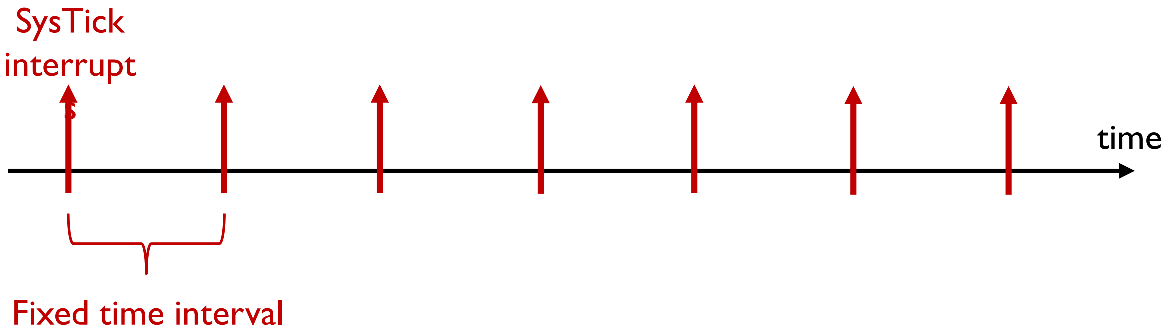
L6 Timer and PWM

Zonghua Gu, 2018

Acknowledgement: some slides taken from Yifeng Zhu's courseware

System Timer (SysTick)

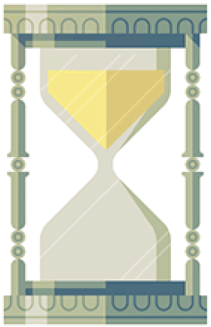
- A timer is a hardware module that generates periodic clock ticks at a fixed time interval



SysTick interrupts

time

Fixed time interval



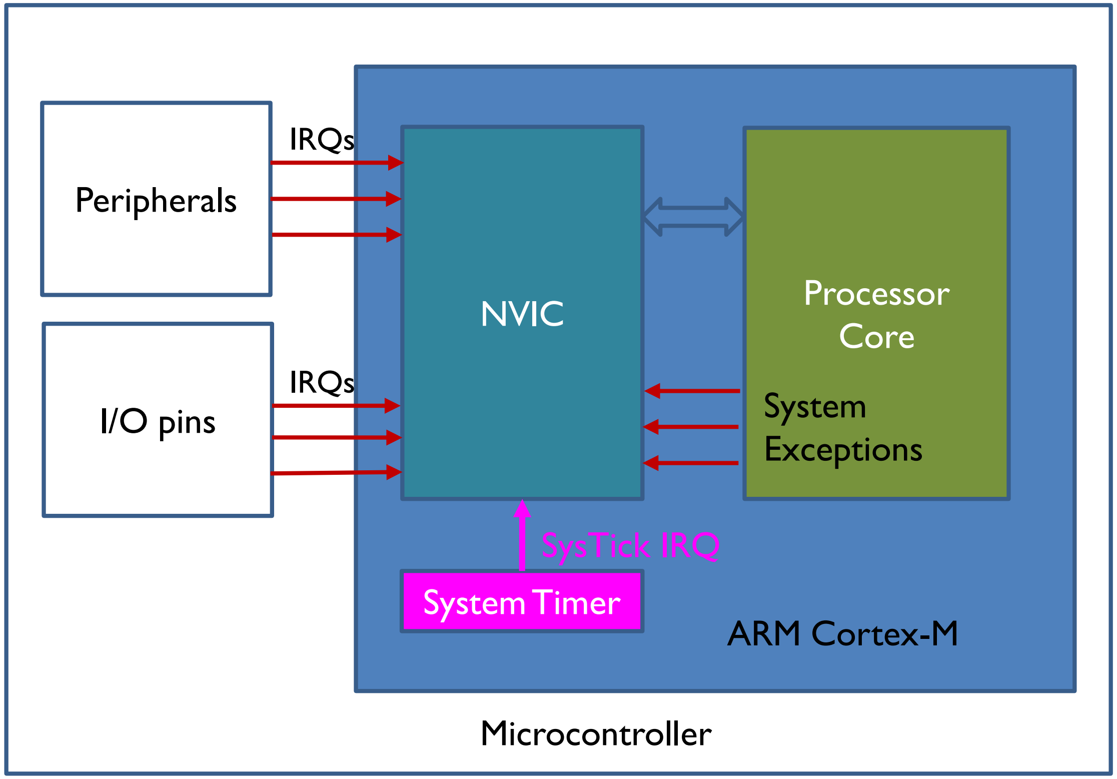
- Example Usages:
  - To measure time elapsed, e.g., to implement a time delay function Delay(100ms).
  - To implement periodic polling to check peripheral device status, or to implement CPU scheduler. In the OS scheduler, which periodically selects a new process, from the ready queue, to run next.

System Timer (SysTick)

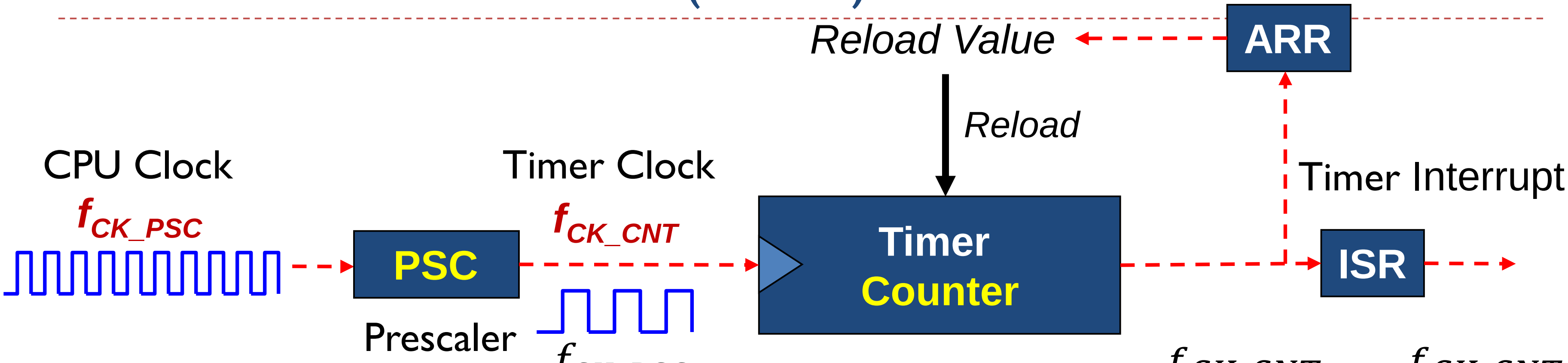
- System timer (SysTick) is a standard hardware component built into ARM Cortex-M.

