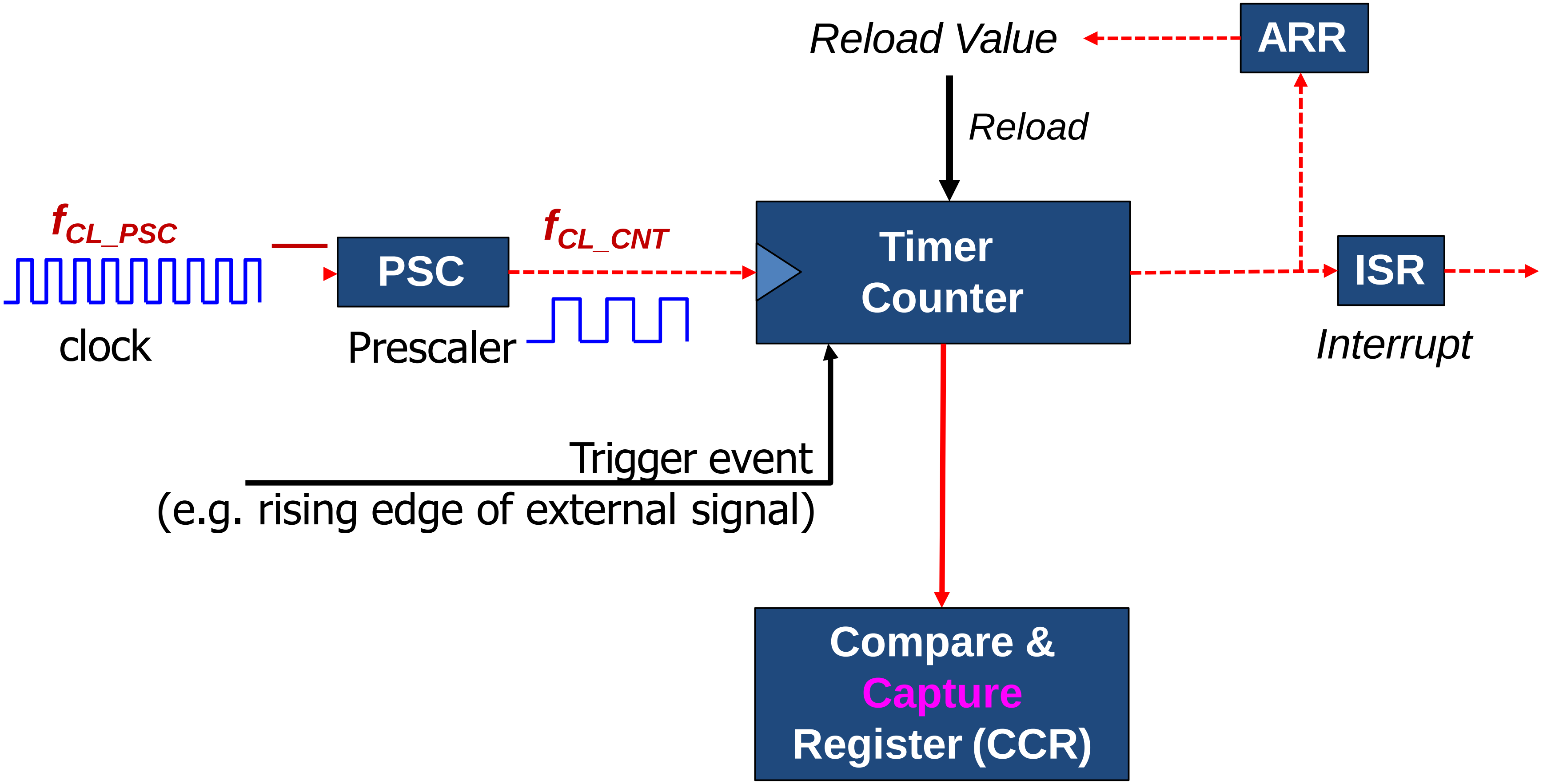
=

Timer Output (OCREF)

Compare & Capture Register (CCR)

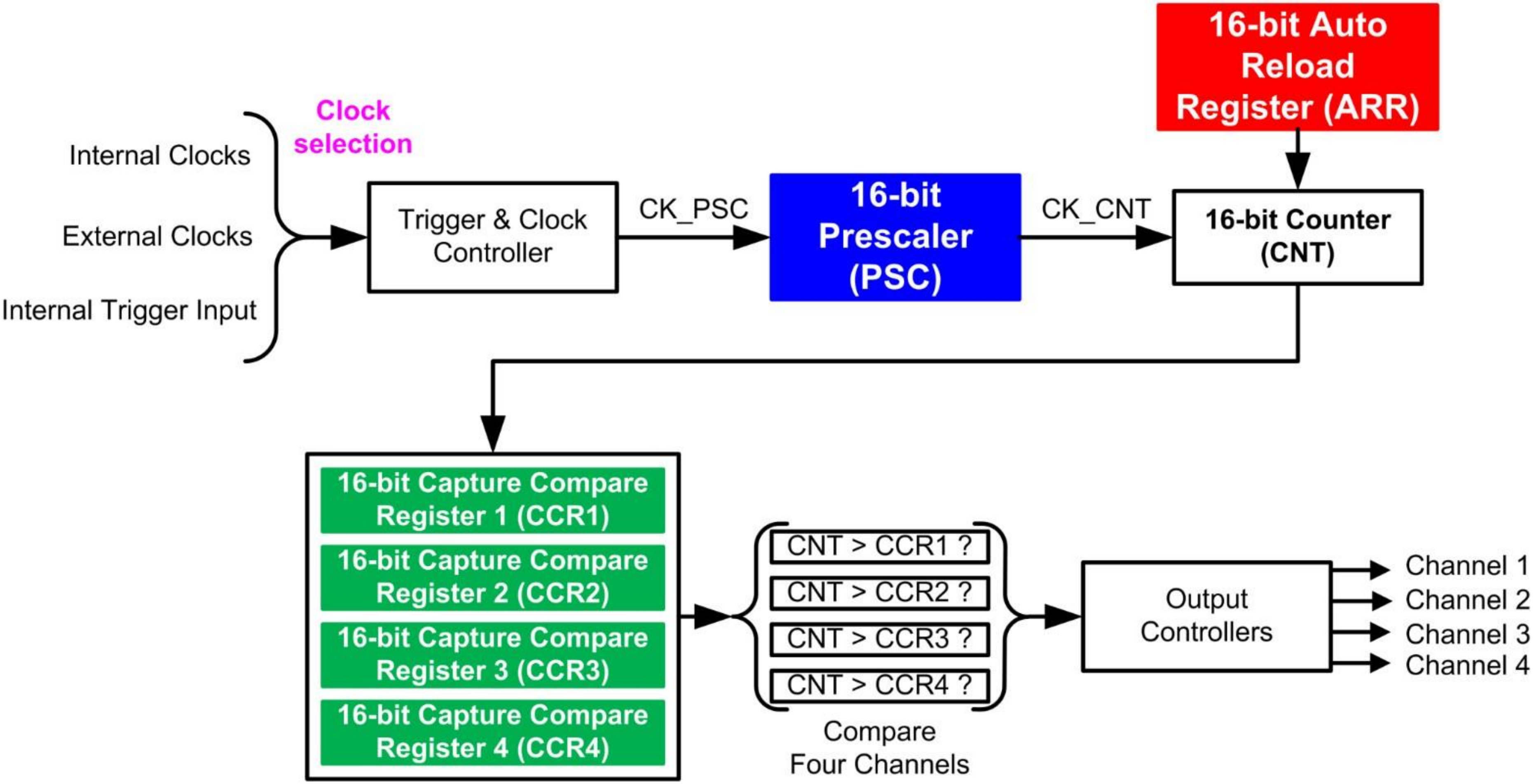
http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

Timer: Input Capture



http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

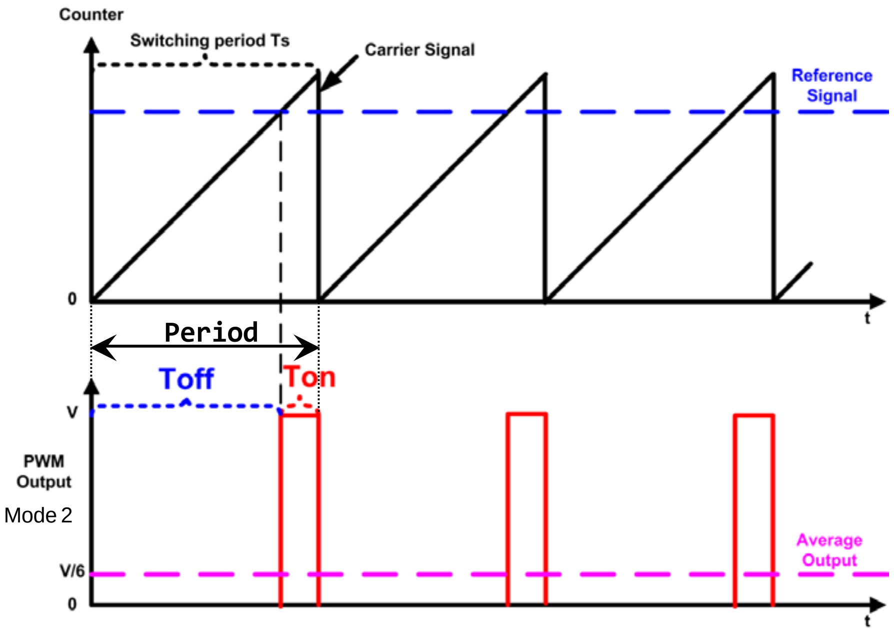
Multi-Channel Outputs



http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

PWM Mode

(Pulse Width Modulation)



$$\text{Period} = T_{off} + T_{on}$$

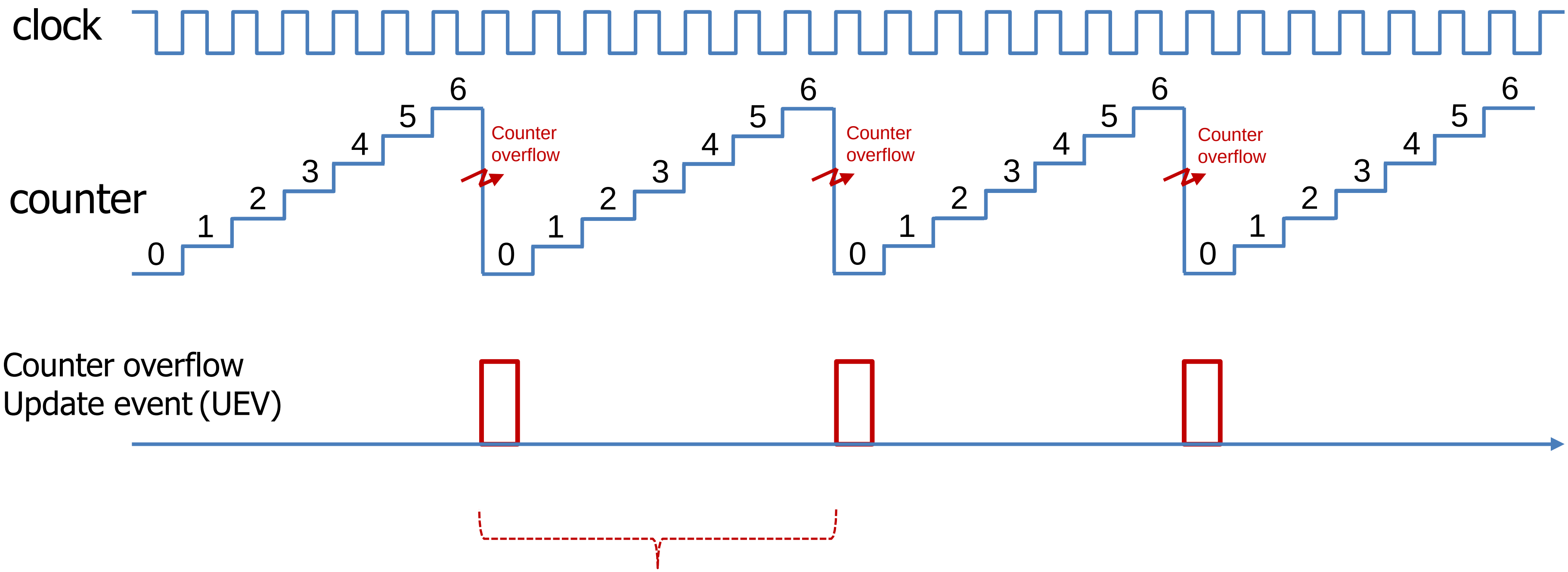
$$\text{Duty Cycle} = \frac{\mathbb{C}_{ooo\qquad o}}{CC_{ooooo}+CC_{oooo}}$$

