Embedded Systems with ARM Cortex-M Microcontrollers
in Assembly Language and C

Chapter 15
General-purpose Timers

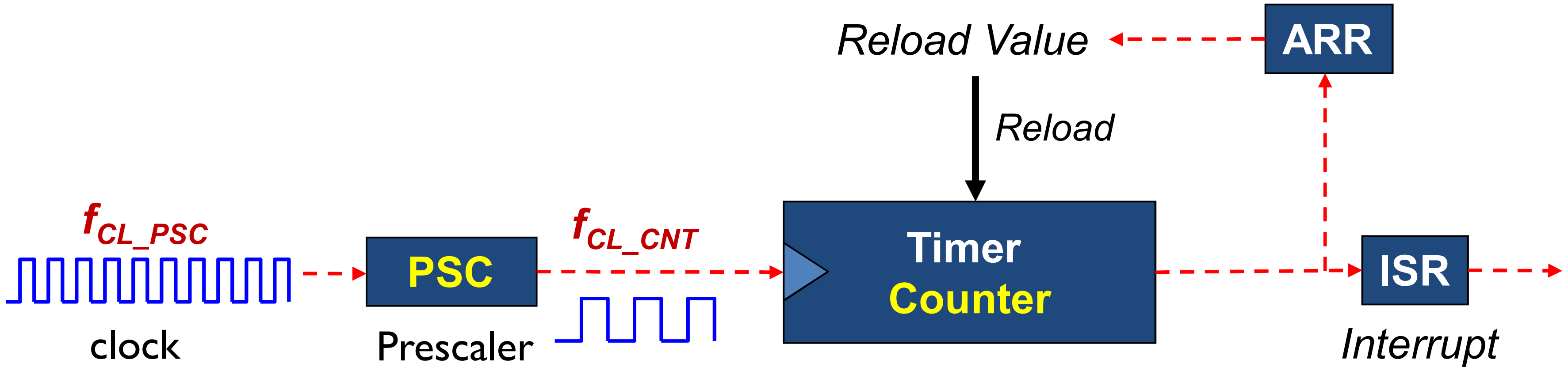
Dr. Yifeng Zhu
ECE, University of Maine
With corrections by Prof. Mark Lawford
McMaster University

Fall 2017

Timer

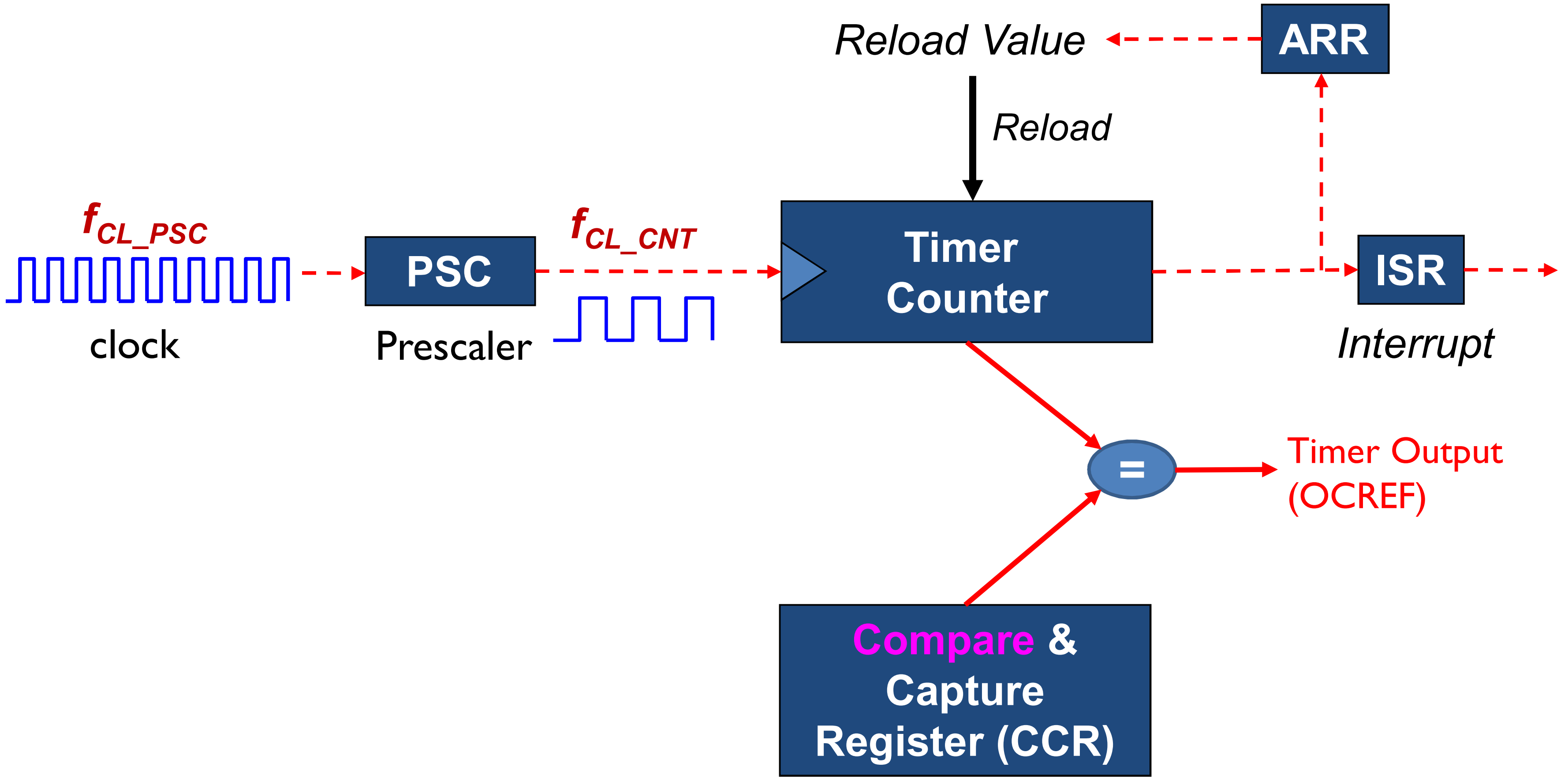
- Free-run counter (independent of processor)
- Functions
  - Input capture
  - Output compare
  - Pulse-width modulation (PWM) generation
  - One-pulse mode output