- This hardware periodically interrupts the processor to execute the following ISR:

```
void
SysTick_Handler(void)
{
   ...
}
```



Timer: Prescaler (PSC)



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

$$f_{\text{Timer}} = \frac{f_{CK\_CNT}}{ARR+1} \text{ or } \frac{f_{CK\_CNT}}{2*ARR}$$

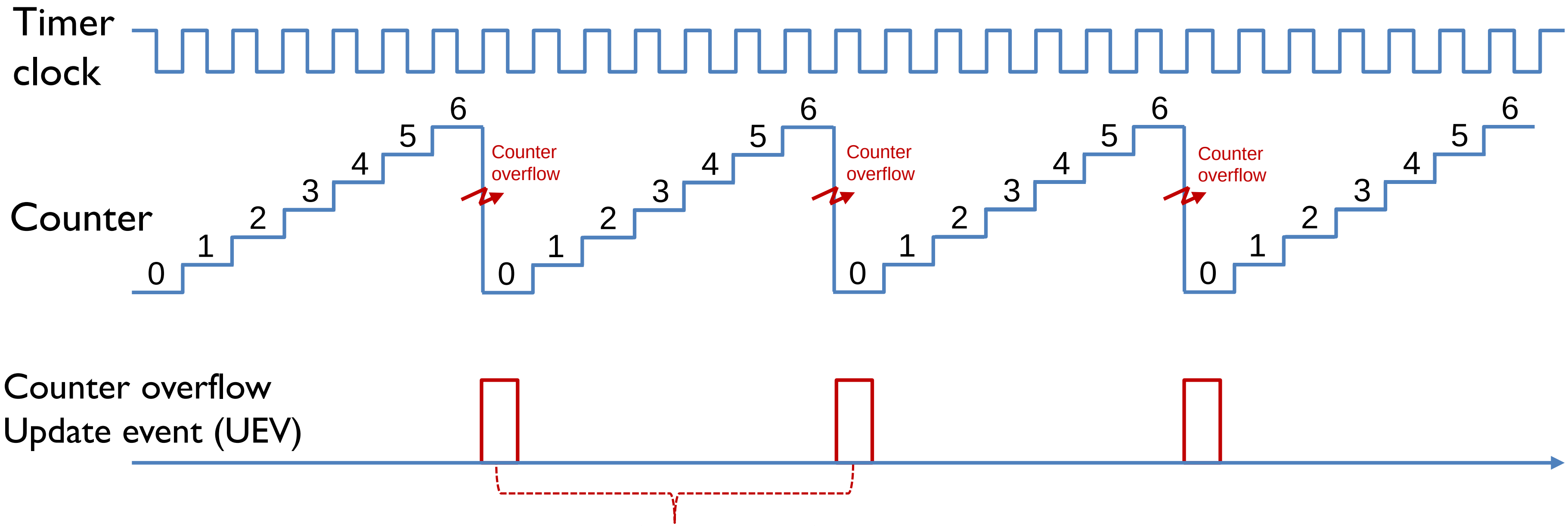
- The 16-bit PSC register holds the frequency pre-scaler value in the range of [0, 216-1=65535]. The CPU Clock Frequency f_{CK_PSC} is divided by a constant integer, PSC+1, in the range of [1, 216=65536], to generate the Timer Clock with frequency f_{CK_CNT} that drives the Timer Counter. The prescaler serves as a "the frequency divider" that enables tradeoffs between timer resolution and timer range. Larger PSC value leads to slower timer clock rate, coarser timer resolution, and larger timer range.
- The 16-bit Auto-Reload Register (ARR) holds the maximum timer counter value in the range of [0, 65535].
- CPU clock and timer clock: measured in MHz/GHz; Timer Clock: measured in Hz, e.g., the Linux OS timer frequency is 100Hz (period=10ms). Since each timer interrupt triggers an ISR software execution, it is not realistic to have timer frequency higher than 1KHz.

3 Modes

- A timer can be configured in 3 modes:
  - Up-counting mode
  - Down-counting mode
  - Center-aligned counting mode （Alternating up-counting and down-counting）
- Related registers include:
  - Counter register (TIMx_CNT)
  - Prescaler register (TIMx_PSC)
  - Auto-reload register (TIMx_ARR)
  - Their values are denoted as CNT, PSC and ARR.

Up-counting Mode

ARR = 6

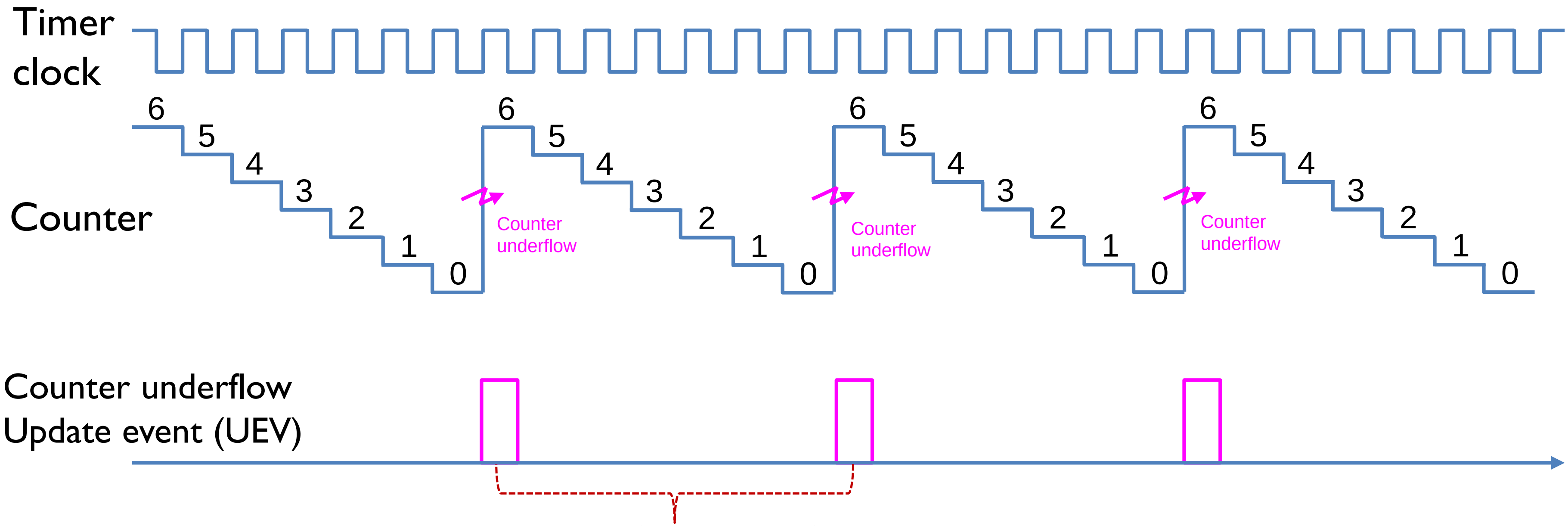


```
Timer Period = (ARR + 1) * Clock Period
             = 7 * Clock Period
```

Auto-Reload Register ARR=6. In up-counting mode, the counter counts up, from 0 to 6. When the counter is reset to 0 from 6, counter overflow occurs, and a Update event (UEV) is generated. On the next clock cycle, the counter counts up again. The Timer Period is 1+ARR clock ticks, with duration of (1+ARR)*Clock Period.