Mode	Counter < Reference	Counter ≥ Reference
PWM mode 1 (Low True)	Active Low	Inactive
PWM mode 2 (High True)	**Inactive**	**Active High**

http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

Edge-aligned Mode (Up-counting)

ARR = 6, RCR = 0



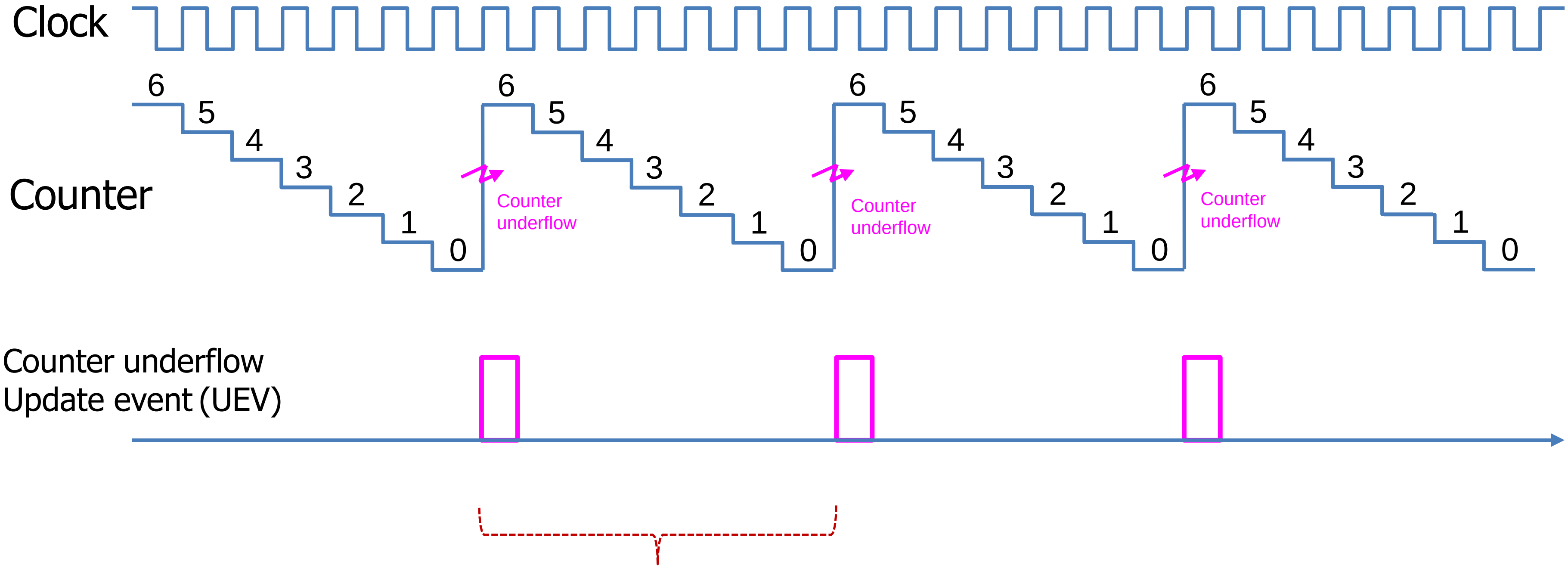
```
Period = (1 + ARR) * Clock Period
       = 7 * Clock Period
```

```
ARR = Auto-Reload Register
UEV = Update Event
RCR = Repetition Count Register
```

http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

Edge-aligned Mode (down-counting)

ARR = 6, RCR = 0

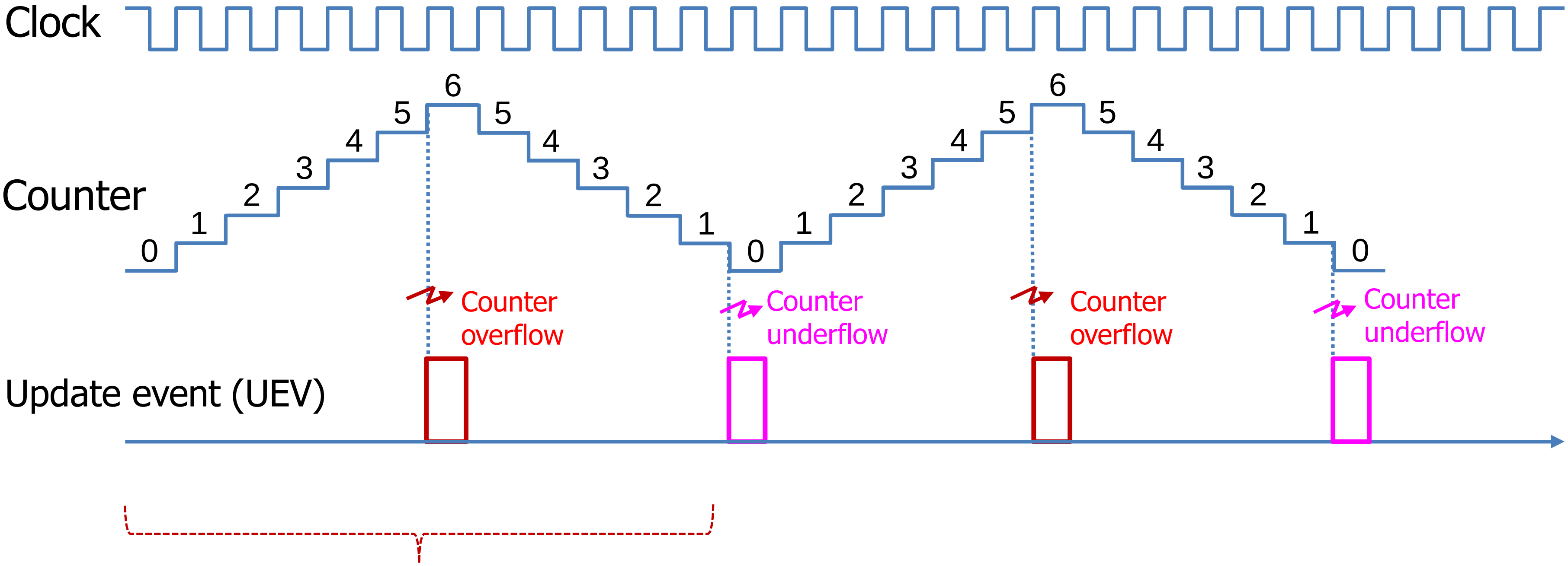


```
Period = (1 + ARR) * Clock Period
       = 7 * Clock Period
```

http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

Center-aligned Mode

ARR = 6, RCR = 0



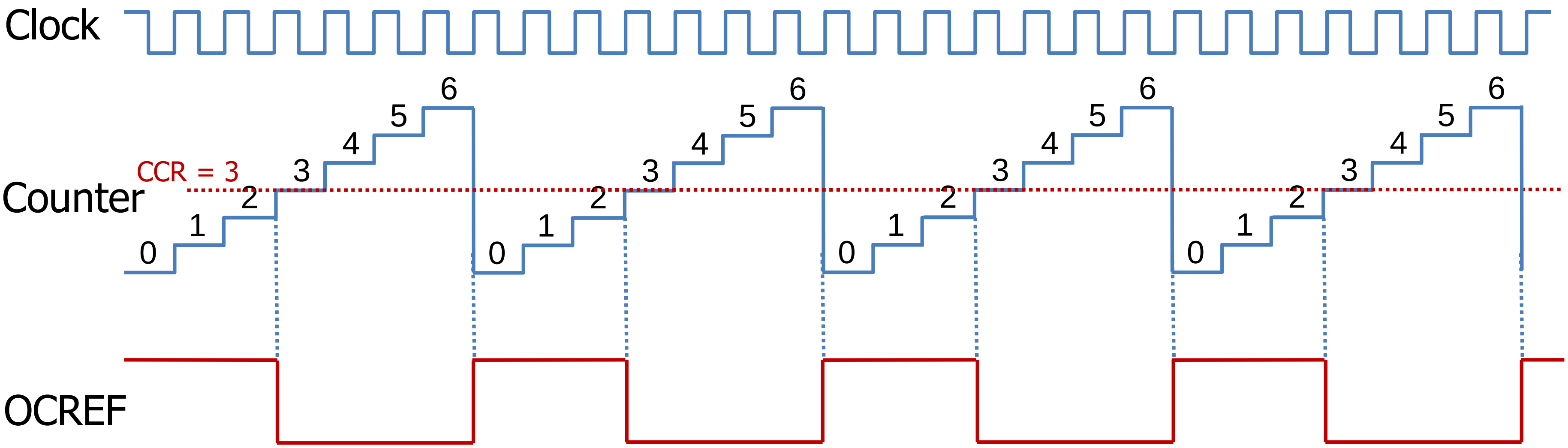
```
Period = (2 * ARR) * Clock Period
       = 12 * Clock Period
```

http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

PWM
Mode 1 (Low-True)

Timer Output = { High if counter < CCR Low if counter ≥ CCR

Upcounting mode, ARR = 6, CCR = 3, RCR = 0



$$\text{Duty Cycle} = \frac{\text{CCR}}{\text{ARR} + 1} = \frac{3}{7}$$

$$\text{Period} = (1 + \text{ARR}) * \text{Clock Period} = 7 * \text{Clock Period}$$