Timer: Clock

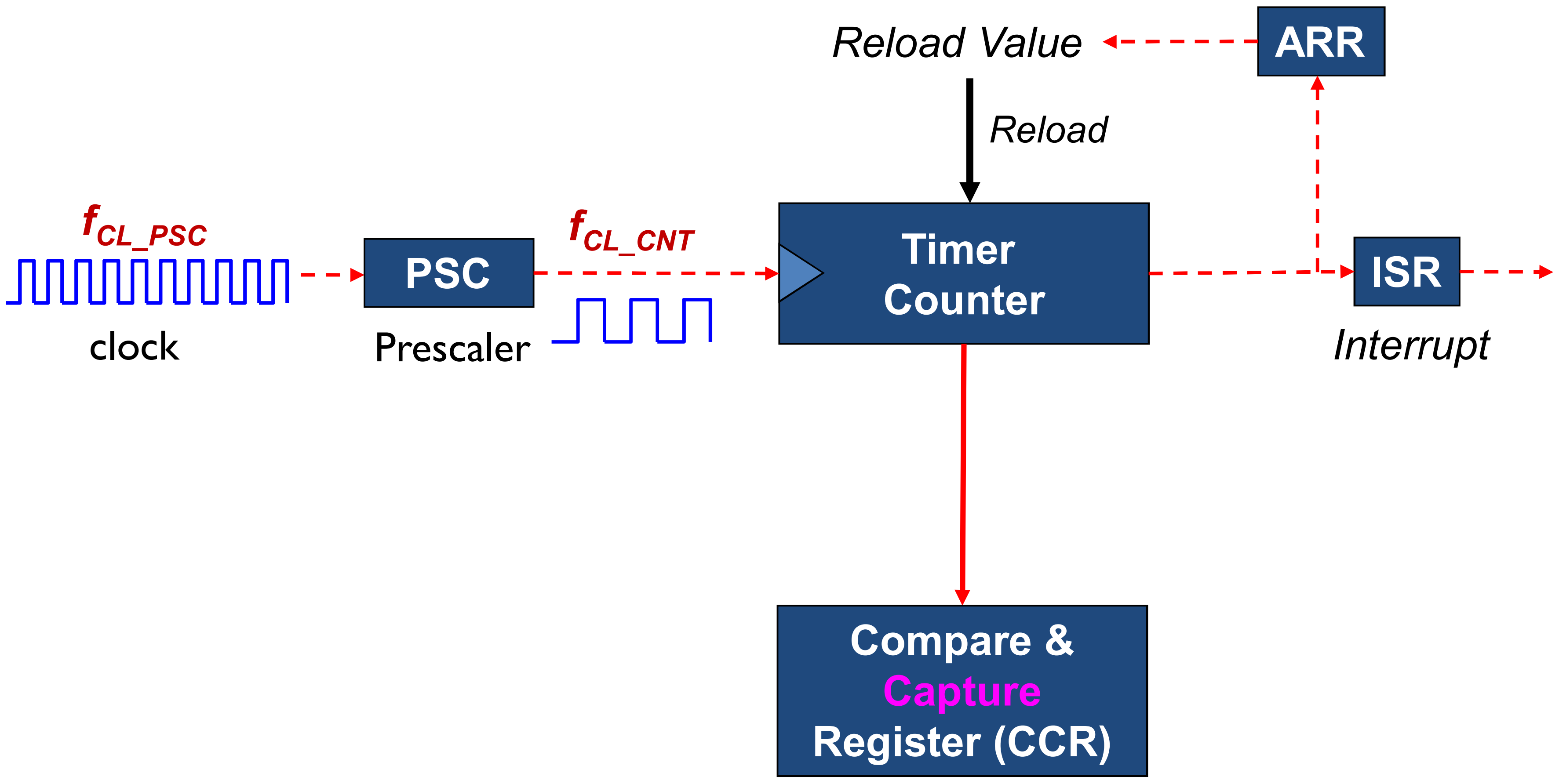


$$f_{CK\_CNT} = \frac{f_{CL\_PSC}}{PSC + 1}$$

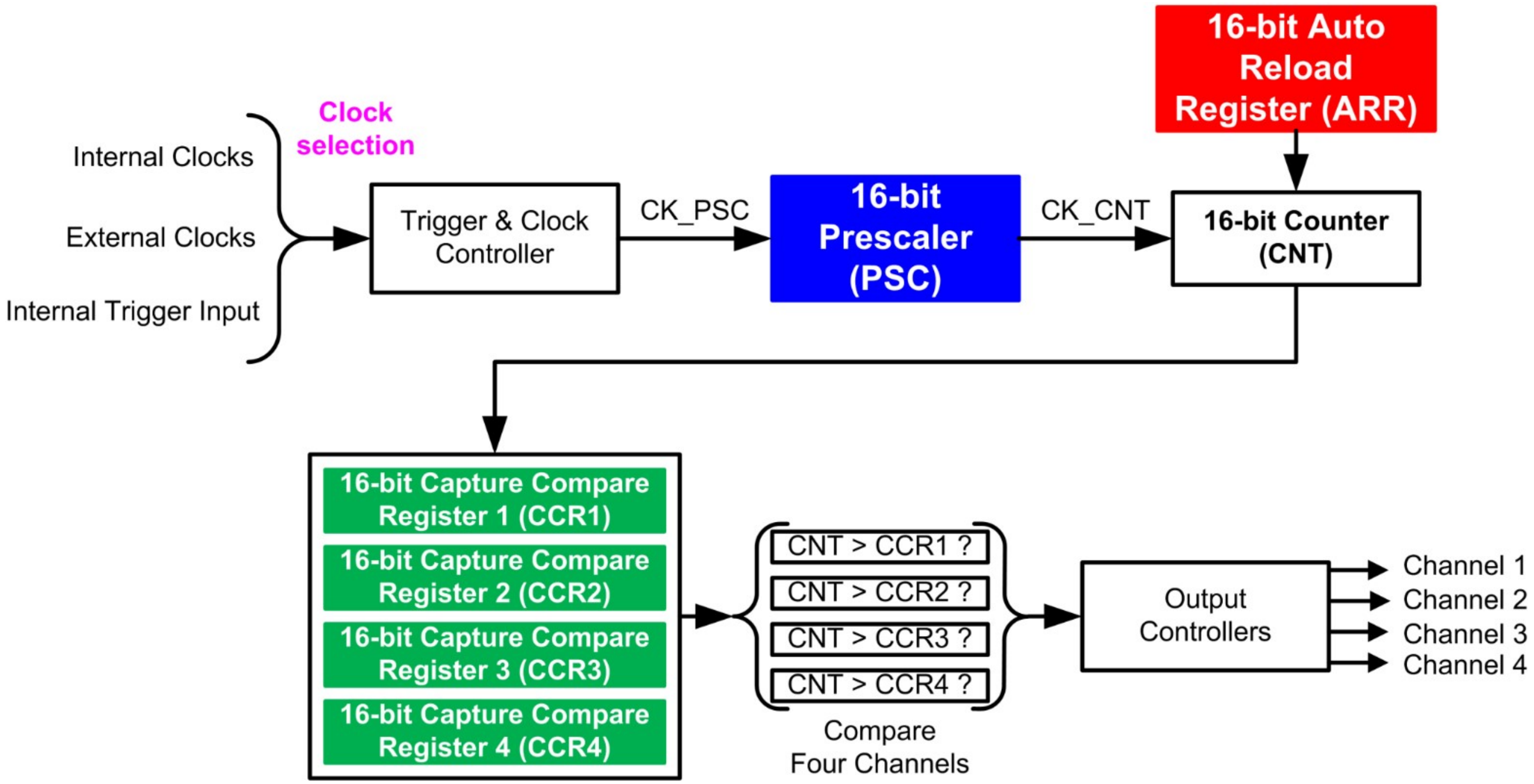
Timer: Output



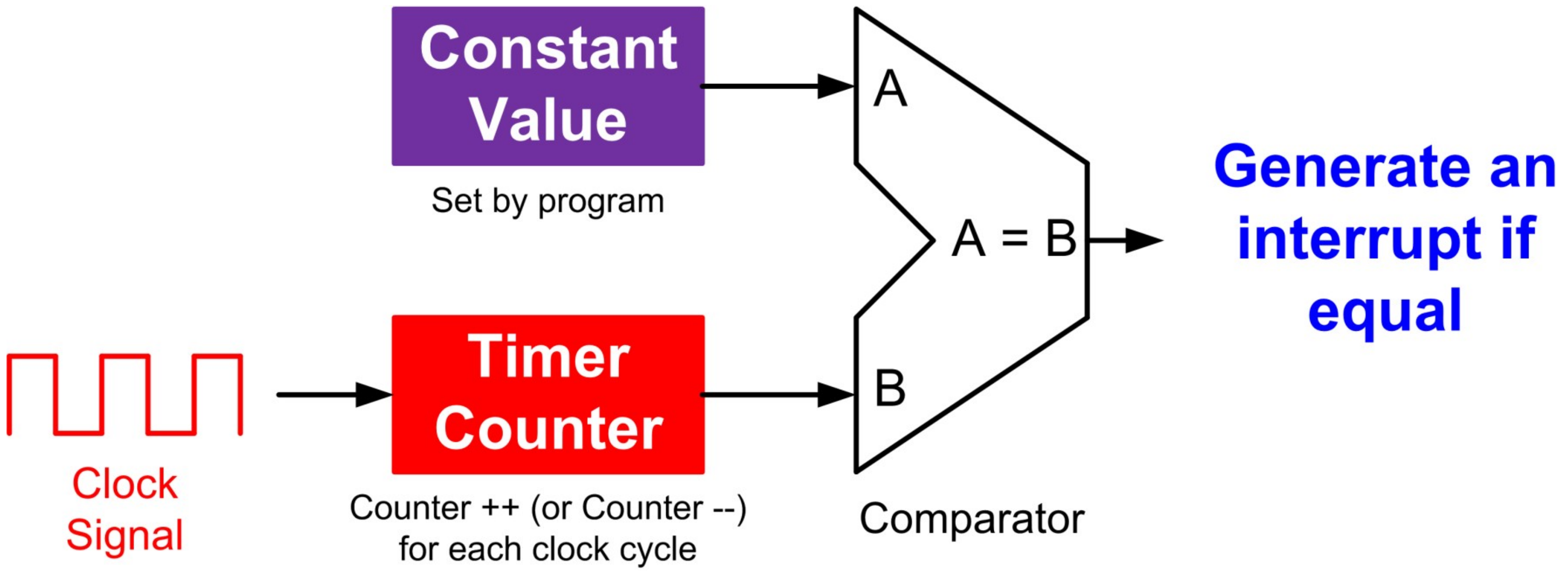
Timer: Input Capture



Multi-Channel Outputs

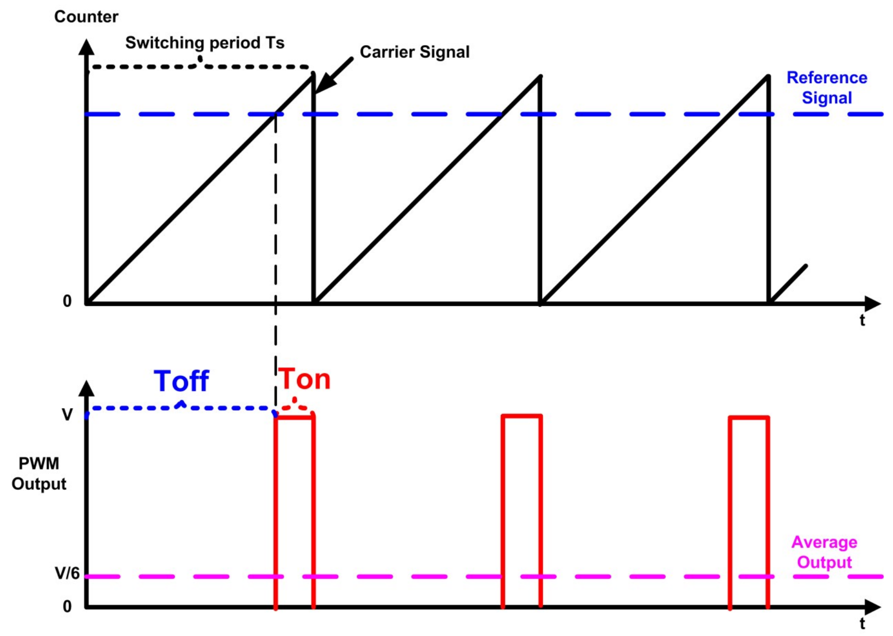


Output Compare


Constant Value
Set by program
A
A = B
B
Comparator
Generate an interrupt if equal
Clock Signal
Timer Counter
Counter ++ (or Counter --) for each clock cycle

Output Compare Mode (OCM)	Timer Output (OCREF)
000	Frozen
001	High if CNT == CCR
010	Low if CNT == CCR
011	Toggle if CNT == CCR
100	Forced low (always low)
101	Forced high (always high)

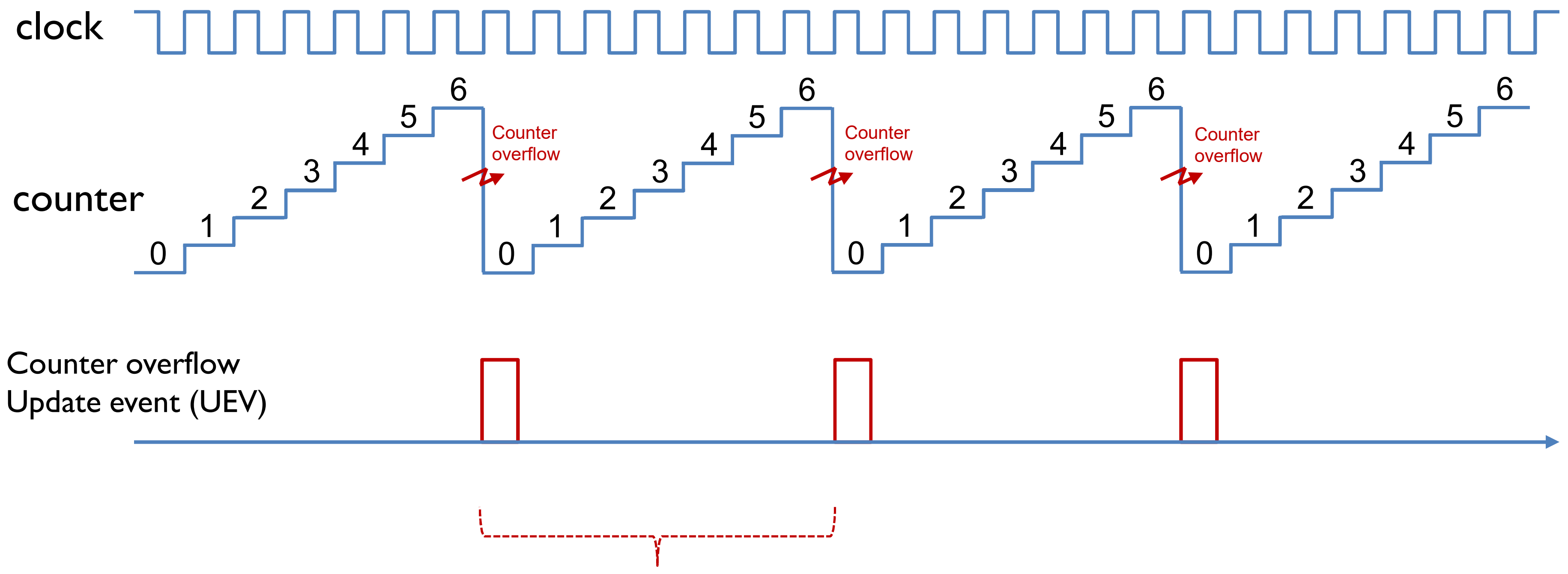
PWM Mode



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

Edge-aligned Mode (Up-counting)

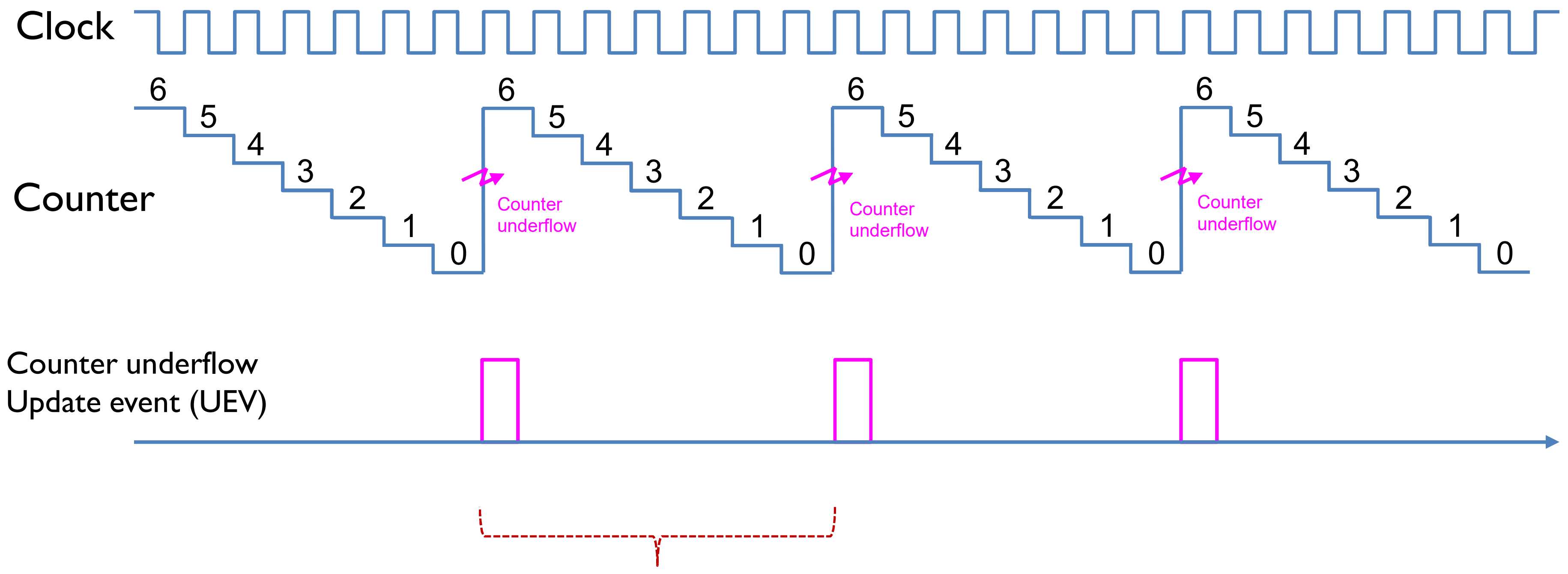
ARR = 6, RCR = 0



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

Edge-aligned Mode (down-counting)