Down-counting Mode

ARR = 6

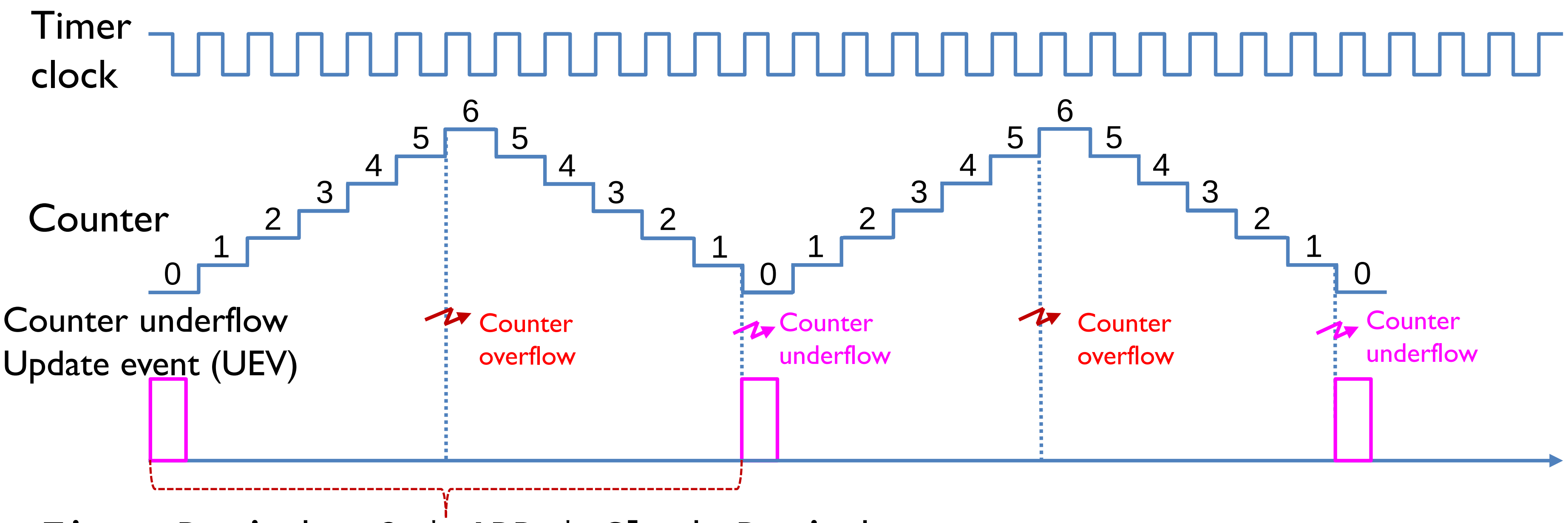


```
Timer Period = (ARR + 1) * Clock Period
             = 7 * Clock Period
```

ARR=6. In down-counting mode, the counter counts down, from 6 to 0. When the counter is reset to 6 from 0, counter underflow occurs, and a UEV is generated. On the next clock cycle, the counts down again. The Timer Period is 1+ARR clock ticks, with duration of (1+ARR)*Clock Period.

Center-aligned Counting Mode

ARR = 6

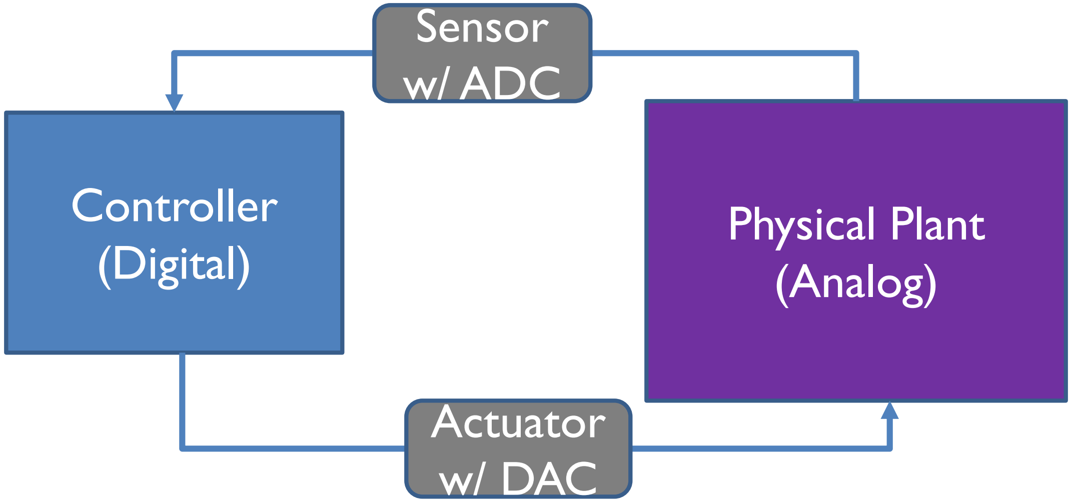


```
Timer Period = 2 * ARR * Clock Period
             = 12 * Clock Period
```

ARR=6. In center-aligned counting mode, the counter first counts up, from 0 to 6, then counts down, from 6, to 0. When the counter makes transition from 1 to 0, a UEV is generated. On the next clock cycle, the counter repeats the up/down counting process. The Timer Period is 2*ARR clock ticks, with duration of 2*ARR*Clock Period.

Pulse-Width Modulation (PWM) as DAC

- A closed-loop control system consisting of a CPU-based controller (digital) interacting with a physical plant (analog) needs
  - Sensors with ADC (Analog-to-Digital Convertor) functionality to convert analog signals into digital domain, e.g., temperature, humidity…
  - Actuators with DAC (Digital-to-Analog Convertor) functionality to convert digital signals into analog domain, e.g., output voltage to control a motor
- PWM is a type of DAC. It uses a rectangular waveform to periodically switch on and off a voltage source to produce a desired average voltage output.