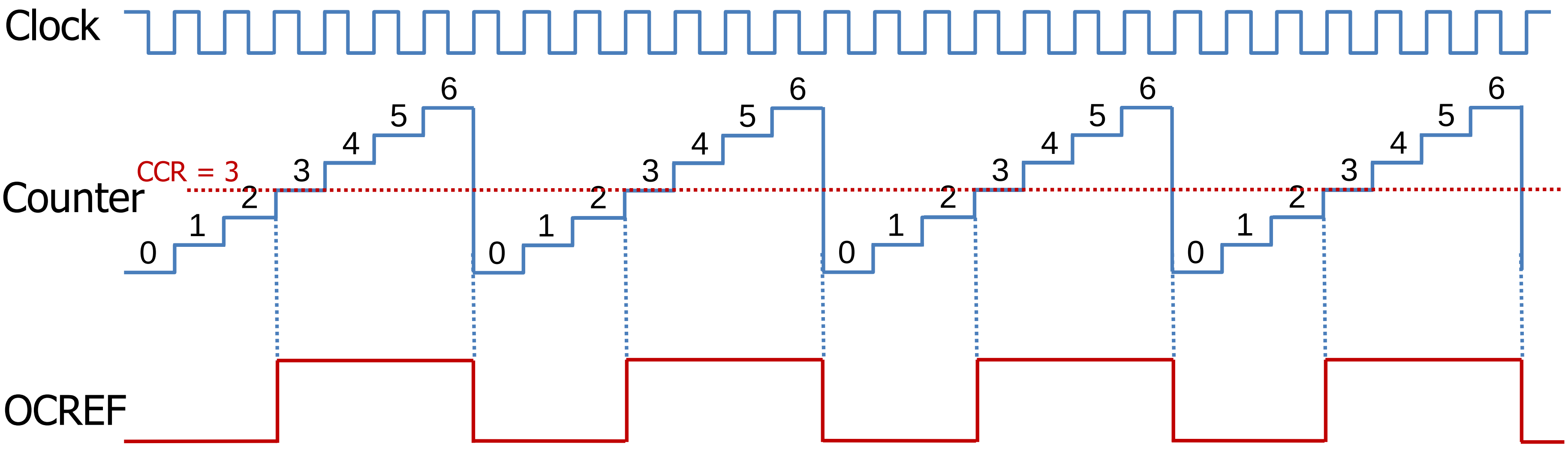
http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

PWM

Mode 2 (High-True)

Timer Output = { Low if counter < CCR High if counter ≥ CCR

Upcounting mode, ARR = 6, CCR = 3, RCR = 0



$$\text{Duty Cycle} = 1 - \frac{CCR}{ARR + 1}$$

$$= \frac{4}{7}$$

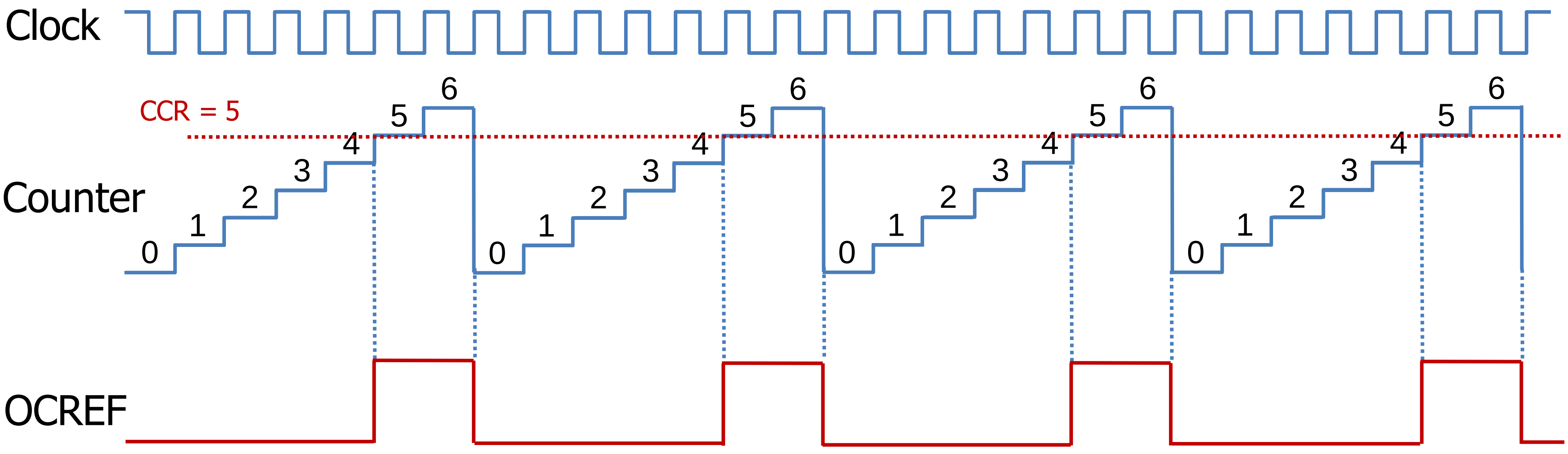
Period = (1 + ARR) * Clock Period
= 7 * Clock Period

http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

PWM
Mode 2 (High-True)

Timer Output = {
Low if counter <
CCR High if
counter ≥ CCR

Upcounting mode, ARR = 6, CCR = 3, RCR = 0



$$\text{Duty Cycle} = 1 - \frac{\text{CCR}}{\text{ARR} + 1}$$

$$= \frac{2}{7}$$

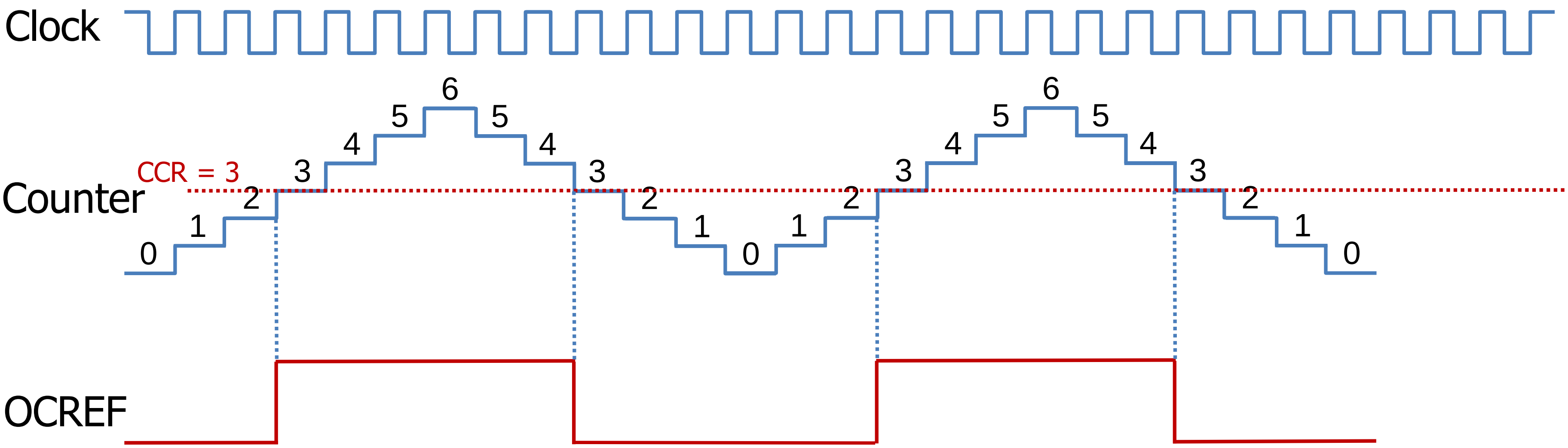
Period = (1 + ARR) * Clock Period
= 7 * Clock Period

http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

PWM
Mode 2 (High-True)

Timer Output = { Low if counter < CCR High if counter ≥ CCR

Center-aligned mode, ARR = 6, CCR = 3, RCR = 0



Duty Cycle = $1 - \frac{CCR}{ARR}$

$= \frac{1}{2}$

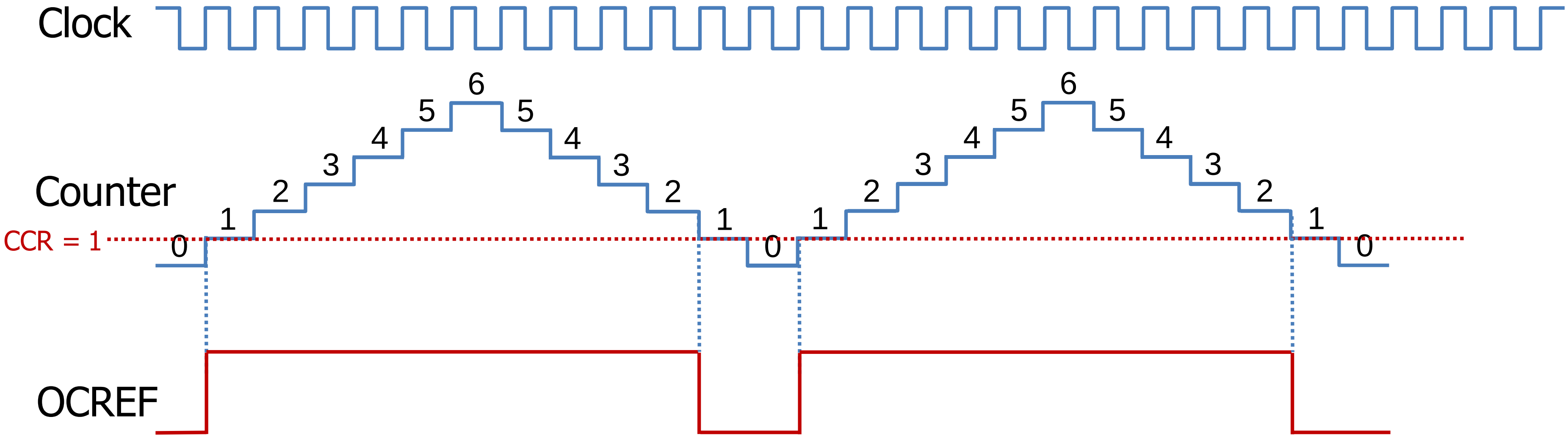
Period = 2 * ARR * Clock Period
= 12 * Clock Period

http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

PWM
Mode 2 (High-True)

Timer Output = { Low if counter <
CCR High if
counter ≥ CCR

Center-aligned mode, ARR = 6, CCR = 3, RCR = 0



$$\text{Duty Cycle} = 1 - \frac{\text{CCR}}{\text{ARR}}$$
$$= \frac{5}{6}$$

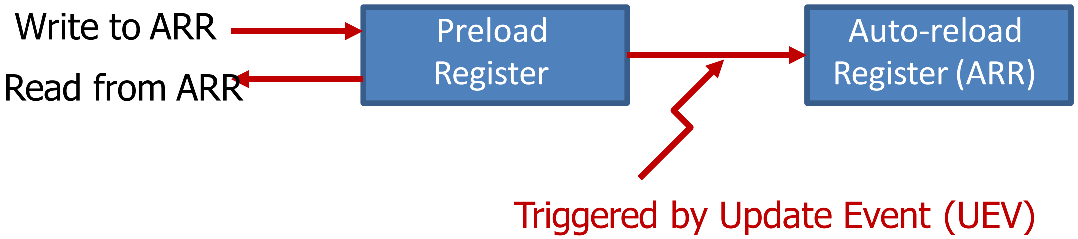
Period = 2 * ARR * Clock Period
= 12 * Clock Period

http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

Auto-Reload Register (ARR)

- Auto-Reload Preload Enable (ARPE) bit in TIMx_CR1

ARPE = 1 (Synchronous Update)



Write to ARR

Read from ARR

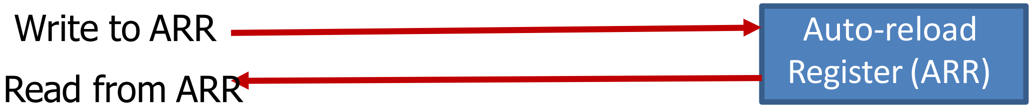
Preload Register

Auto-reload Register (ARR)

Triggered by Update Event (UEV)

If UDIS bit in TIMx_CR1 is 1, UEV event is disabled.