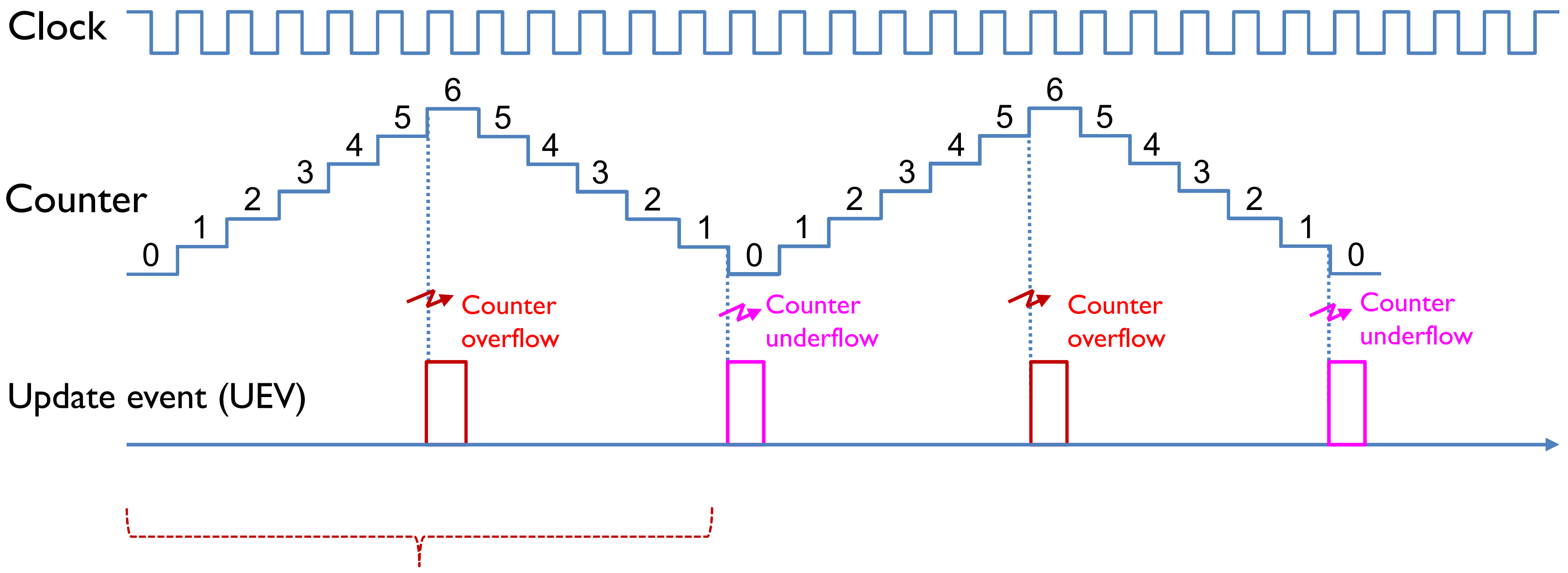
ARR = 6, RCR = 0



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

Center-aligned Mode

ARR = 6, RCR = 0



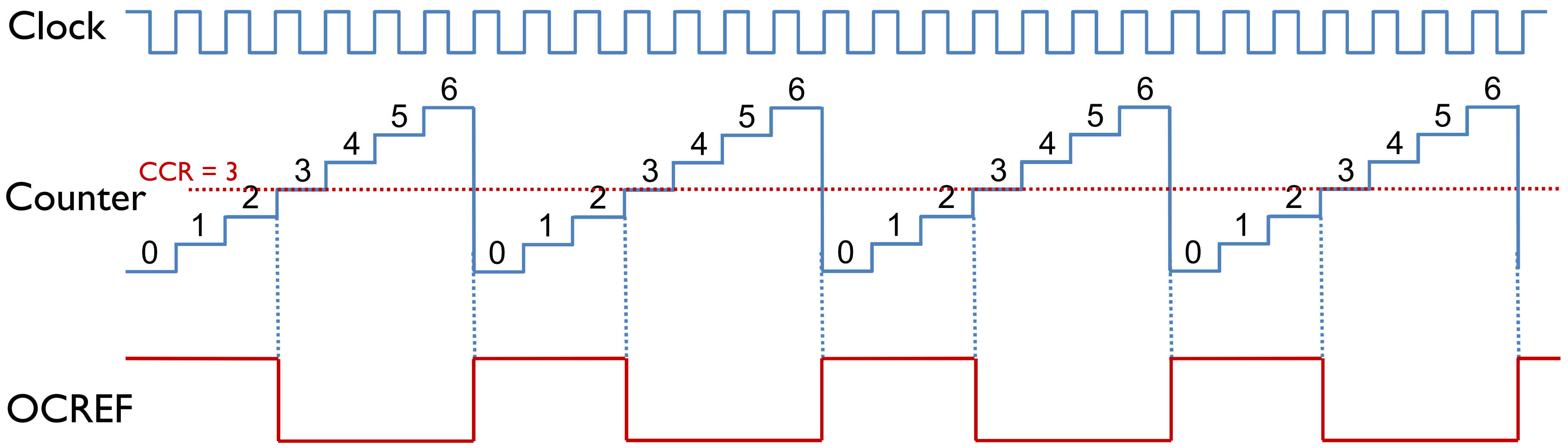
```
Period = 2 * ARR * Clock Period
       = 12 * Clock Period
```

PWM Mode 1 (Low-count True)

Mode 1

$$\text{Timer Output} = \begin{cases} \text{High if counter} < \text{CCR} \\ \text{Low if counter} \geq \text{CCR} \end{cases}$$

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



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

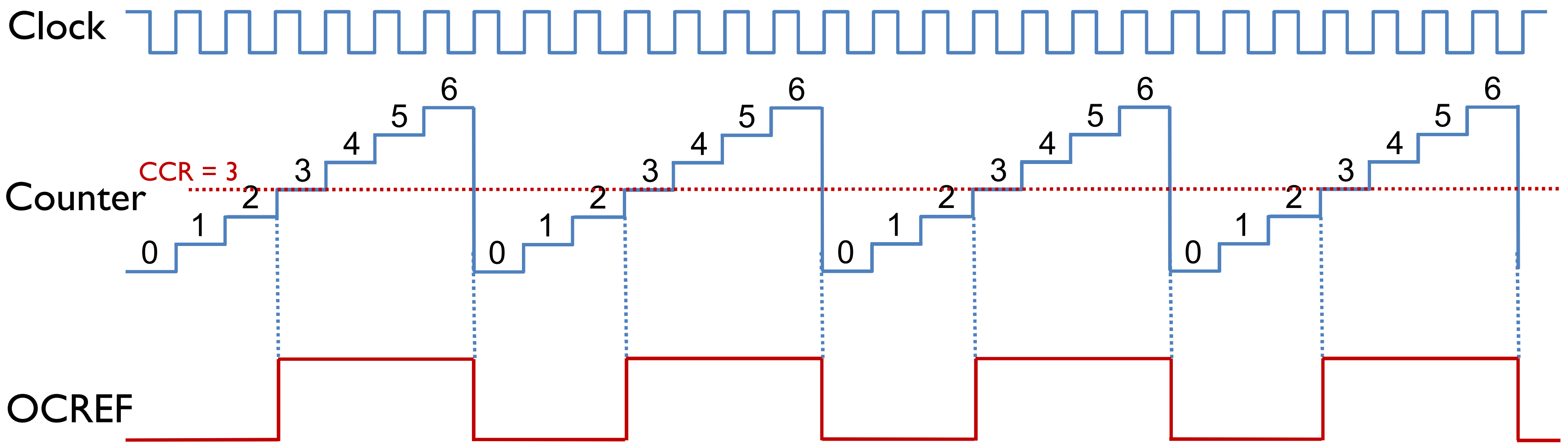
$$\begin{aligned}\text{Period} &= (1 + \text{ARR}) * \text{Clock Period} \\ &= 7 * \text{Clock Period}\end{aligned}$$

PWM Mode 2 (High-count True)

Mode 2

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

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



$$\text{Duty Cycle} = 1 - \frac{\text{CCR}}{\text{ARR} + 1} = \frac{4}{7}$$

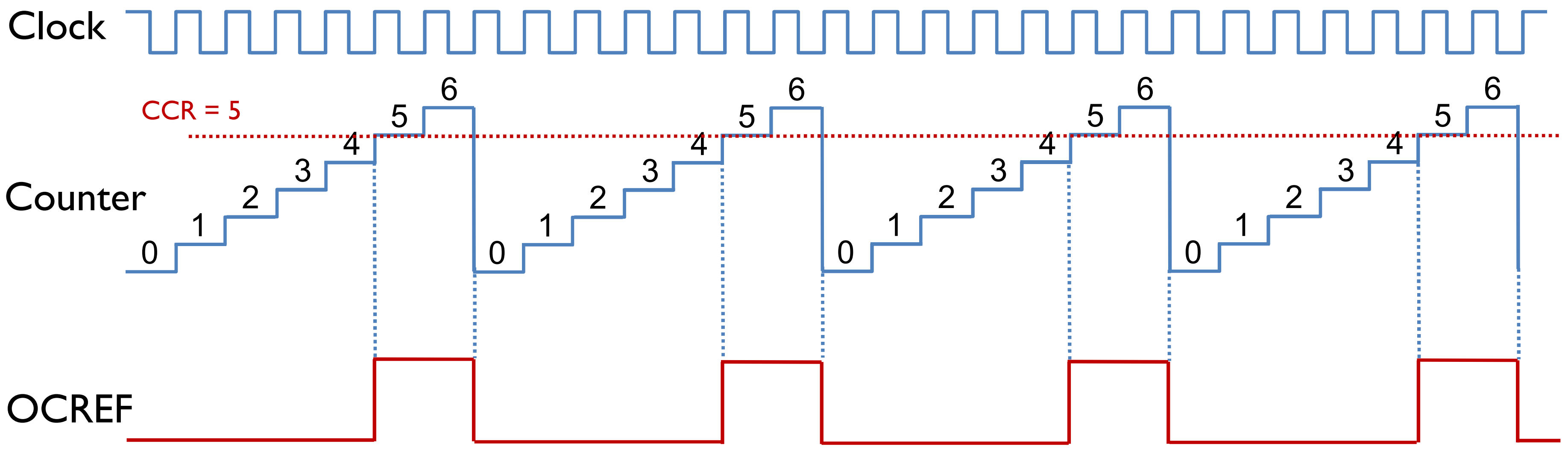
Period = (1 + ARR) * Clock Period
= 7 * Clock Period

PWM Mode 2 (High-count True)

Mode 2

Timer Output = $\begin{cases} \text{Low if counter} < \text{CCR} \\ \text{High if counter} \geq \text{CCR} \end{cases}$

Upcounting mode, ARR = 6, CCR = 5, RCR = 0



$$\text{Duty Cycle} = 1 - \frac{\text{CCR}}{\text{ARR} + 1} = \frac{2}{7}$$

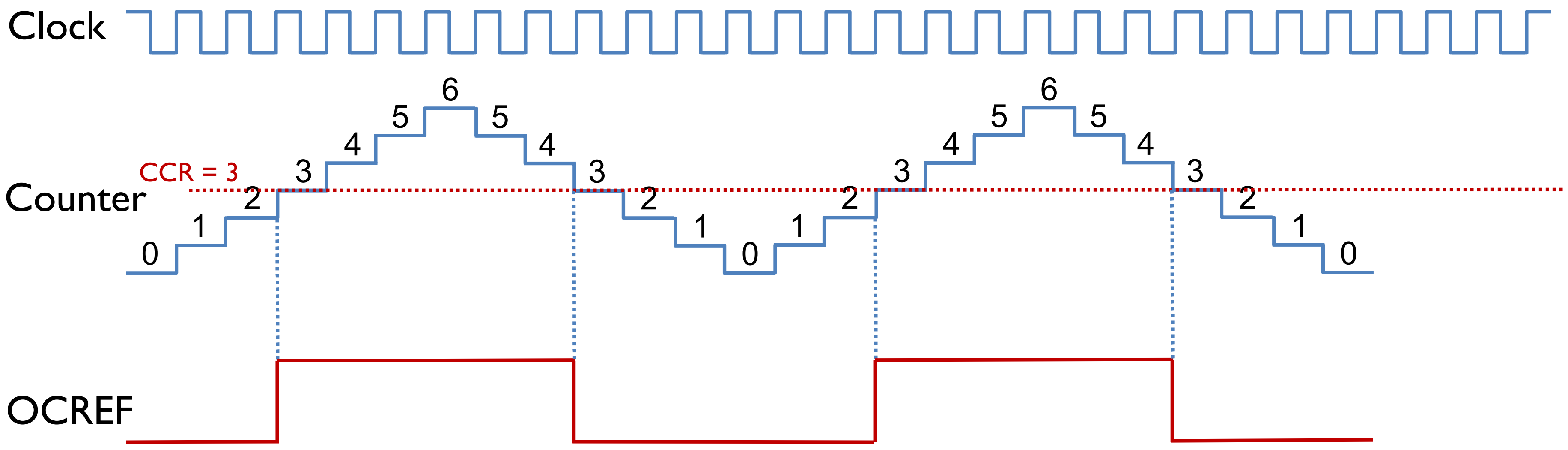
$$\text{Period} = (1 + \text{ARR}) * \text{Clock Period} = 7 * \text{Clock Period}$$