Pulse-Width Modulation (PWM)

- (Analog-to-Digital Convertor) The "Pulse Width" in PWM refers to the time interval when the output signal is On; the Duty Cycle is the percentage of time that the output signal is On ("the pulse width"):

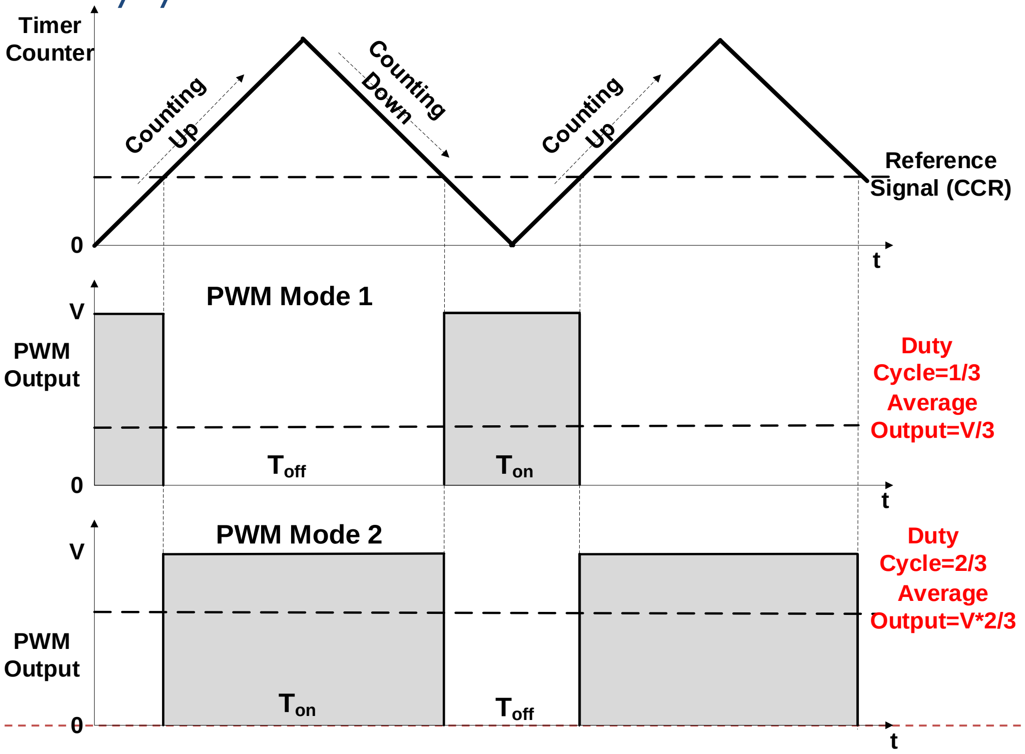
$$\text{Duty Cycle} = \frac{\text{On Time } (Ton)}{\text{Switching Period } (Ts)} = \frac{T_{on}}{T_{on} + T_{off}}$$

- Higher Duty Cycle means higher average output:

$$\text{Average Output} = \text{Duty Cycle } * \text{ Output in On State}$$

Pulse-Width Modulation (PWM)

- The PWM output signal is determined by:
  - Reference signal stored in Compare and Capture Register (CCR); and the PWM mode
- PWM mode 1 (Low True): if the timer counter is lower than the reference signal stored in CCR, the PWM output is on; otherwise it is off.
  - Higher CCR → higher duty cycle
- PWM mode 2 (High True): if the timer counter is higher than the reference signal stored in CCR, the PWM output is on; otherwise it is off.
  - Higher CCR → lower duty cycle



PWM Mode: Rigorous Definition

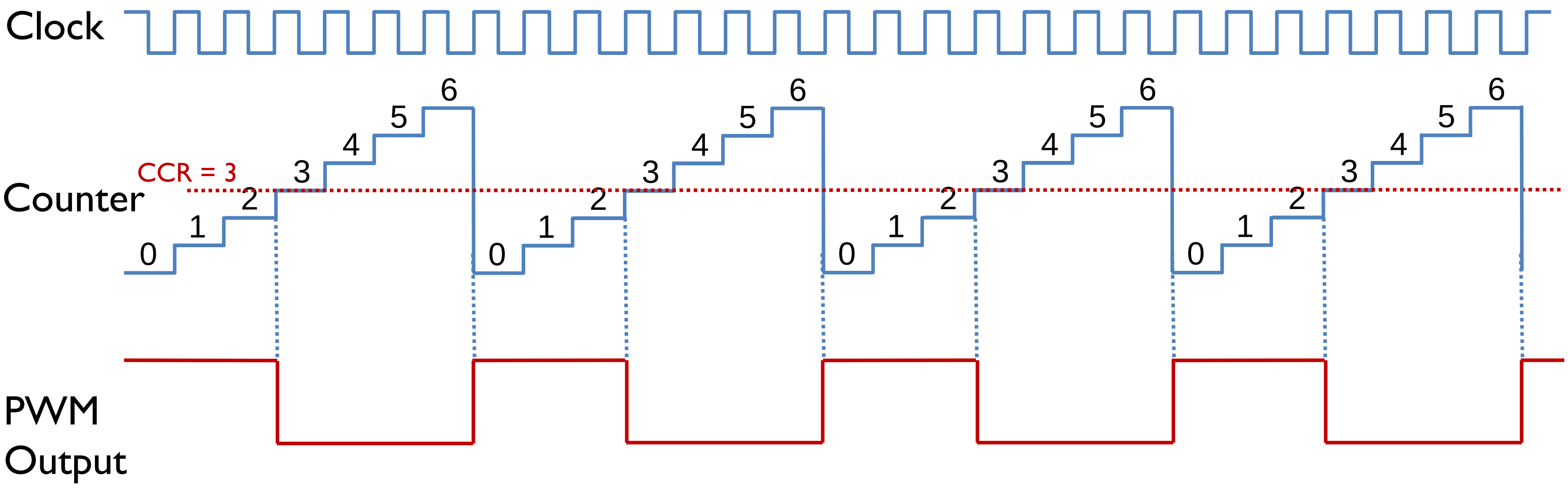
PWM Mode	Timer Counting Mode	On	Off
Mode 1 (Low True)	Up-counting	CNT < CCR	CNT ≥ CCR
	Down-counting	CNT ≤ CCR	CNT > CCR
Mode 2 (High True)	Up-counting	CNT ≥ CCR	CNT < CCR
	Down-counting	CNT > CCR	CNT ≤ CCR

PWM Mode 1 (Low-True)