ARPE = 0 (Asynchronous Update)



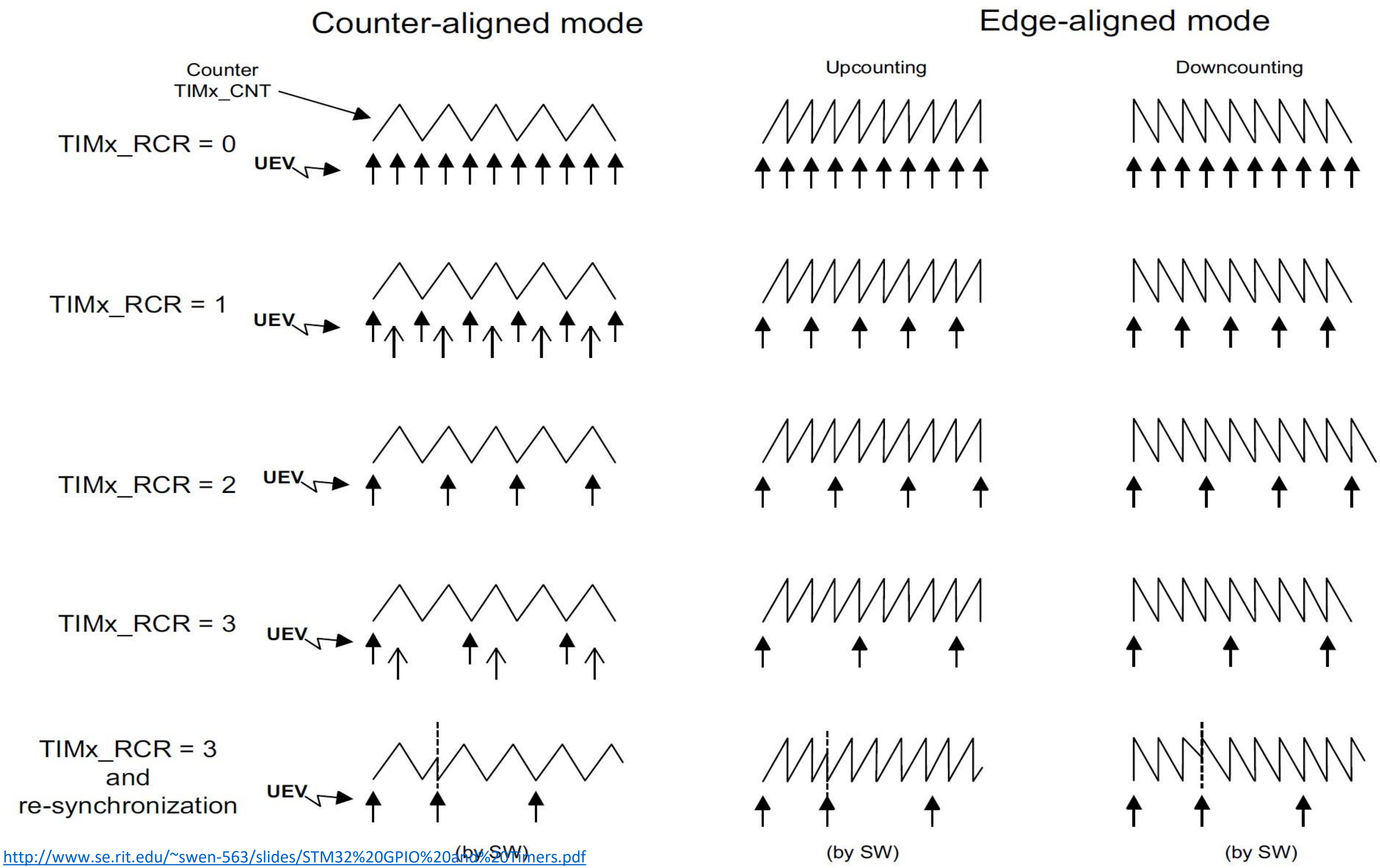
Write to ARR

Read from ARR

Auto-reload Register (ARR)

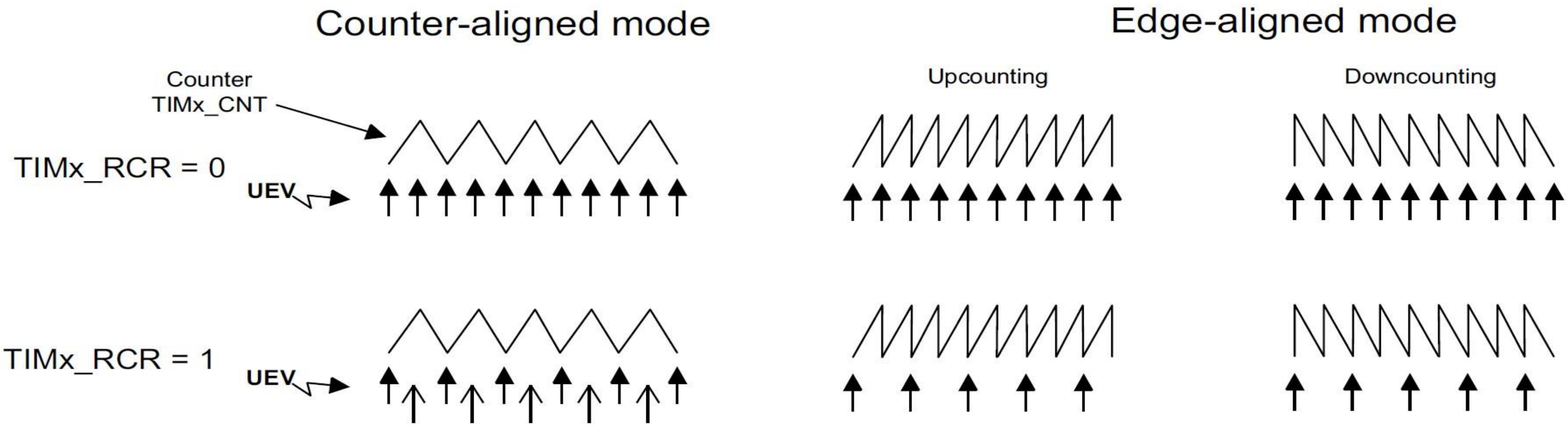
http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

Repetition Counter Register (RCR)



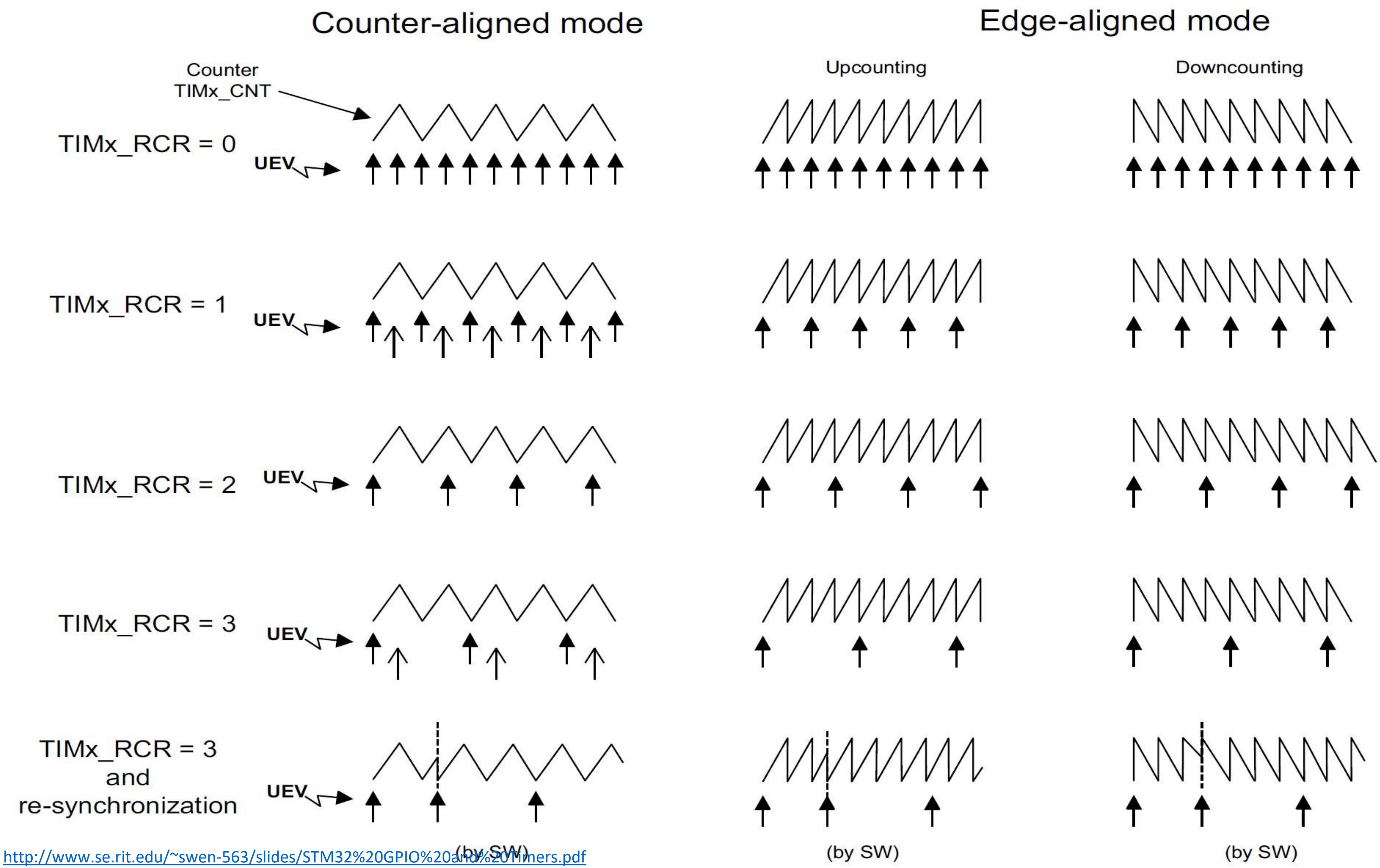
http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

Repetition Counter Register (RCR)