PWM Mode 2 (High-count True)

Mode 2

$$\text{Timer Output} = \begin{cases} \text{Low if counter} < \text{CCR} \\ \text{High if counter} \geq \text{CCR} \end{cases}$$

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



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

$$= \frac{1}{2}$$

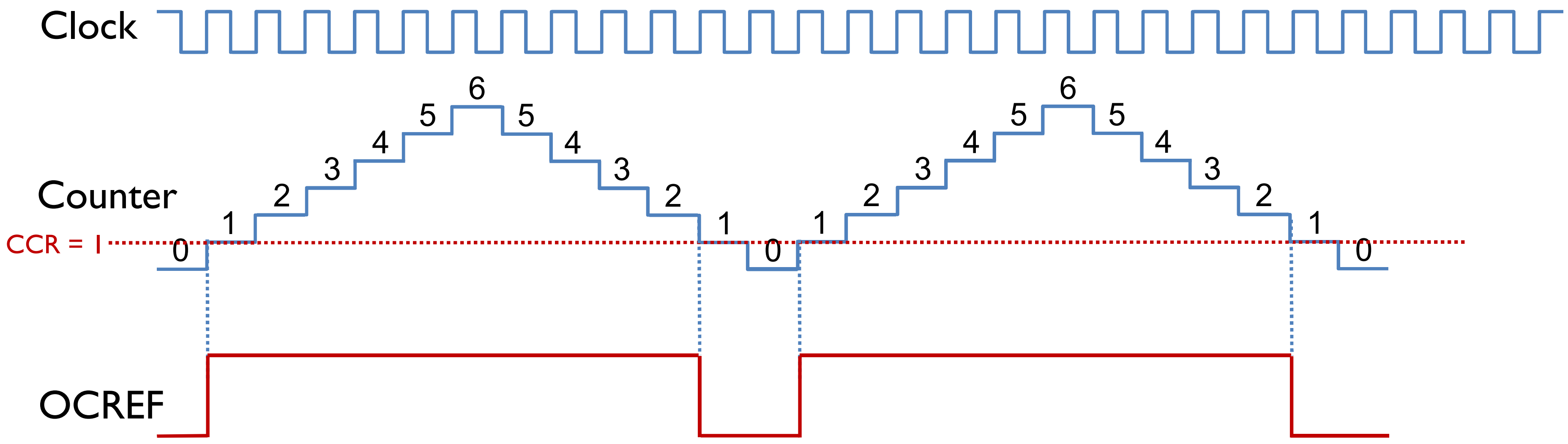
Period = 2 * ARR * Clock Period
= 12 * Clock Period

PWM Mode 2 (High count True)

Mode 2

$$\text{Timer Output} = \begin{cases} \text{Low if counter} < \text{CCR} \\ \text{High if counter} \geq \text{CCR} \end{cases}$$

Center-aligned mode, ARR = 6, CCR = 1, RCR = 0



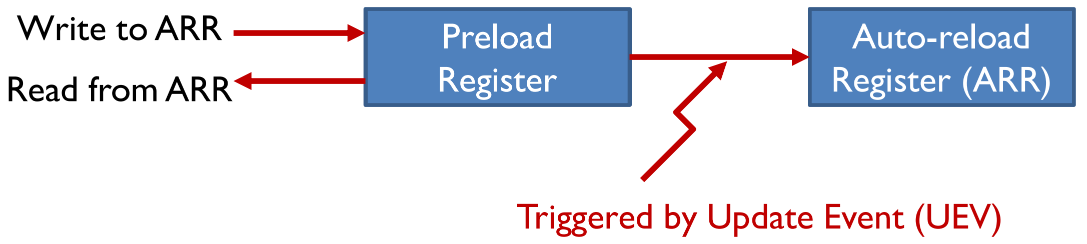
$$\text{Duty Cycle} = 1 - \frac{\text{CCR}}{\text{ARR}} = \frac{5}{6}$$

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

Auto-Reload Register (ARR)

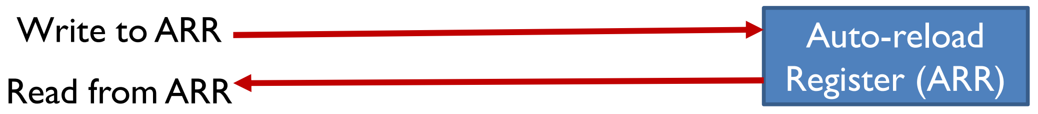
- Auto-Reload Preload Enable (ARPE) bit in TIMx_CR1

ARPE = 1 (Syn Update)

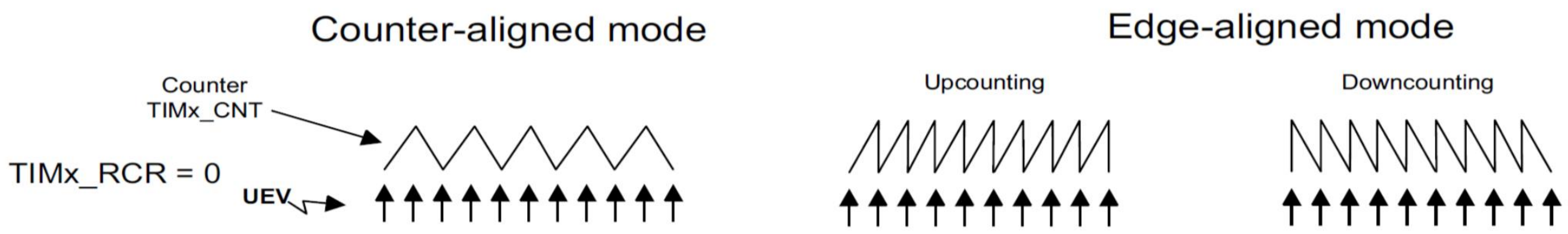


If UDIS bit in TIMx_CR1 is 1, UEV event is disabled.

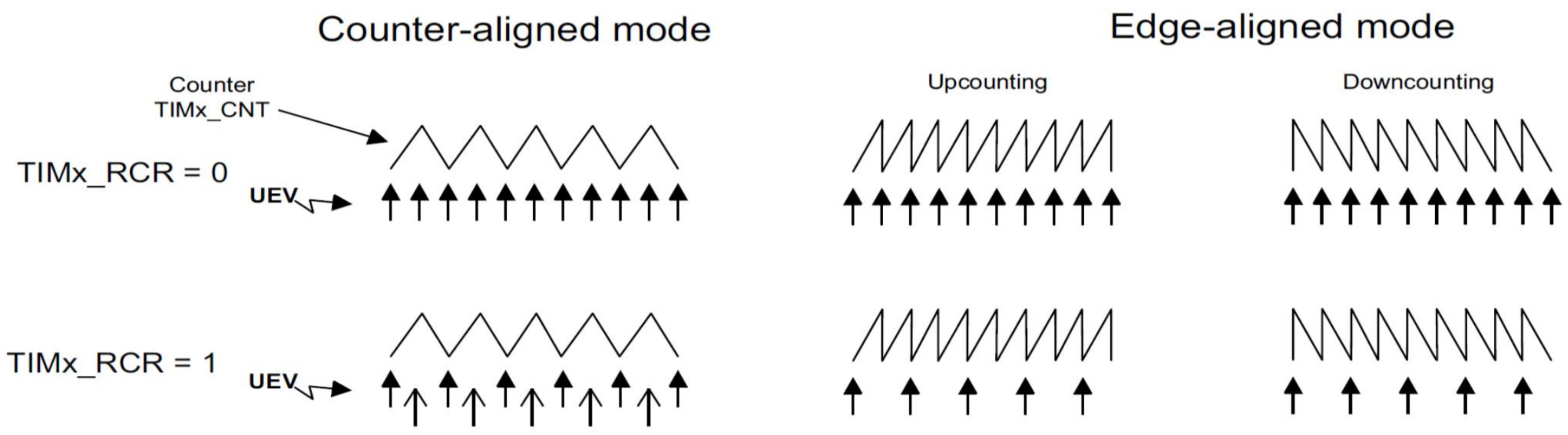
ARPE = 0 (Asyn Update)



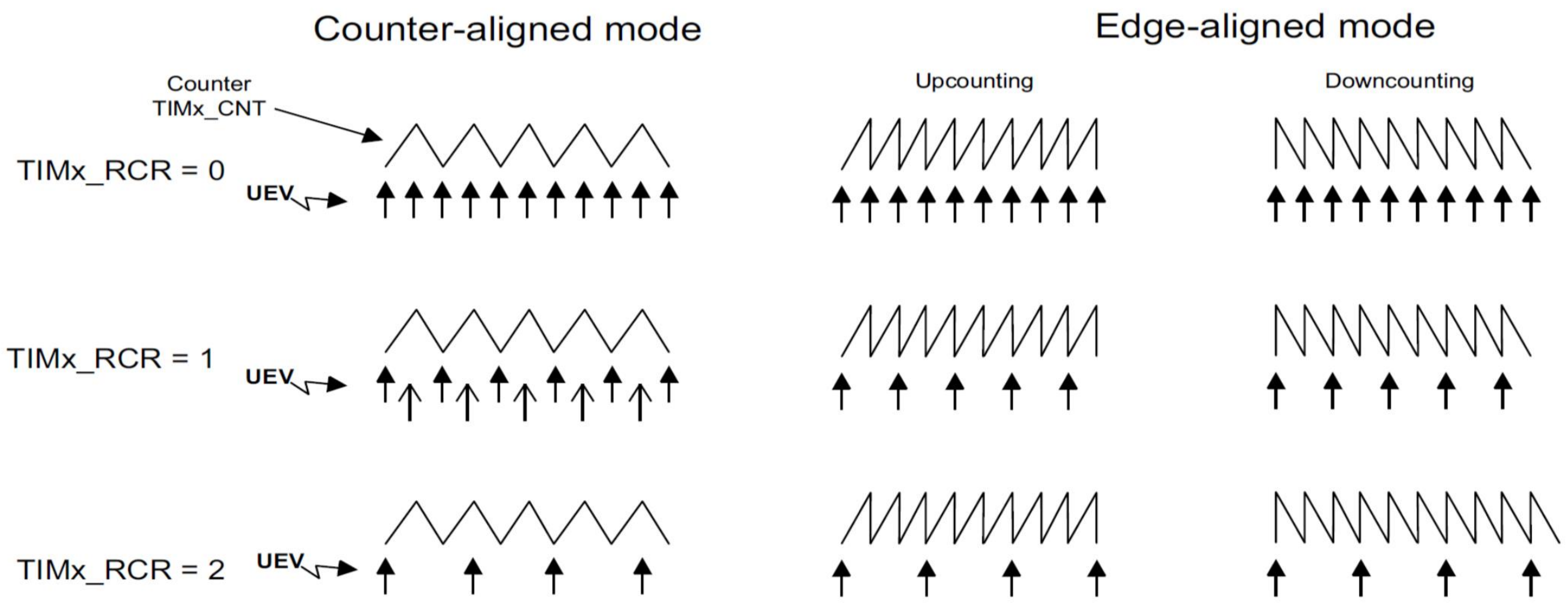
Repetition Counter Register (RCR)