Mode 1

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

Up-counting mode, ARR = 6, CCR = 3



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

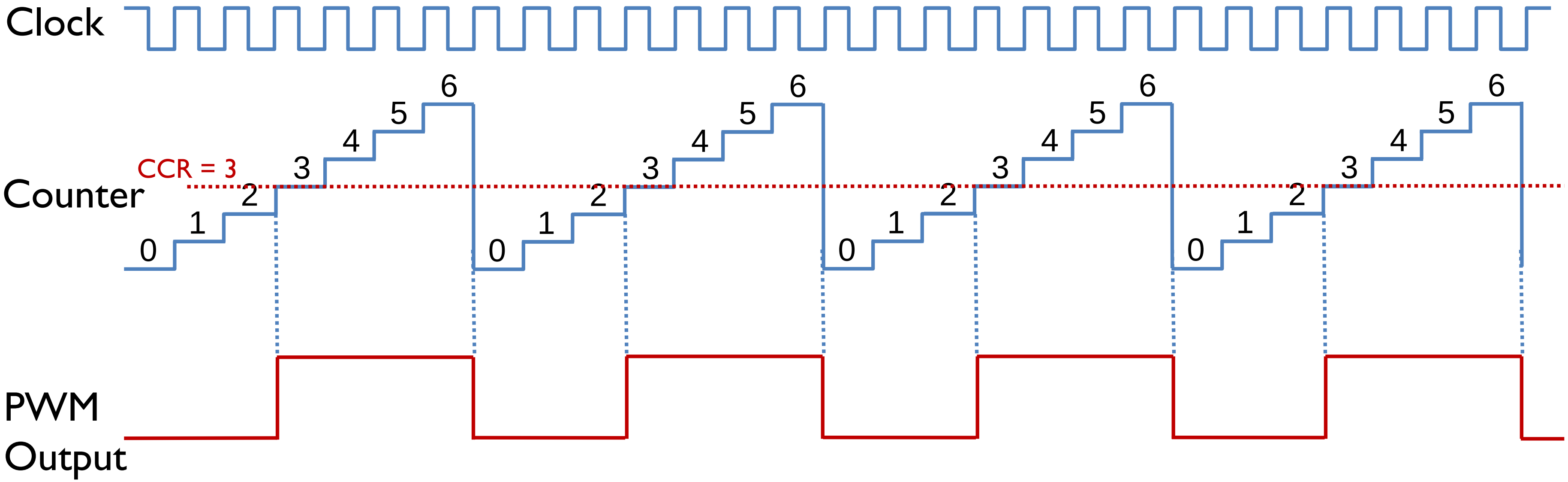
Timer Period = (ARR + 1) * Clock Period = 7 * Clock Period

PWM Mode 2 (High-True)

Mode 2

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

Up-counting mode, ARR = 6, CCR = 3



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

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

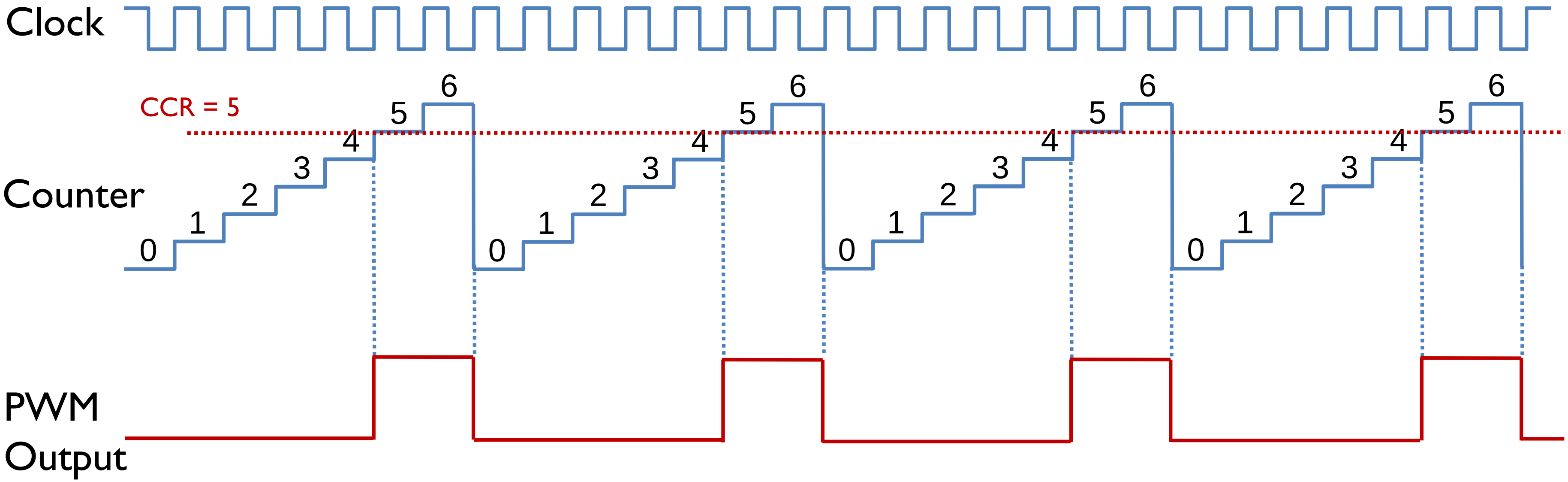
Timer Period = (ARR + 1) * Clock Period = 7 * Clock Period

PWM Mode 2 (High-True)

Mode 2

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

Up-counting mode, ARR = 6, CCR = 5



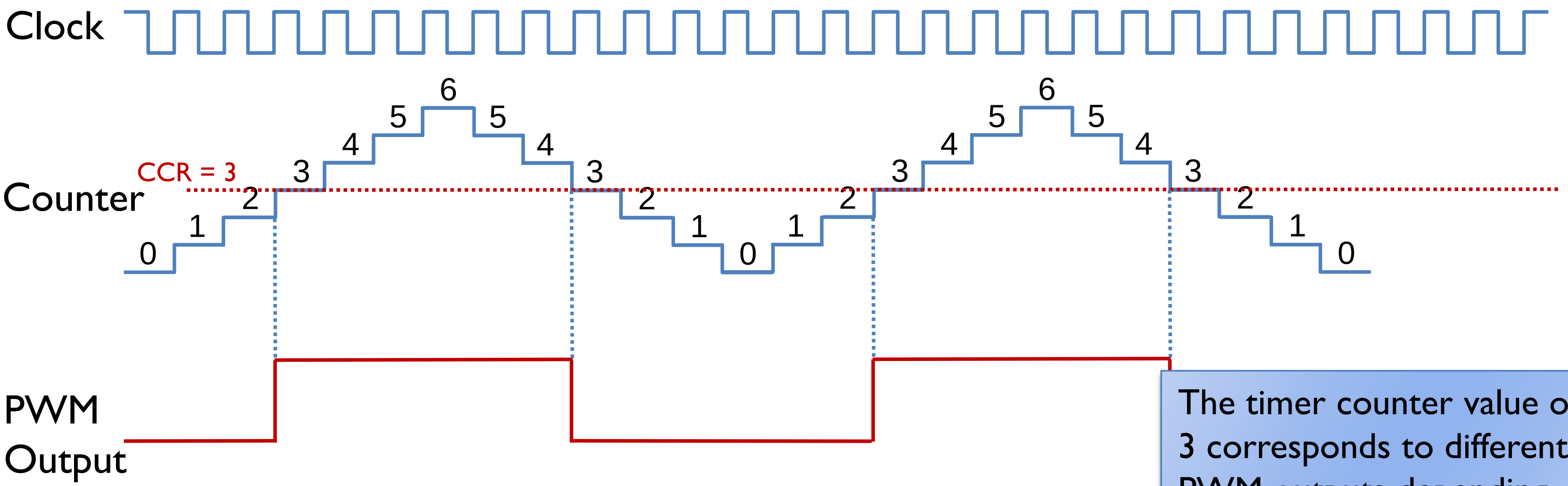
$$\text{Duty Cycle} = 1 - \frac{\text{CCR}}{\text{ARR} + 1} = \frac{2}{7}$$