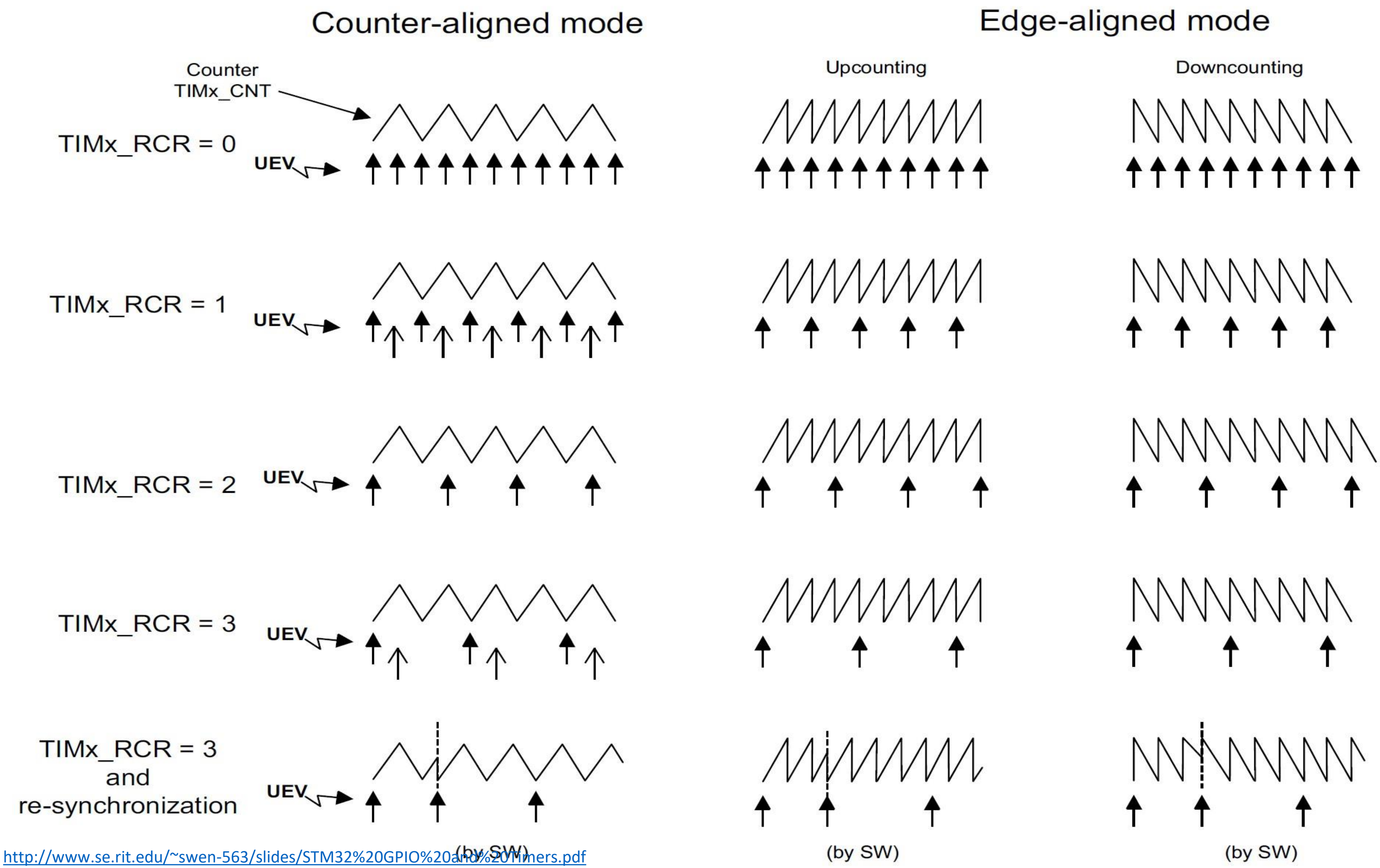
http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

Repetition Counter Register (RCR)



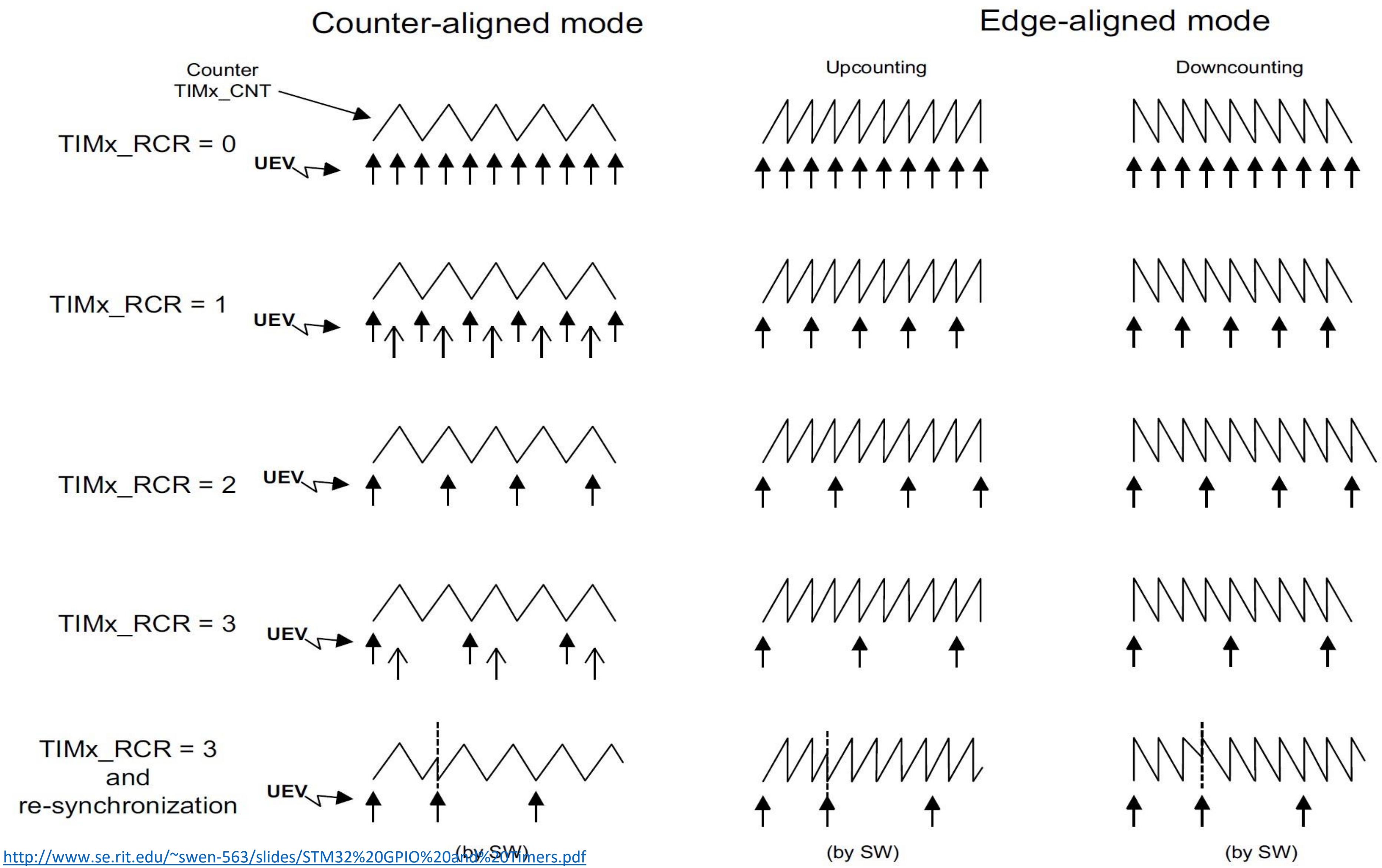
http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

Repetition Counter Register (PCR)



http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

Repetition Counter Register (PCR)



http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

PWM Output Polarity

Mode	Counter < CCR	Counter ≥ CCR
PWM mode 1 (Low True)	Active (Low)	Inactive
PWM mode 2 (High True)	Inactive	Active (High)

Output Polarity:

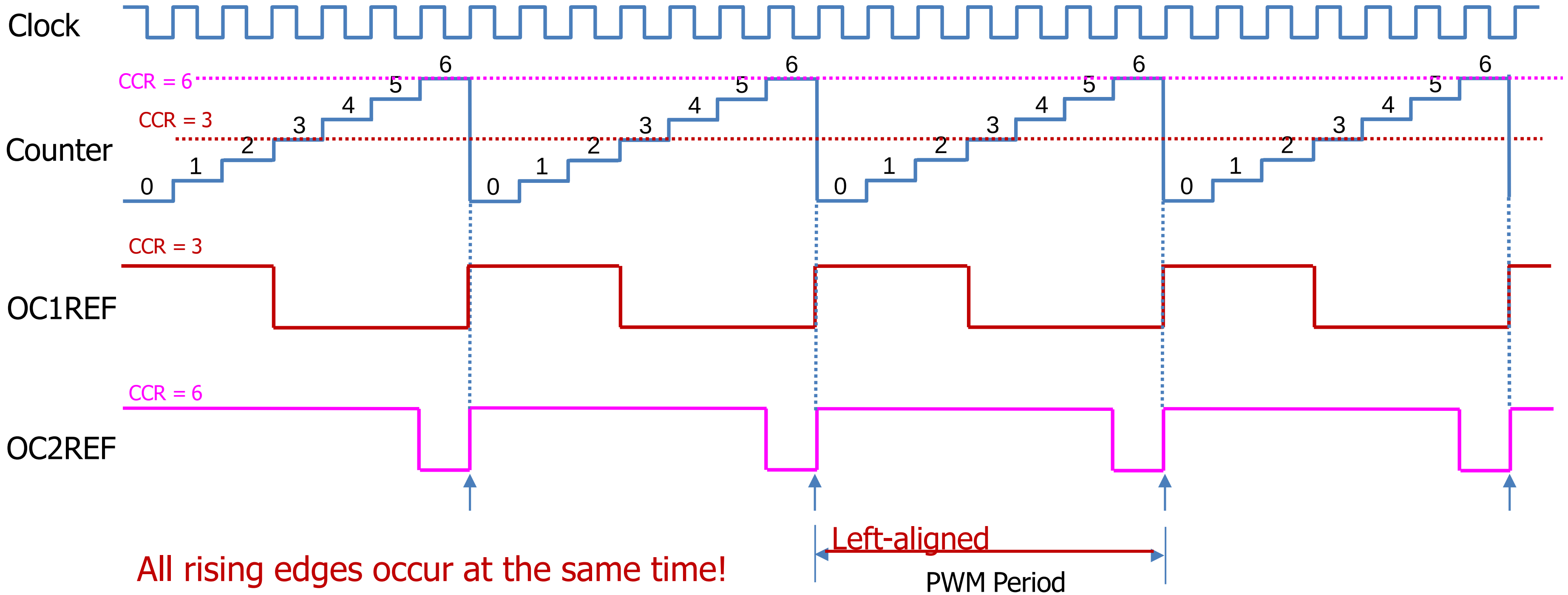
- Software can program the CCxP bit in the TIMx_CCER register