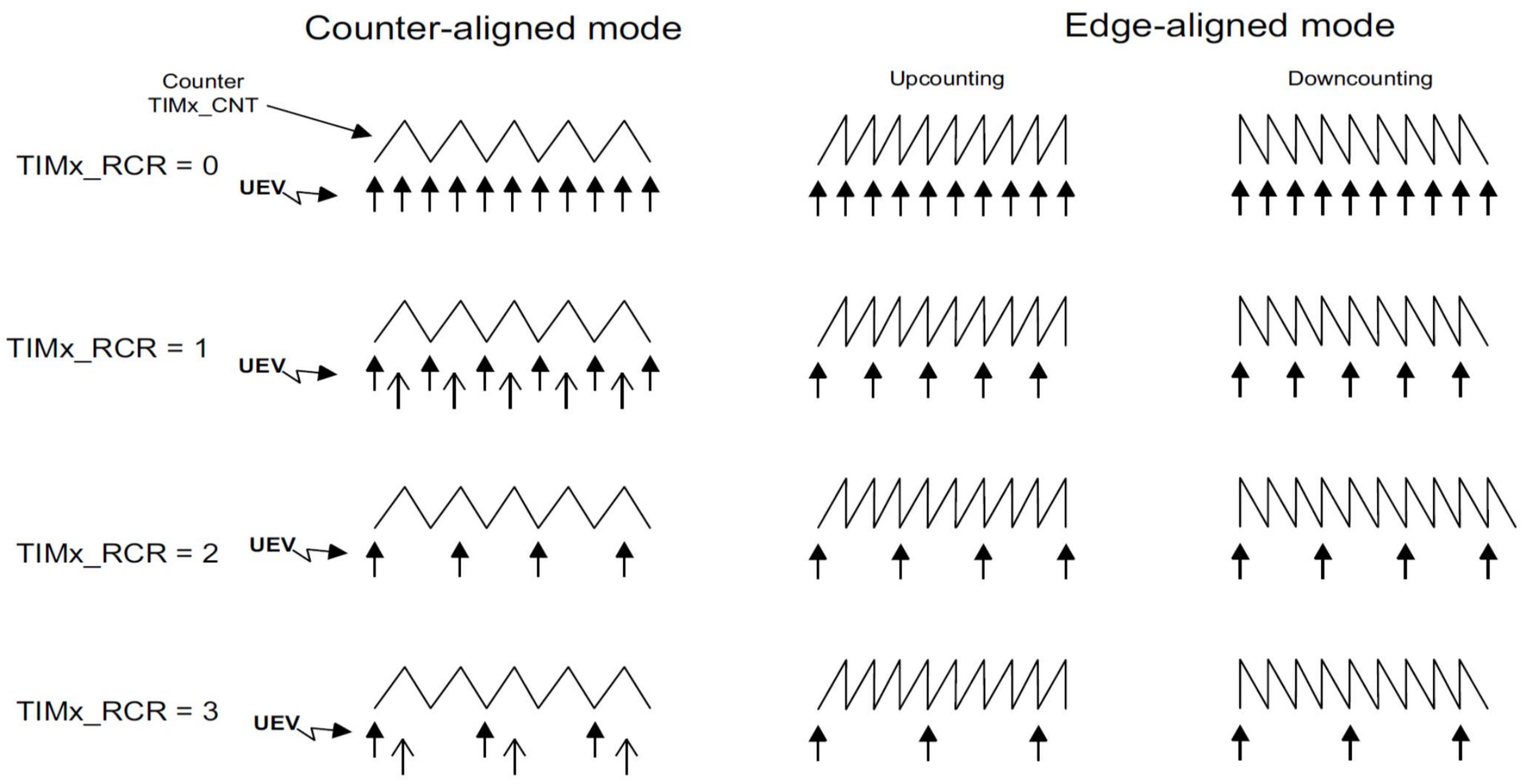
Repetition Counter Register (RCR)



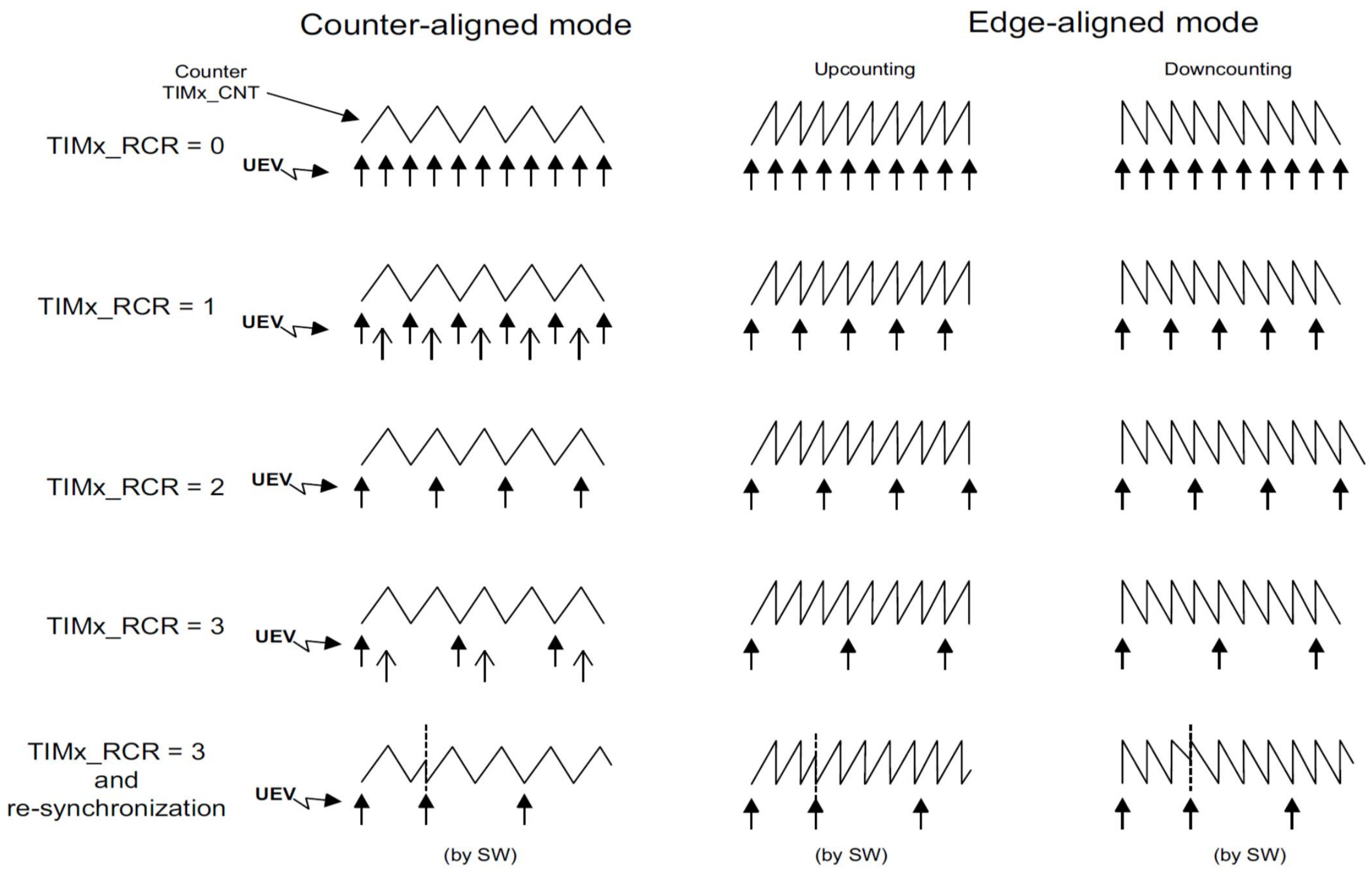
Repetition Counter Register (RCR)



Repetition Counter Register (RCR)



Repetition Counter Register (RCR)



PWM Output Polarity

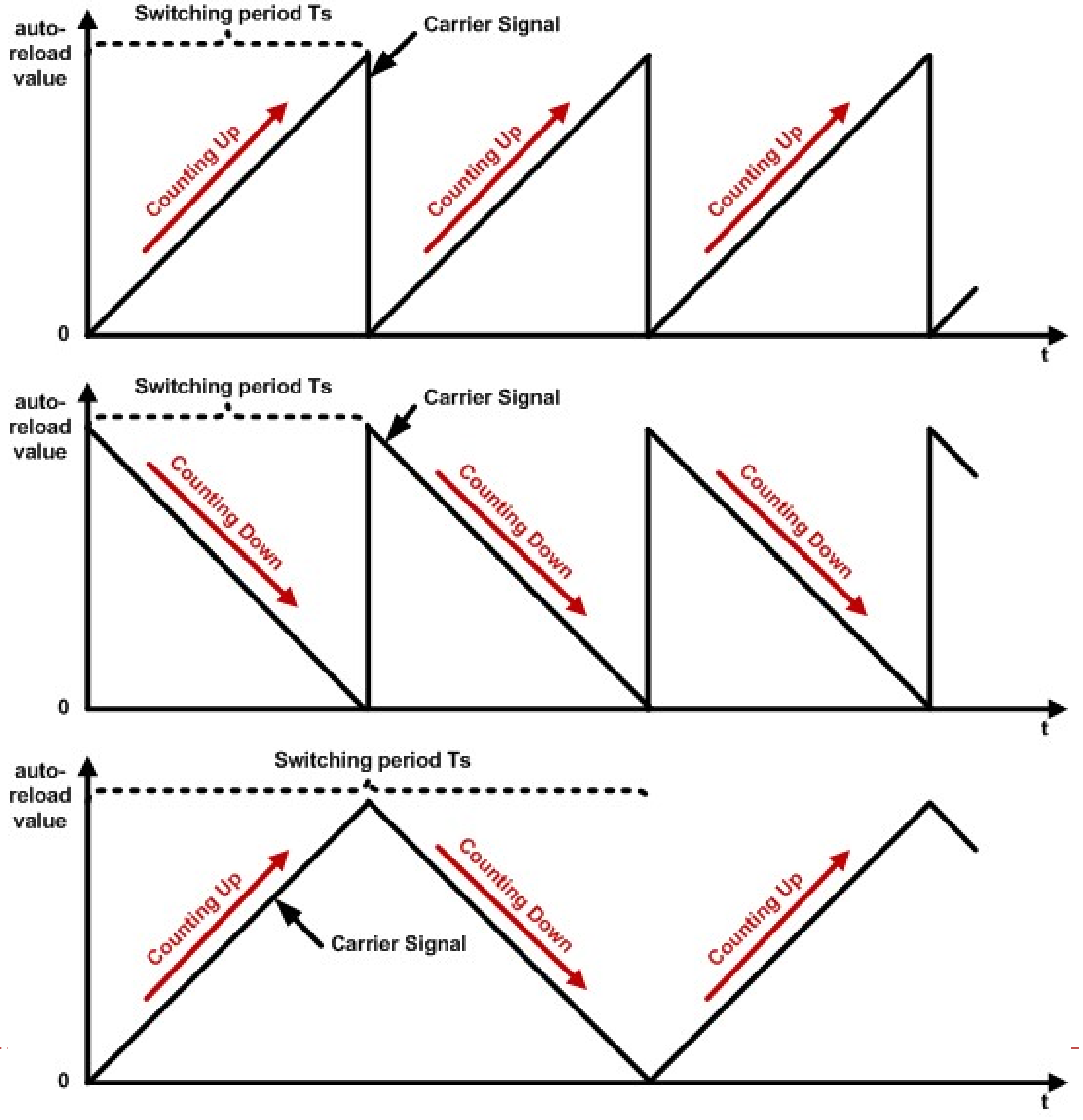
Mode	Counter < CCR	Counter ≥ CCR
PWM mode 1 (Low count True)	Active	Inactive
PWM mode 2 (High count True)	Inactive	Active

Output Polarity:

- Software can program the CCxP bit in the TIMx_CCER register

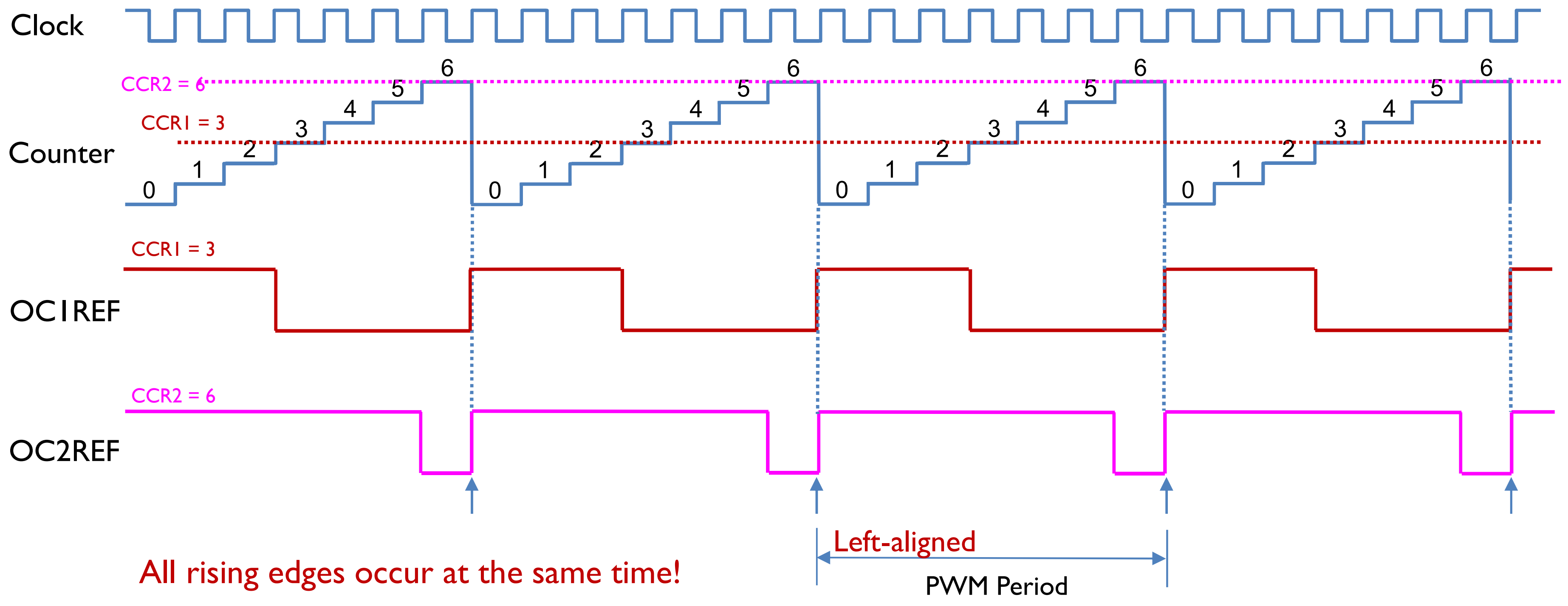
	Active	Inactive
Active High	High Voltage	Low Voltage
Active Low	Low Voltage	High Voltage