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

PWM Mode 2 (High-True)

PWM Mode	Counting Mode	On	Off
Mode 1	Up-counting	CNT < CCR	CNT ≥ CCR
	Down-counting	CNT ≤ CCR	CNT > CCR
Mode 2	Up-counting	CNT ≥ CCR	CNT < CCR
	Down-counting	CNT > CCR	CNT ≤ CCR

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



The timer counter value of 3 corresponds to different PWM outputs depending on the phase (up-counting or down-counting)

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

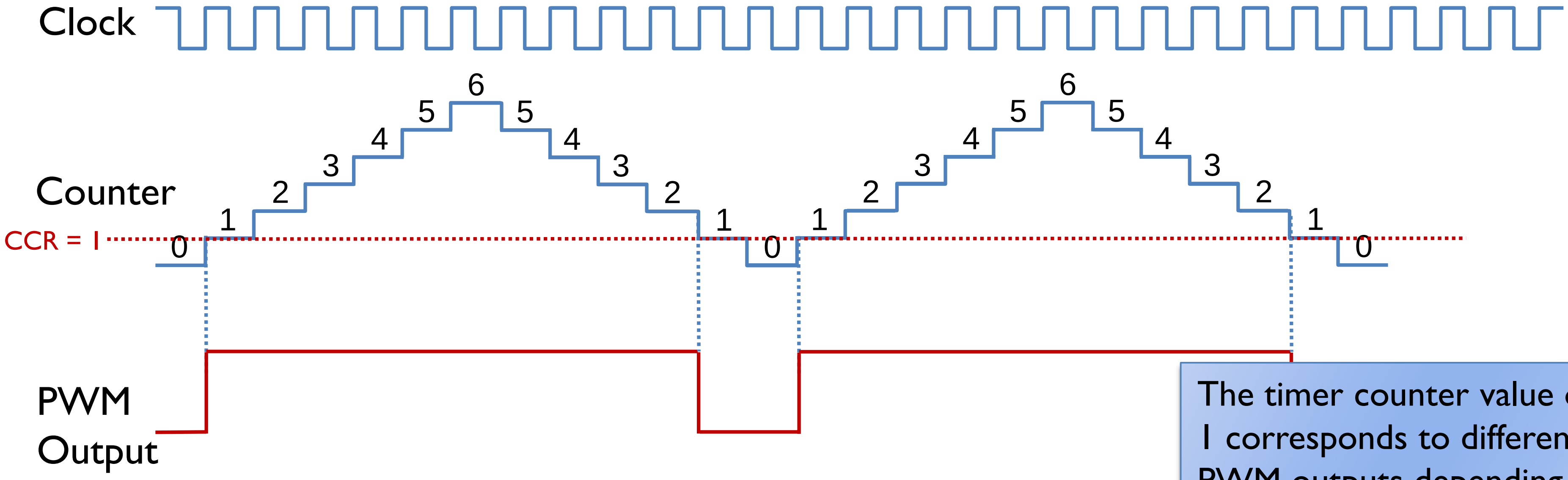
$$= \frac{1}{2}$$

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

PWM Mode 2 (High-True)

PWM Mode	Counting Mode	On	Off
Mode 1	Up-counting	CNT < CCR	CNT ≥ CCR
	Down-counting	CNT ≤ CCR	CNT > CCR
Mode 2	Up-counting	CNT ≥ CCR	CNT < CCR
	Down-counting	CNT > CCR	CNT ≤ CCR

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



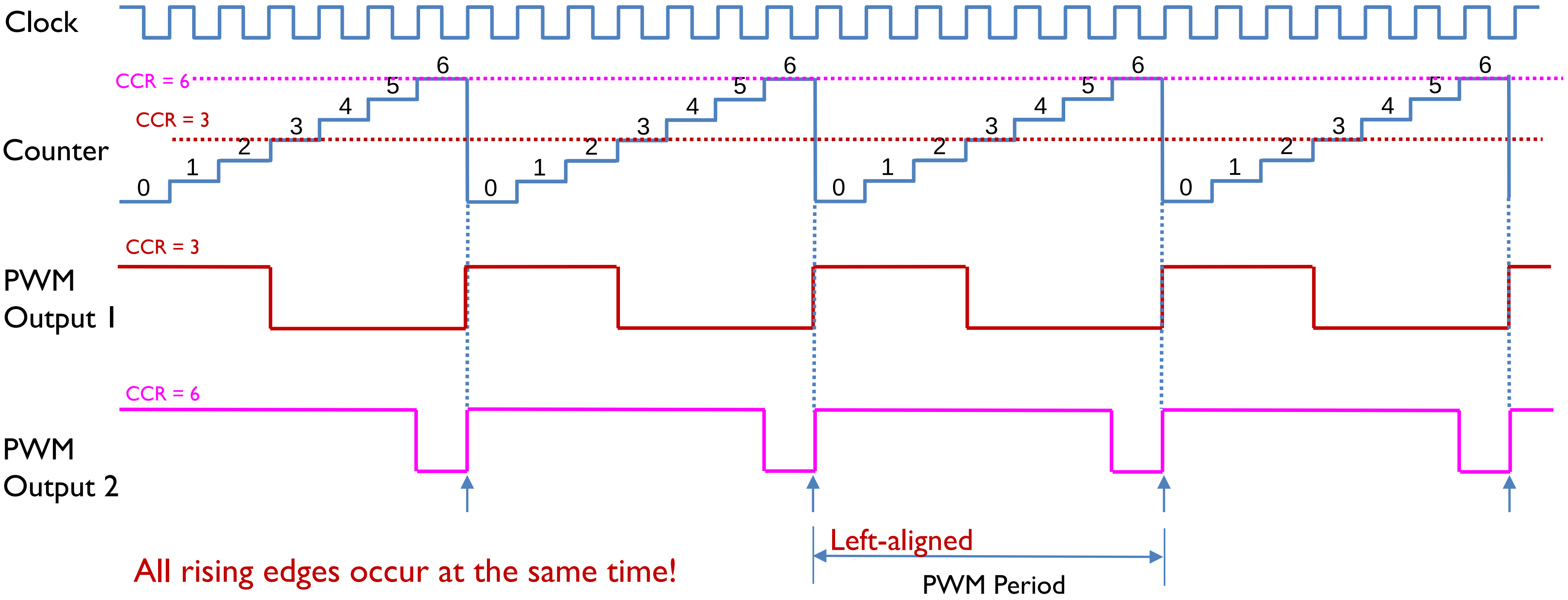
The timer counter value of 1 corresponds to different PWM outputs depending on the phase (up-counting or down-counting)