	Active	Inactive
Active High	High Voltage	Low Voltage
Active Low	Low Voltage	High Voltage

http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

Up-Counting: Left Edge-aligned

Upcounting mode, ARR = 6, CCR = 3, RCR = 0

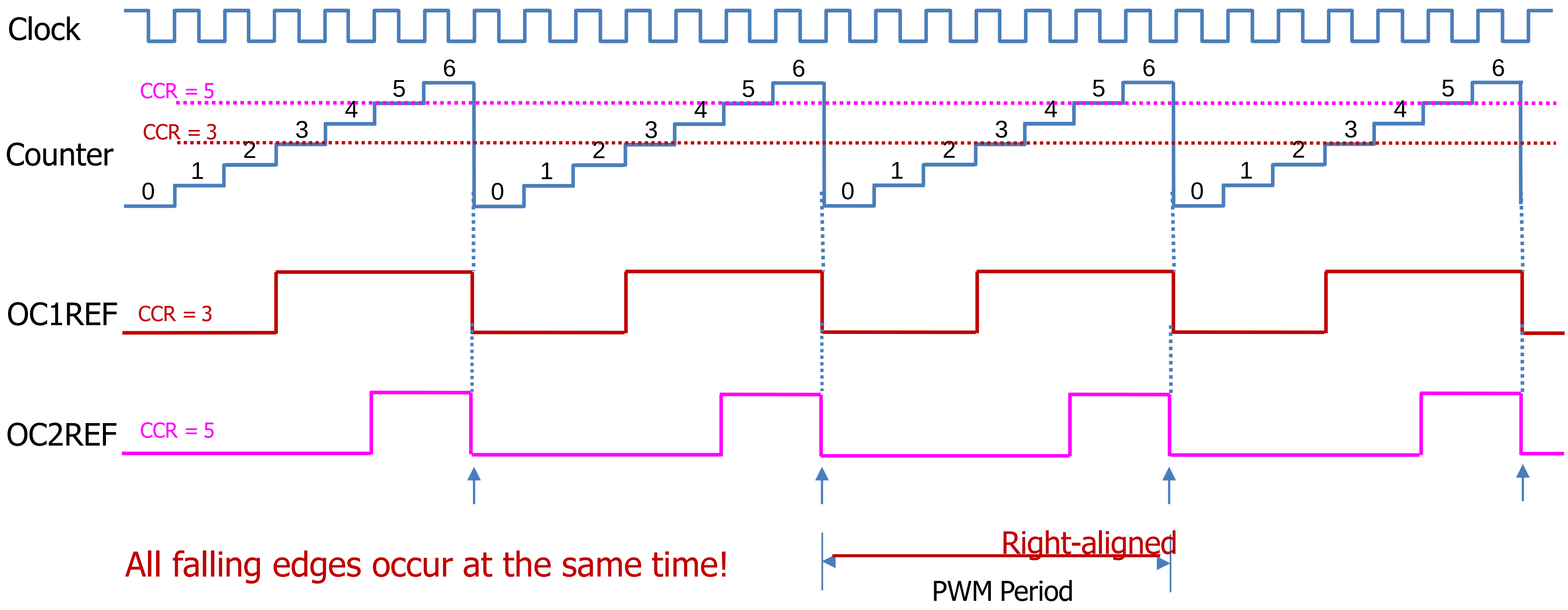


All rising edges occur at the same time!

http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

PWM Mode 2: Right Edge-aligned

Upcounting mode, ARR = 6, CCR = 3, RCR = 0

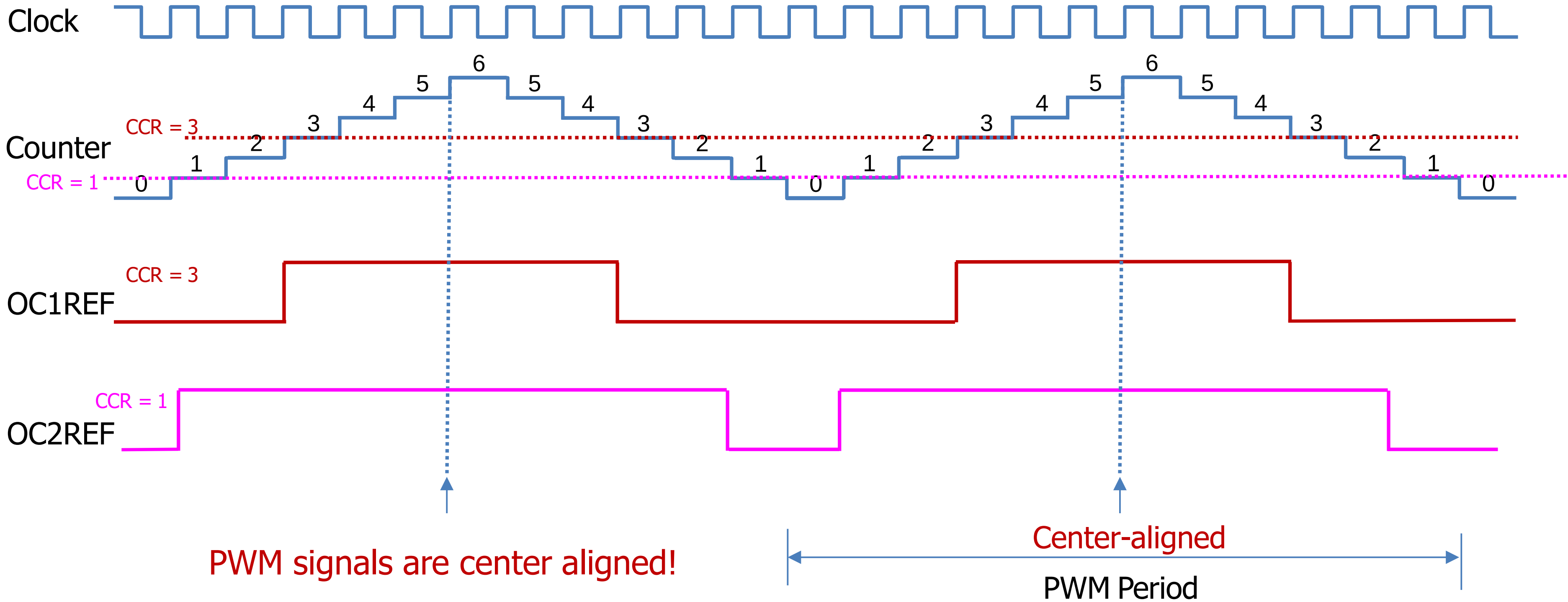


All falling edges occur at the same time!

http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

PWM Mode 2: Center Aligned

Center-aligned mode, ARR = 6, CCR = 3, RCR = 0



PWM signals are center aligned!

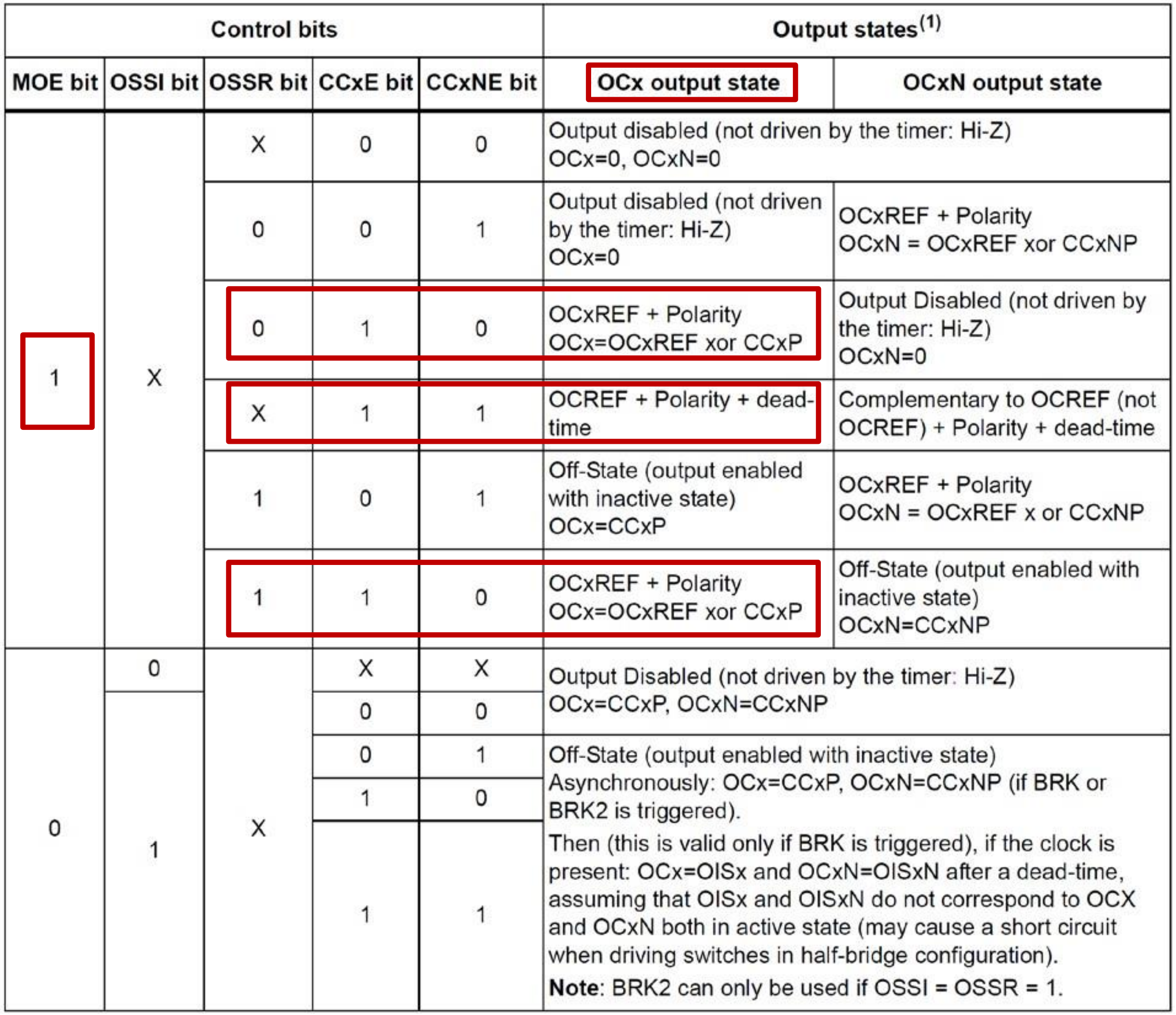
Center-aligned

PWM Period

http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

The devil is in the detail

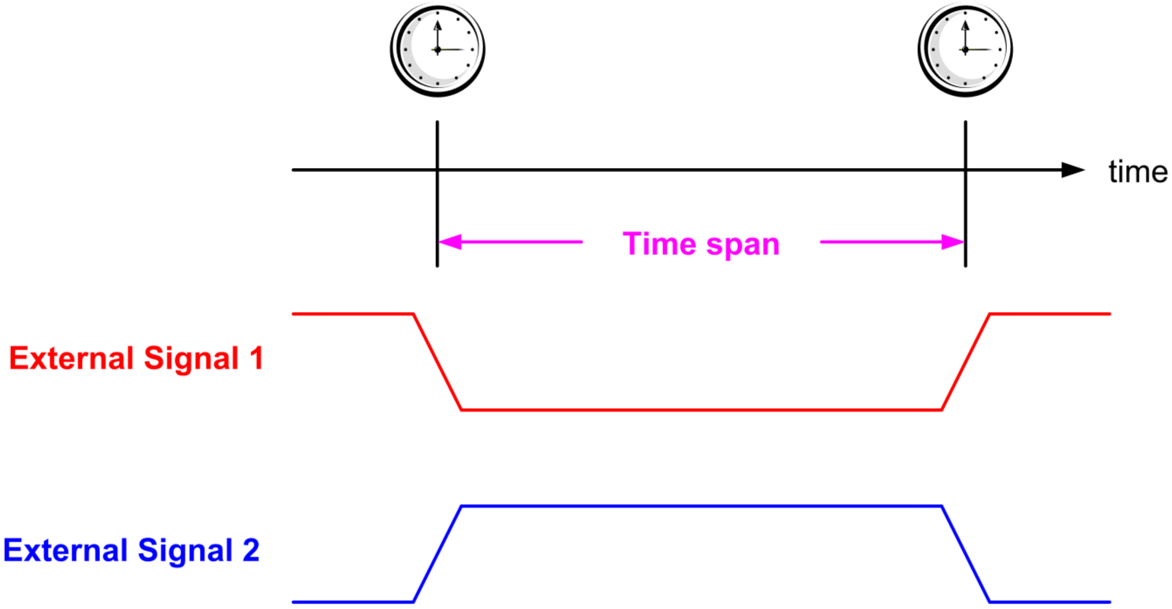
- Timer output control
- Enable Timer Output
  - MOE: Main output enable
  - OSSI: Off-state selection for Idle mode
  - OSSR: Off-state selection for Run mode
  - CCxE: Enable of capture/compare output for channel x
  - CCxNE: Enable of capture/compare complementary output for channel x