Counting up, down, center



PWM Mode 1 Up-Counting: Left Edge-aligned

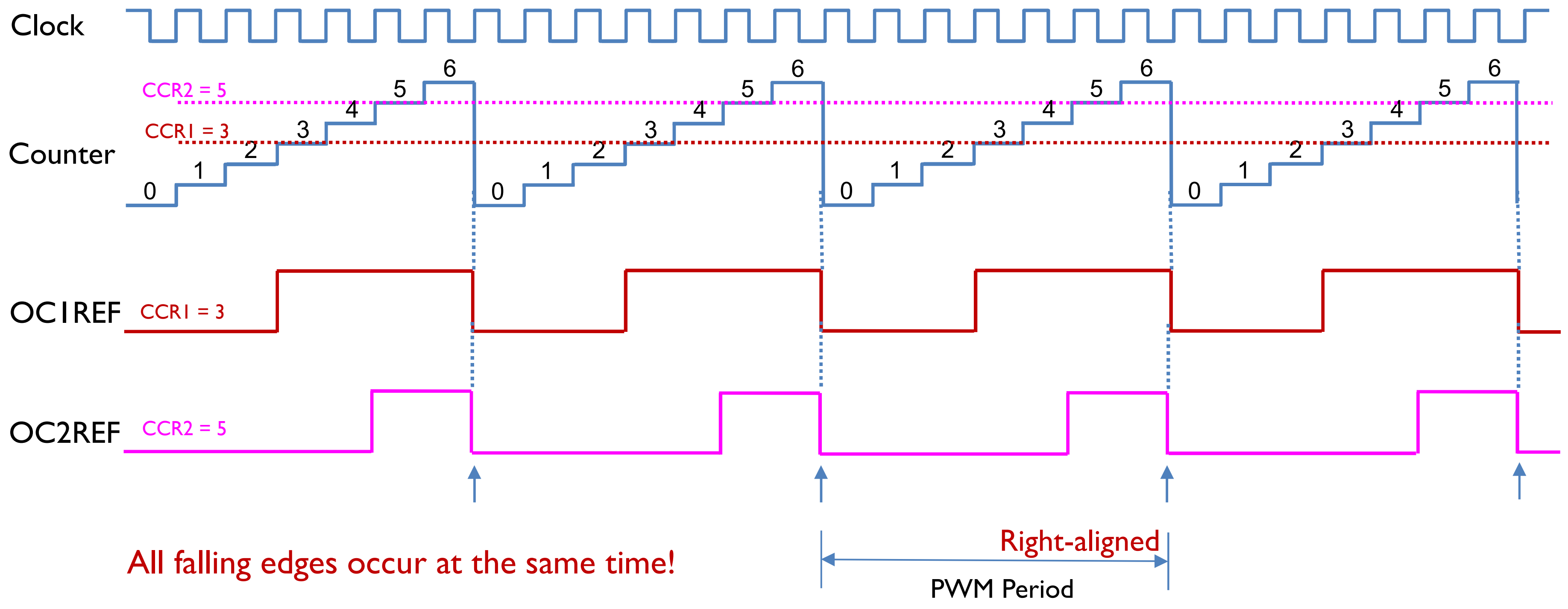
PWM Mode 1, Upcounting mode, ARR = 6, CCR1 = 3, CCR2=6, RCR = 0



All rising edges occur at the same time!

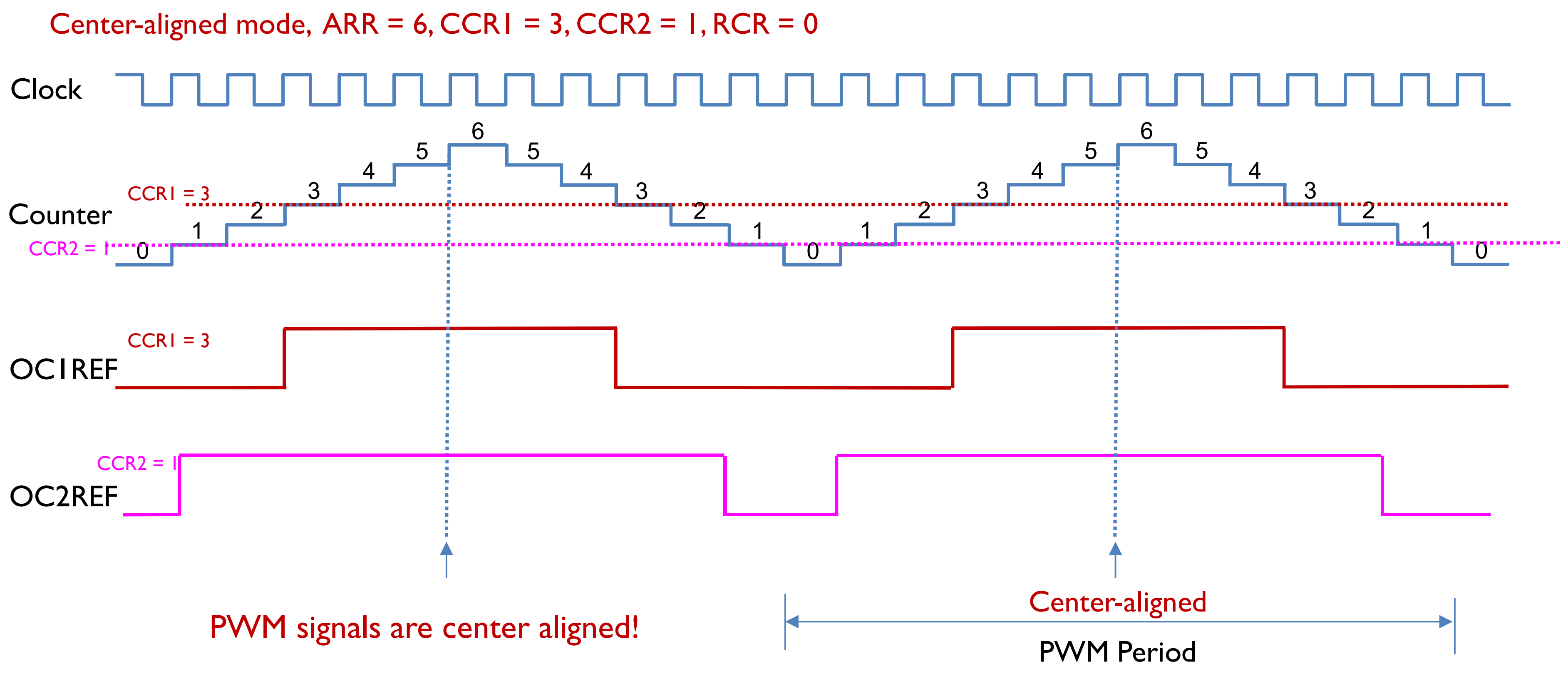
PWM Mode 2 Up-Counting: Right Edge-aligned

Upcounting mode, ARR = 6, CCR1 = 3, CCR2=5, RCR = 0



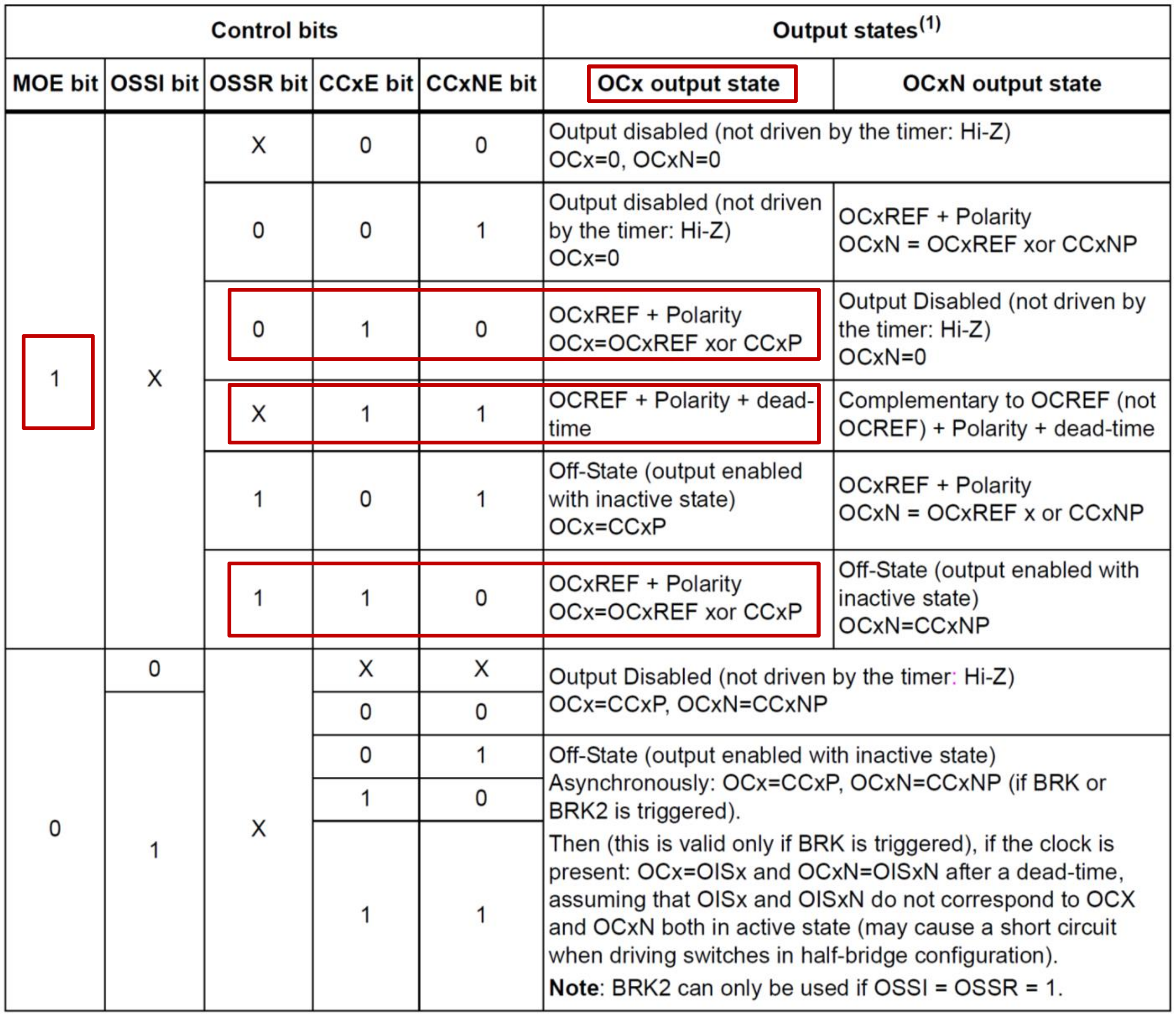
All falling edges occur at the same time!