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

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

Up-Counting, PWM Mode 1: Left Edge-aligned

Upcounting mode, ARR = 6, CCR = 3

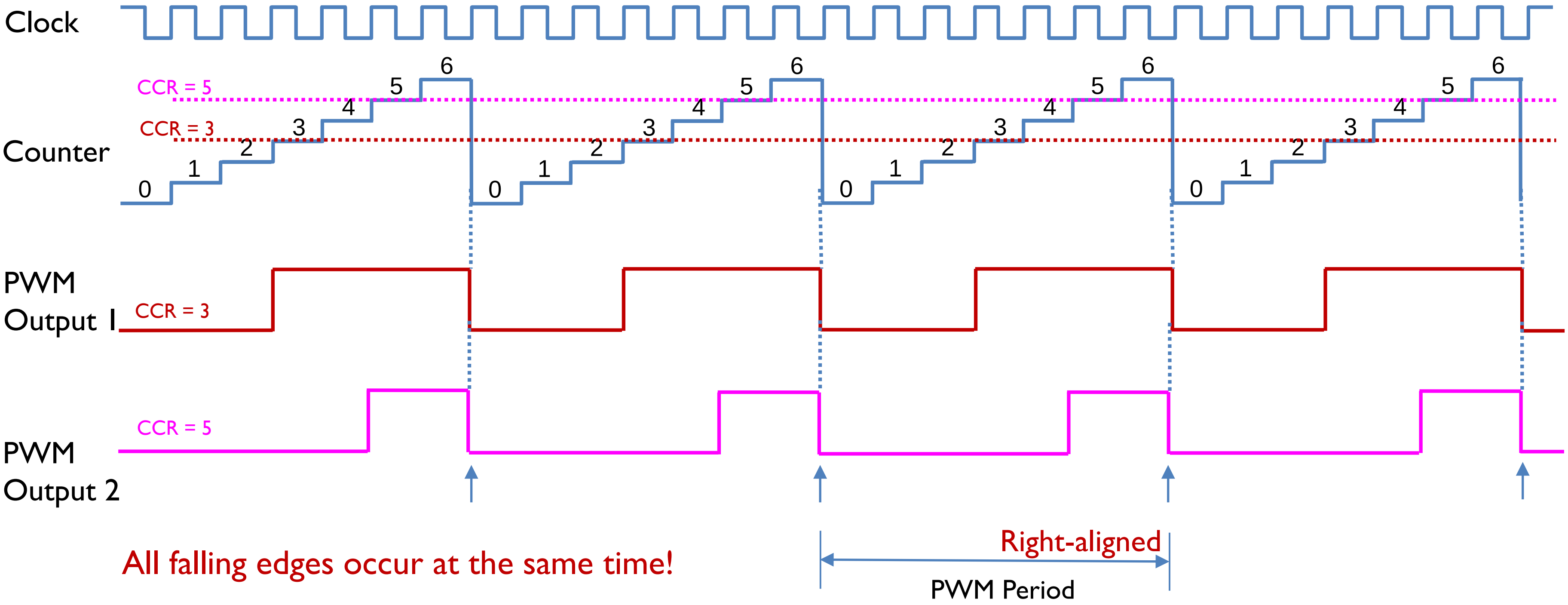


All rising edges occur at the same time!

In the up-counting mode, when multiple PWM signals are generated by the same timer, the PWM pulses are left edge aligned, because all rising edges are aligned to the left side of the PWM period.

Up-Counting, PWM Mode 2: Right Edge-aligned

Upcounting mode, ARR = 6, CCR = 3

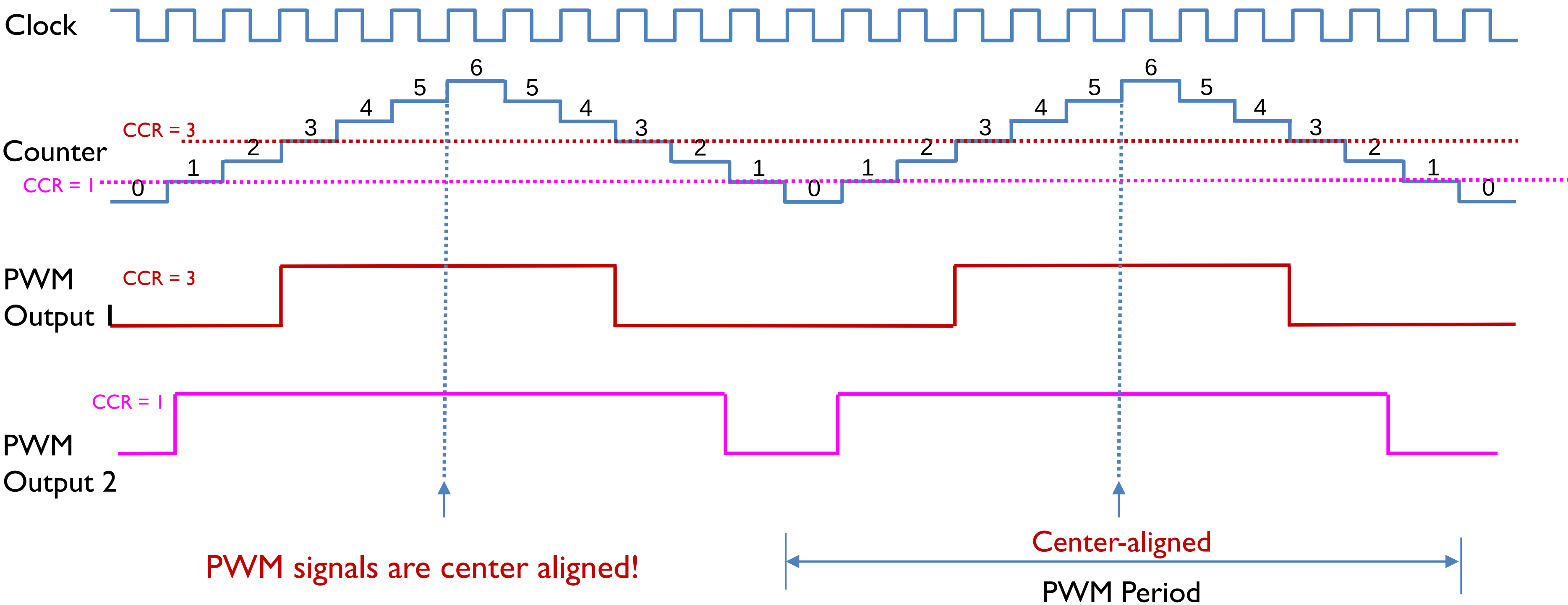


All falling edges occur at the same time!

In the down-counting mode, when multiple PWM signals are generated by the same timer, the PWM pulses are right edge aligned, because all falling edges are aligned to the right side of the PWM period.

Center-aligned Counting, PWM Mode 2: Center aligned

Center-aligned counting mode, ARR = 6, CCR = 3



PWM signals are center aligned!

In the center-aligned counting mode, when multiple PWM signals are generated by the same timer, the PWM pulses are center aligned, because their centers are aligned with peaks of the timer counter.