Control bits					Output states(1)	
MOE bit	OSSI bit	OSSR bit	CCxE bit	CCxNE bit	OCx output state	OCxN output state
1	X	X	0	0	Output disabled (not driven by the timer: Hi-Z) OCx=0, OCxN=0	
		0	0	1	Output disabled (not driven by the timer: Hi-Z) OCx=0	OCxREF + Polarity OCxN = OCxREF xor CCxNP
		0	1	0	OCxREF + Polarity OCx=OCxREF xor CCxP	Output Disabled (not driven by the timer: Hi-Z) OCxN=0
		X	1	1	OCREF + Polarity + dead-time	Complementary to OCREF (not OCREF) + Polarity + dead-time
		1	0	1	Off-State (output enabled with inactive state) OCx=CCxP	OCxREF + Polarity OCxN = OCxREF x or CCxNP
		1	1	0	OCxREF + Polarity OCx=OCxREF xor CCxP	Off-State (output enabled with inactive state) OCxN=CCxNP
0	0	X	X	X	Output Disabled (not driven by the timer: Hi-Z) OCx=CCxP, OCxN=CCxNP	
	1		0	0		
			0	1	Off-State (output enabled with inactive state) Asynchronously: OCx=CCxP, OCxN=CCxNP (if BRK or BRK2 is triggered). Then (this is valid only if BRK is triggered), if the clock is present: OCx=OISx and OCxN=OISxN after a dead-time, assuming that OISx and OISxN do not correspond to OCX and OCxN both in active state (may cause a short circuit when driving switches in half-bridge configuration). **Note**: BRK2 can only be used if OSSI = OSSR = 1.	
			1	0		
			1	1		

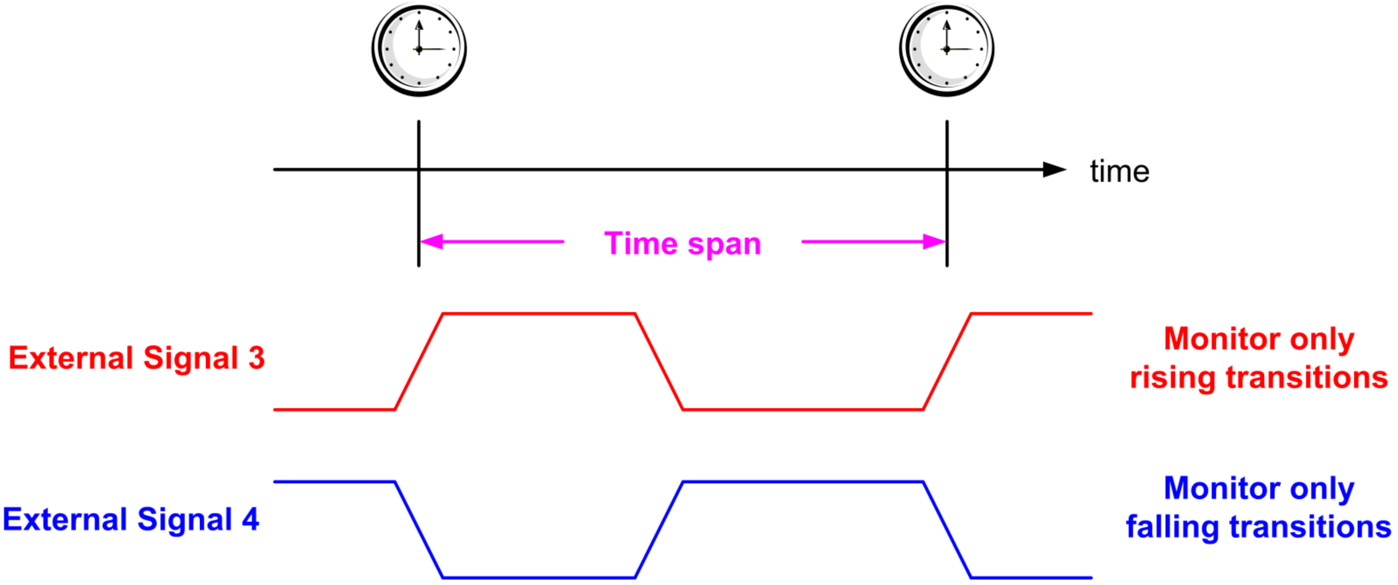
http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

- Monitor both rising and falling edge



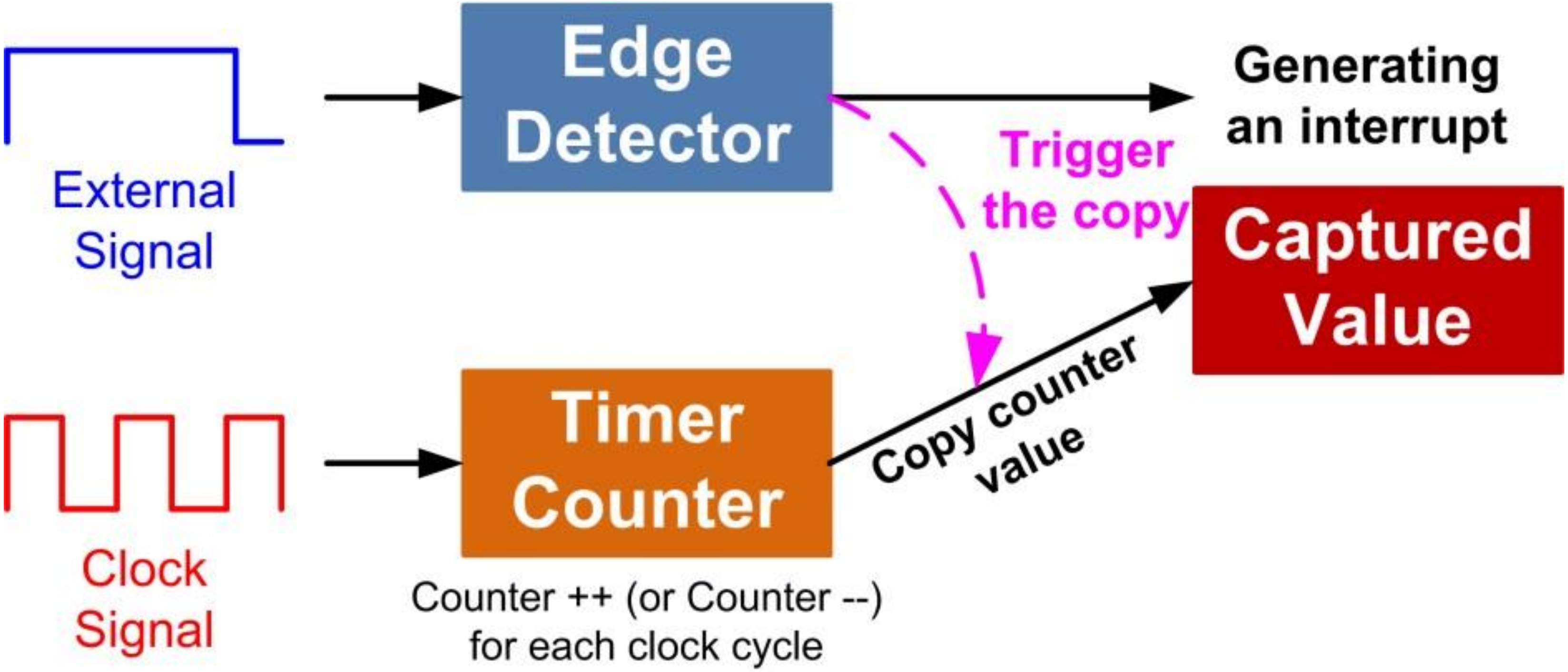
http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

- Monitor only rising edges or only falling edge


time
Time span
External Signal 3
Monitor only rising transitions
External Signal 4
Monitor only falling transitions

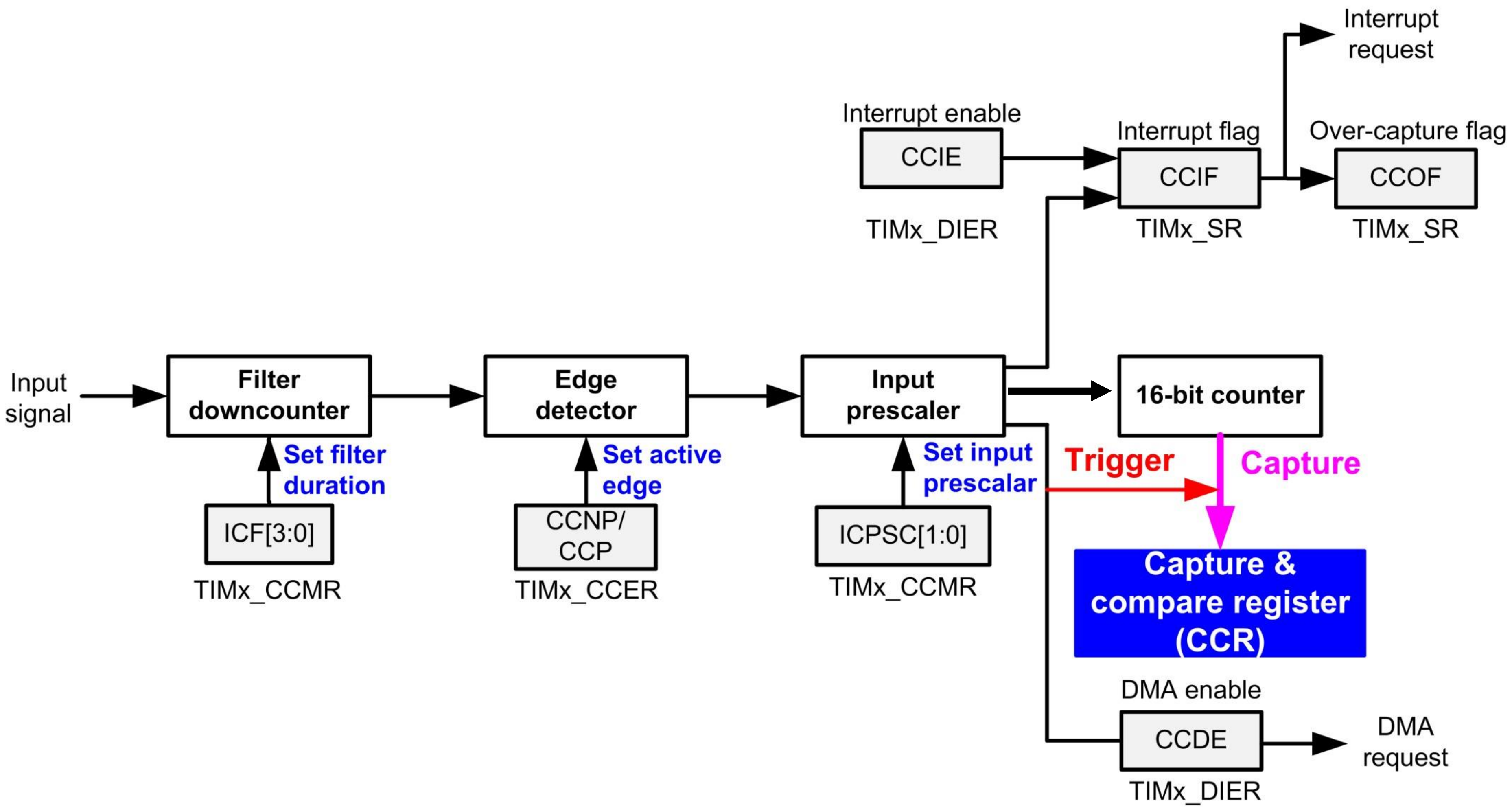
http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