PWM Mode 2: Center Aligned



The devil is in the detail

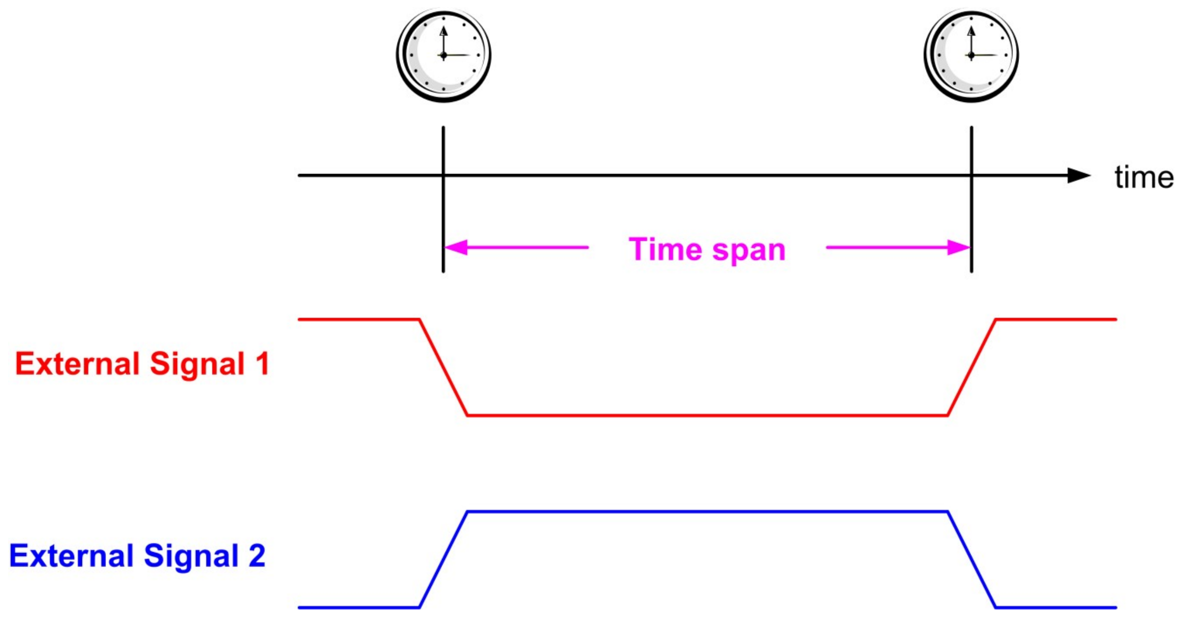
- Timer output control
- Enable Timer Output
  - MOE: Main output enable
  - OSSI: Off-state selection for Idle mode
  - OSSR: Off-state selection for Run mode
  - CCxE: Enable of capture/compare output for channel x
  - CCxNE: Enable of capture/compare complementary output for channel x



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		

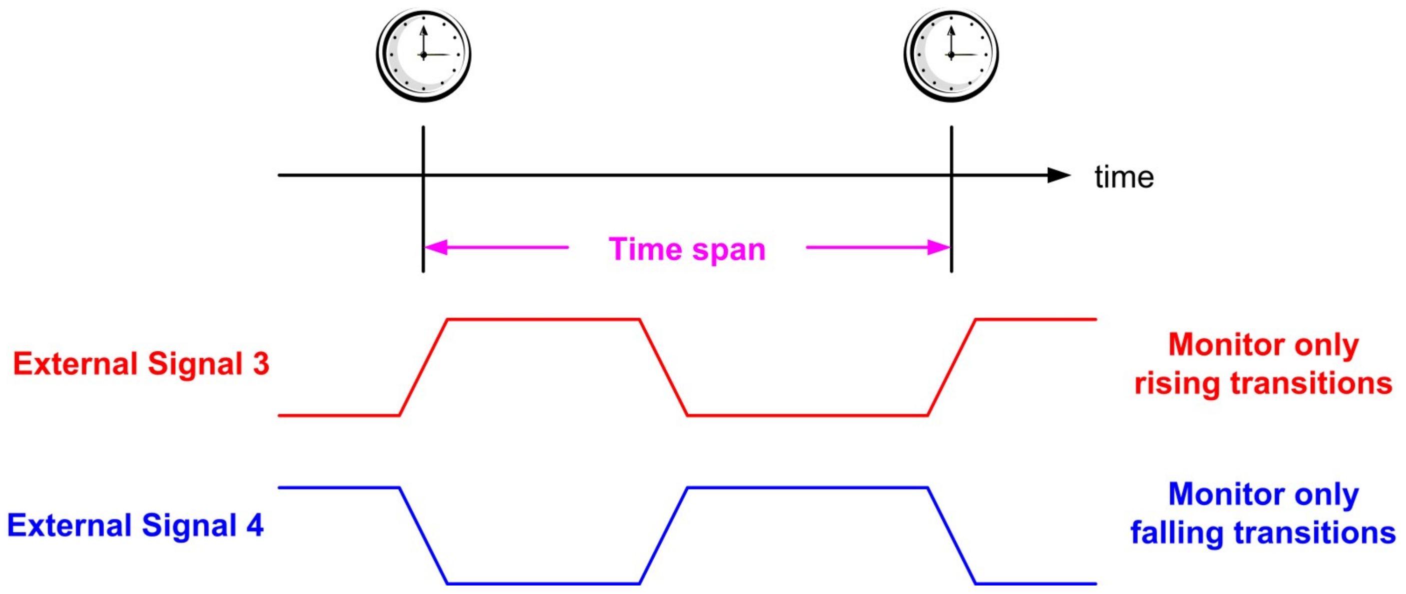
Input Capture

- Monitor both rising and falling edge

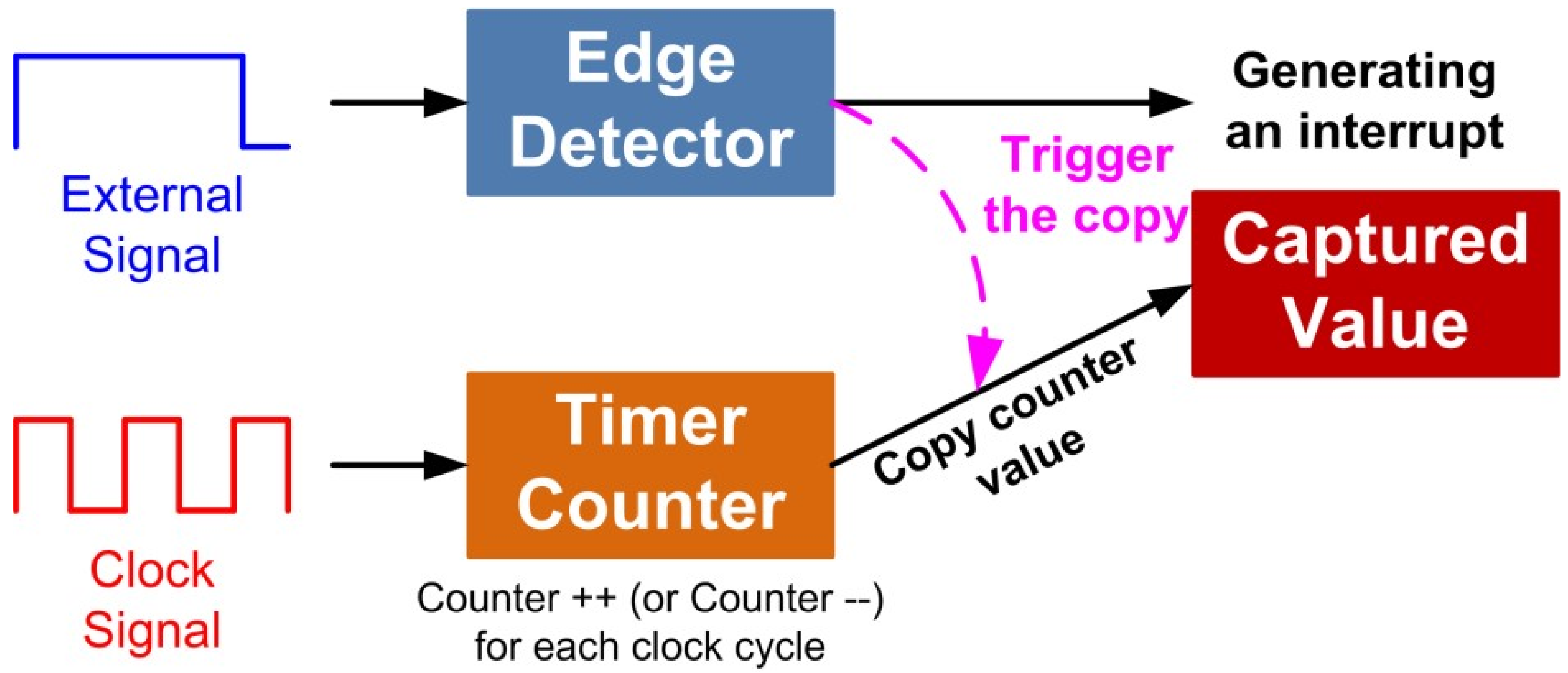


Input Capture

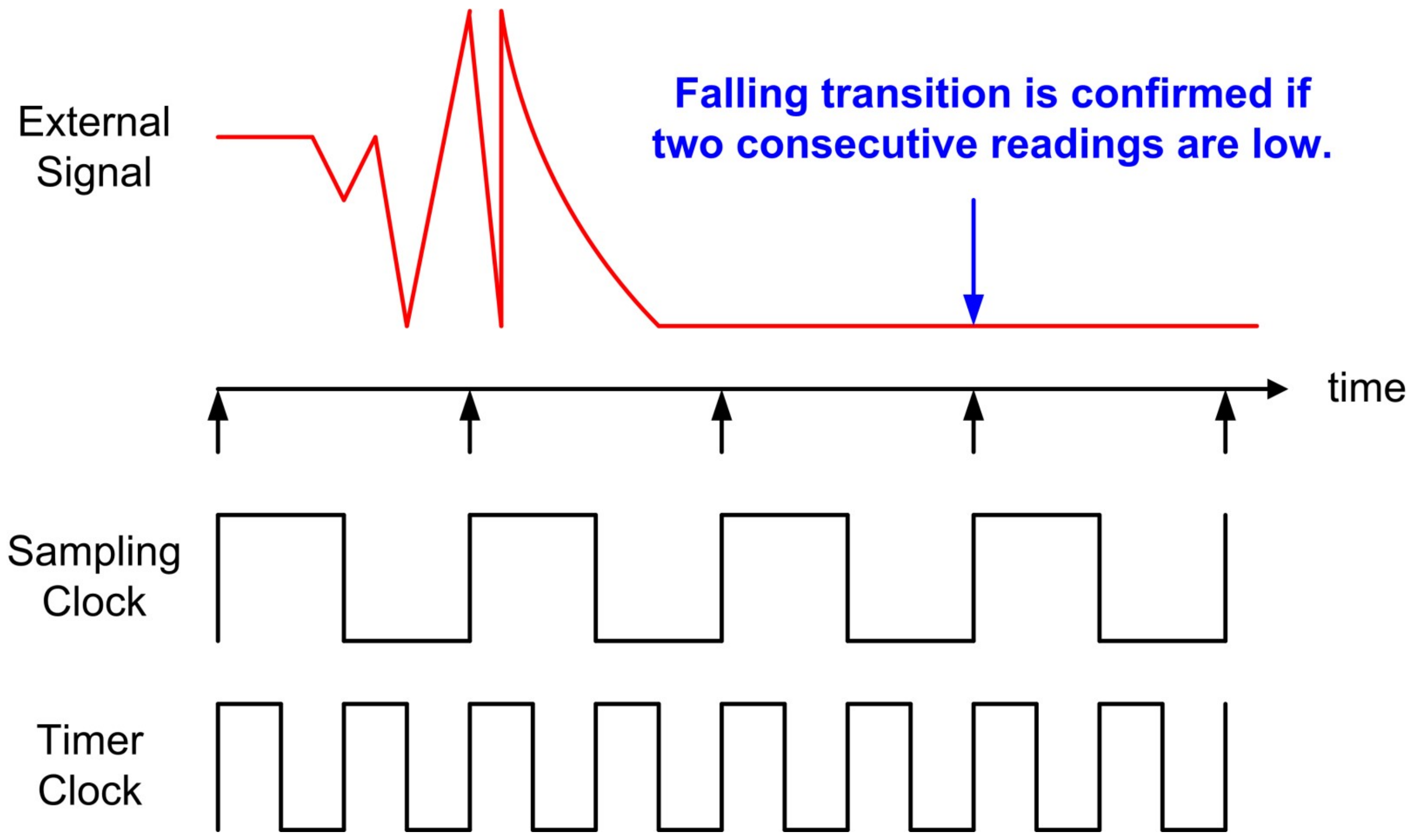
- Monitor only rising edges or only falling edge