PWM Pulse Alignment

PWM Mode	Timer Counting Mode	Pulse Alignment
Mode 1 **(Low True)**	Up-counting	Left edge
	Down-counting	Right edge
	Center-aligned Counting	Center aligned
Mode 2 **(High True)**	Up-counting	Right edge
	Down-counting	Left edge
	Center-aligned Counting	Center aligned

Can you work out the down-counting cases by hand?

Summary of Equations

- Timer clock frequency f_{CK_CNT} vs. CPU Clock Frequency f_{CK_PSC}

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

- Timer frequency f_{Timer} with up-counting or down-counting mode:

$$f_{Timer} = \frac{f_{CK\_CNT}}{ARR + 1}; (Timer\ Period = (ARR + 1) * Clock\ Period)$$

- Timer frequency f_{Timer} with center-aligned counting mode:

$$f_{Timer} = \frac{f_{CK\_CNT}}{2 * ARR}; (Timer\ Period = (2 * ARR) * Clock\ Period)$$

- PWM duty cycle for Mode 1 (Low-True):

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

- PWM duty cycle for Mode 2 (High-True):

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

Quiz: Calculating ARR

- Suppose Timer Clock Frequency = 80MHz
- Goal: Timer Frequency = 100Hz
- What should the ARR value be for 1. up-counting mode; 2. down-counting mode; 3. center-aligned counting mode?

Quiz Answer: Calculating ARR

- Suppose Timer Clock Frequency = 80MHz
- Goal: Timer Frequency = 100Hz
- What should the ARR value be for 1. up-counting mode; 2. down-counting mode; 3. center-aligned counting mode?

For up-counting or down-counting mode:

$$f_{Timer} = \frac{f_{CK\_CNT}}{ARR + 1} = \frac{80MHz}{ARR + 1} = 100Hz$$

$$ARR = 799999$$

For center-aligned counting mode:

$$f_{Timer} = \frac{f_{CK\_CNT}}{2 * ARR} = \frac{80MHz}{2 * ARR} = 100Hz$$

$$ARR = 400000$$

But a 16-bit ARR register has value range of [0, 65535], so ARR value of 799999 or 400000 is out of range → cannot generate a 10ms timer from a 80MHz Timer Clock!
Solution: use prescaler (PSC) to reduce Timer Clock Frequency

Quiz: Calculating ARR with prescaler

- Suppose CPU Clock Frequency = `80MHz, PSC=79`
- Goal: Timer Frequency = 100Hz
- What should the ARR value be for 1. up-counting mode; 2. down-counting mode; 3. center-aligned counting mode?

Quiz Answer: Calculating ARR with prescaler

- Suppose CPU Clock Frequency = 80MHz， prescaler PSC=79
- Goal: Timer Frequency = 100Hz
- What should the ARR value be for 1. up-counting mode; 2. down-counting mode; 3. center-aligned counting mode?

$$Timer\ Clock\ Freq\ f_{CK\_CNT} = \frac{f_{CK_{PSC}}}{PSC + 1} = \frac{80MHz}{80} = 1MHz$$

For up-counting or down-counting mode:

$$f_{Timer} = \frac{f_{CK\_CNT}}{ARR + 1} = \frac{1MHz}{ARR + 1} = 100Hz$$

$$ARR = 9999$$

For center-aligned counting mode:

$$f_{Timer} = \frac{f_{CK\_CNT}}{2 * ARR} = \frac{1MHz}{2 * ARR} = 100Hz$$

$$ARR = 5000$$

With prescaler, the ARR value of 9999 or 5000 is now within the range of 16-bit ARR register.

Quiz

- Suppose a 16-bit timer has the following settings
  - CPU clock frequency is 4 MHz.
  - Prescaler PSC = 39
  - Counting direction: center-aligned counting
  - Desired timer frequency = 100 Hz
- Calculate the ARR.

Quiz Answer

- Suppose a 16-bit timer has the following settings
  - CPU clock frequency is 4 MHz.
  - Prescaler PSC = 39
  - Counting direction: center-aligned counting
  - Desired timer frequency = 100 Hz
- Calculate the ARR.

$$Timer\ Clock\ Freq\ f_{CK\_CNT} = \frac{f_{CK\_PSC}}{PSC+1} = \frac{4\ \mathrm{MHz}}{40} = 0.1\ \mathrm{MHz}$$

$$Timer\ Freq\ fTim_{er} = \frac{f_{CK\_CNT}}{2 * ARR} = \frac{0.1\ MHz}{2 * \mathrm{ARR}} = 100Hz$$

$$ARR = 500$$

Quiz

- Suppose all registers are 16 bits. CPU clock frequency is 400 MHz.
- What is the maximum and minimum Timer Clock Frequency, and Timer Frequency, assuming up-counting mode?

Quiz Answer

- Suppose all registers are 16 bits. CPU clock frequency is 400 MHz.
- What is the maximum and minimum Timer Clock Frequency, and Timer Frequency, assuming up-counting mode?
- PSC, ARR are both in the range [0, 2^{16}-1=65535].

$$f_{CK\_CNT} = \frac{f_{CK\_PSC}}{PSC+1} = \frac{400\text{ MHz}}{\text{PSC+1}} \in \left[\frac{400\text{MHz}}{65536}, \frac{400\text{MHz}}{1}\right] = [6.1\text{ KHz}, 400\text{MHz}]$$

$$f_{Timer} = \frac{f_{CK\_CNT}}{ARR+1} \in \left[\frac{6.1KHz}{65536}, \frac{400MHz}{1}\right] = [0.09Hz, 400\ MHz]$$

- (A realistic Timer Frequency should not be higher than 1KHz, as mentioned before.)

Quiz

- Suppose a 16-bit timer has the following settings
  - CPU Clock frequency is 16 MHz.
  - Prescaler PSC = 159
  - ARR = 1999
  - CCR = 499
  - Counting direction: up-counting
  - Output is set as PWM Mode 2 (High True).
- Calculate the timer frequency and PWM duty cycle.

Quiz Answer

- Suppose a 16-bit timer has the following setting
  - CPU clock frequency is 16 MHz.
  - Prescaler PSC = 159
  - ARR = 1999
  - CCR = 499
  - Counting direction: up-counting
  - Output is set as PWM Mode 2 (High True).
- Calculate the timer frequency and PWM duty cycle.

$$Timer\ Clock\ Freq\ f_{CK\_CNT} = \frac{f_{CK\_PSC}}{PSC+1} = \frac{16\ MHz}{159+1} = 0.1MHz$$

$$Timer\ Freq\ fTim_{er} = \frac{f_{CK\_CNT}}{ARR+1} = \frac{0.1\ MHz}{1999+1} = 50Hz$$

$$Duty\ Cycle = 1 - \frac{CCR}{ARR+1} = 1 - \frac{499}{1999+1} = \frac{1501}{2000} = 75.05\%$$

PWM output is high on when the counter is 499, 500, 501, …, 1999, a total of 1501 cycles. (This statement is not required in the exam.)