Input Capture



http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

Input Capture Diagram

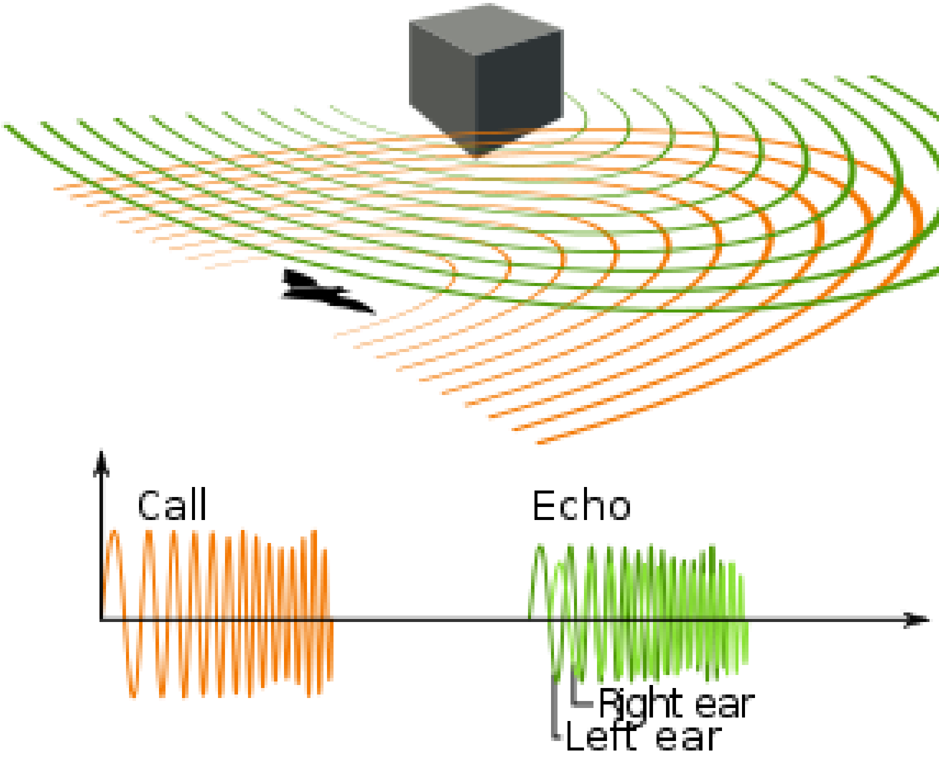


http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

WHY IS THE MEASUREMENT OF TIME INTERESTING???

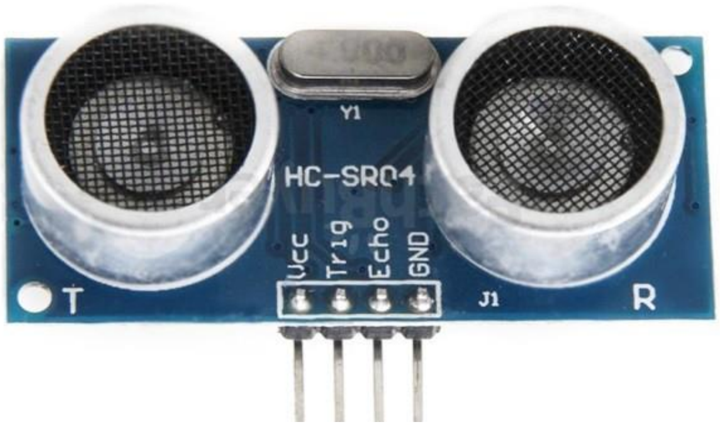
http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

use echolocation to map their surroundings?



http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

Ultrasonic Distance Sensor



$$Distance = \frac{Round\ Trip\ Time \times Speed\ of\ Sound}{2}$$

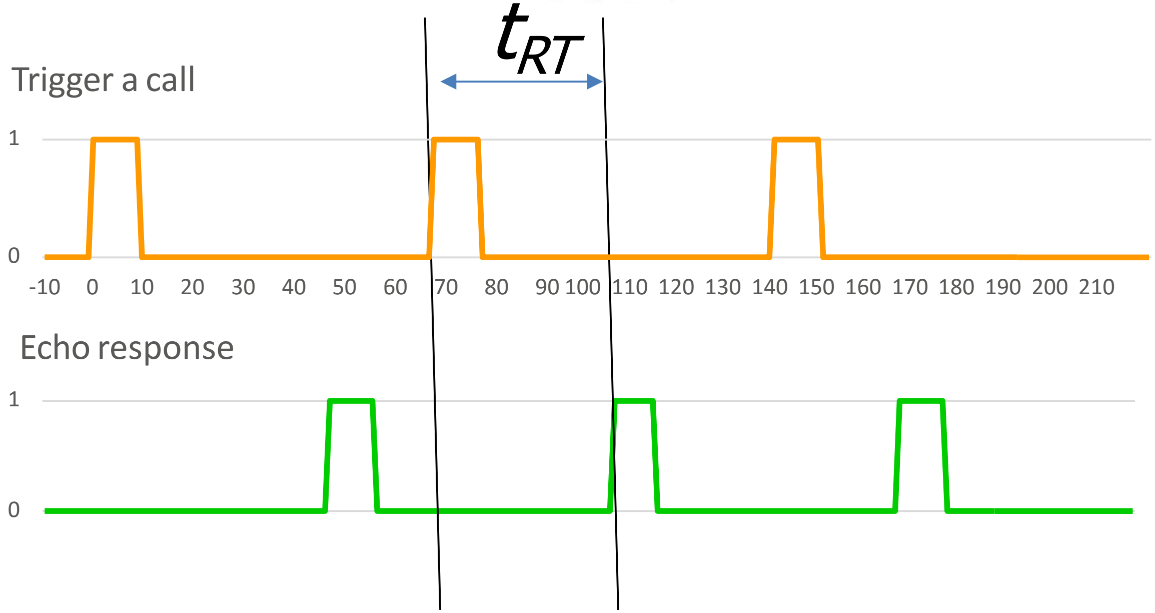
$$= \frac{Round\ Trip\ Time(\mu s) \times 10^{-6} \times 340 m/s}{2}$$

$$= \frac{Round\ Trip\ Time(\mu s)}{58}$$

http://www.se.rit.e[illegible]

Ultrasonic Distance Sensor

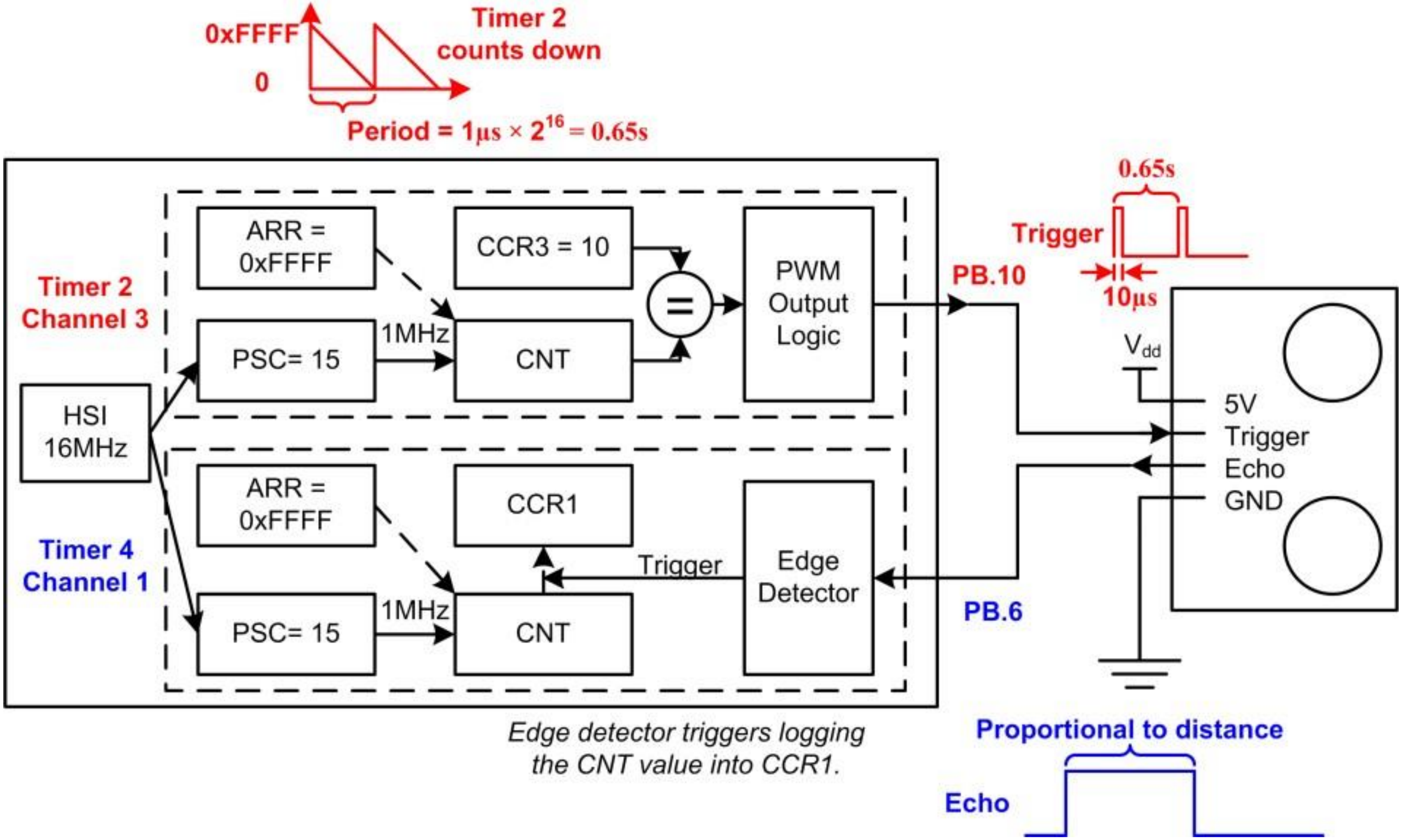
- The delay from triggered chirp to echo response is the round-trip time t_{RT}.



If $t_{RT >}$ is 60*ms*, no obstacle is detected.

http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf

Ultrasonic Distance Sensor



Edge detector triggers logging the CNT value into CCR1.

http://www.se.rit.edu/~swen-563/slides/STM32%20GPIO%20and%20Timers.pdf