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

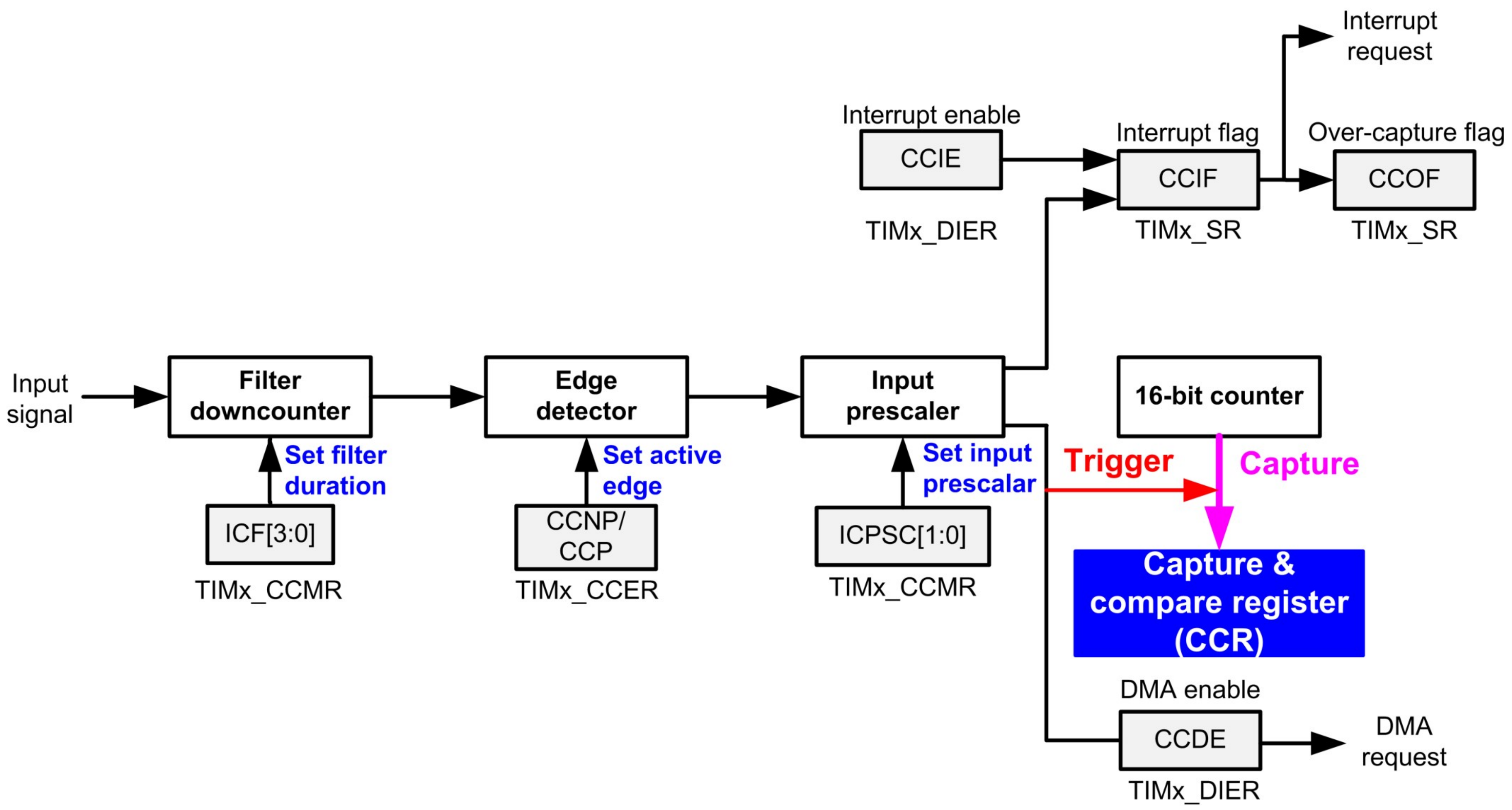
Input Capture



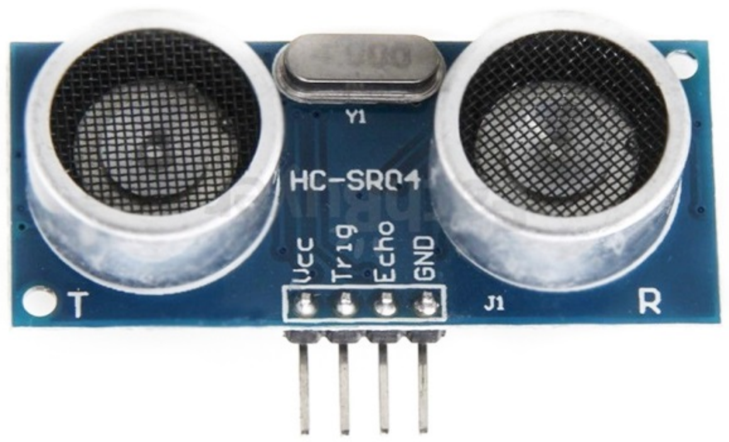
Input Filtering


External Signal
Falling transition is confirmed if two consecutive readings are low.
time
Sampling Clock
Timer Clock

Input Capture Diagram



Ultrasonic Distance Sensor

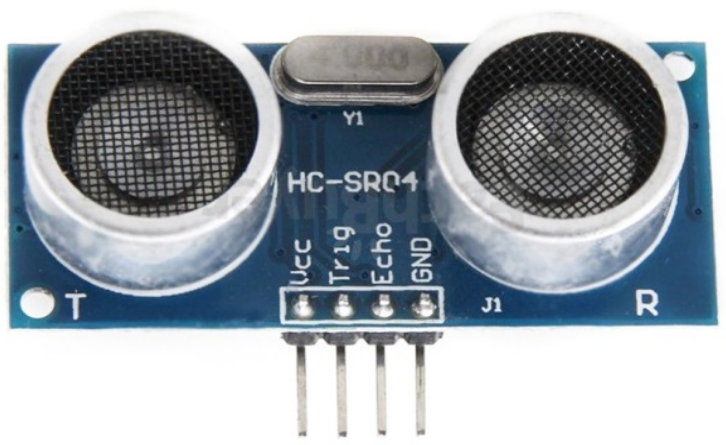


$$Distance = \frac{Round\ Trip\ Time \times Speed\ of\ Sound}{2}$$

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

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

Ultrasonic Distance Sensor

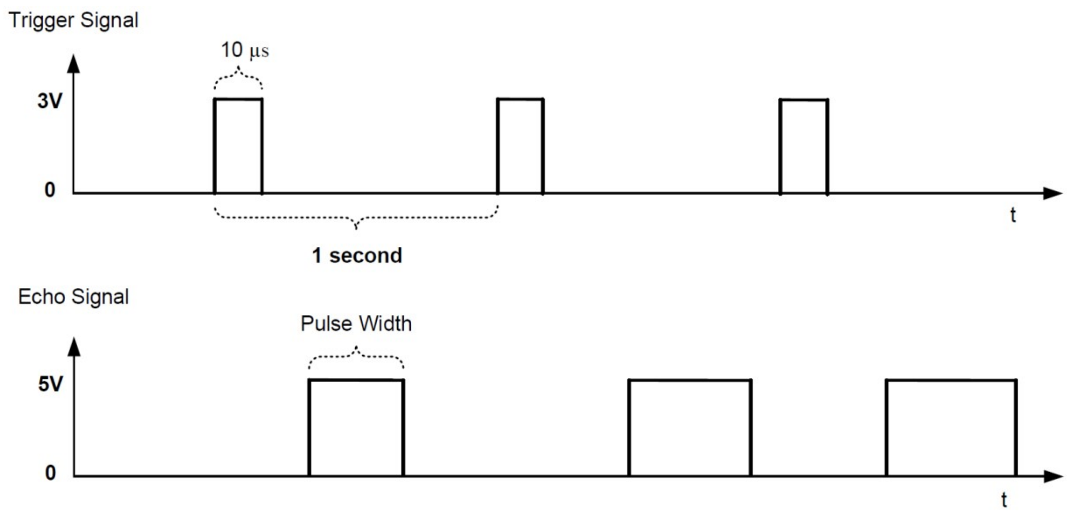


The echo pulse width corresponds to round-trip time.

$$Distance\ (cm) = \frac{Pulse\ Width\ (\mu s)}{58}$$

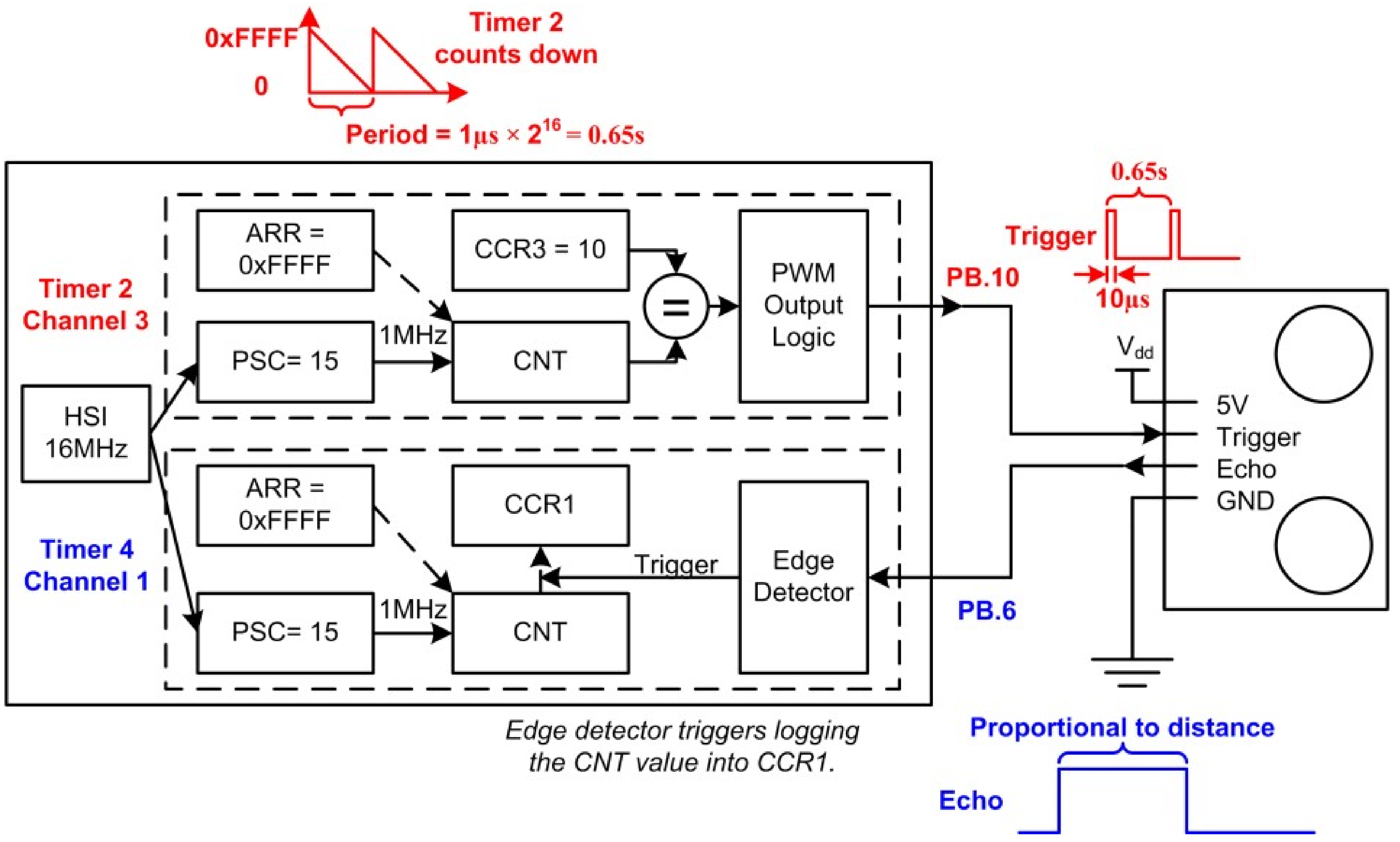
or

$$Distance\ (inch) = \frac{Pulse\ Width\ (\mu s)}{148}$$



If pulse width is 38*ms*,
no obstacle is detected.

Ultrasonic Distance Sensor



0xFFFF
0
Timer 2 counts down
Period = 1μs × 2^16 = 0.65s
Timer 2 Channel 3
ARR = 0xFFFF
CCR3 = 10
PSC= 15
1MHz
CNT
=
PWM Output Logic
HSI 16MHz
Timer 4 Channel 1
ARR = 0xFFFF
CCR1
PSC= 15
1MHz
CNT
Trigger
Edge Detector
PB.10
PB.6
Trigger
0.65s
10μs
Vdd
5V
Trigger
Echo
GND
Edge detector triggers logging the CNT value into CCR1.
Proportional to distance
Echo