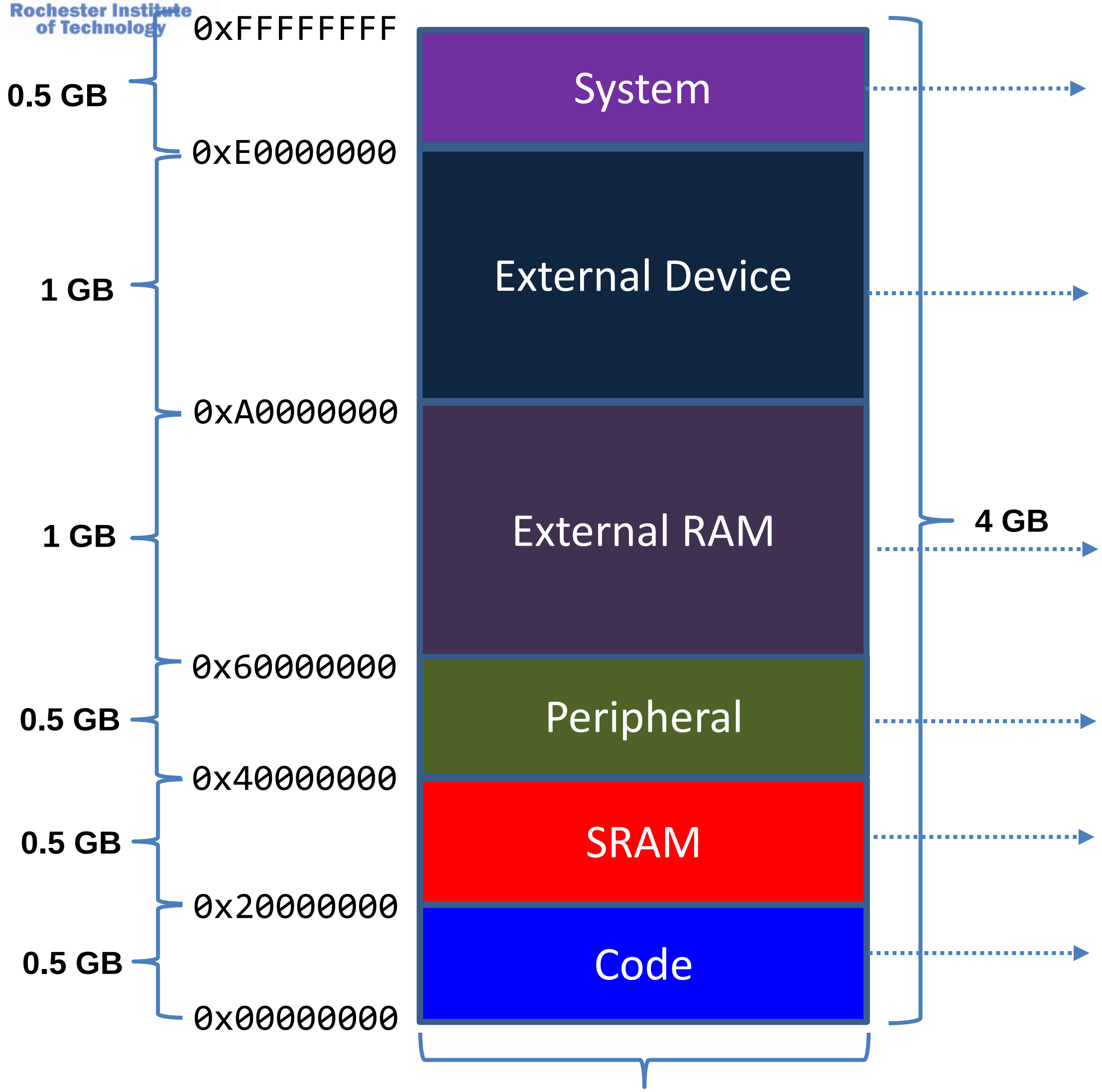
Software Engineering
Rochester Institute
of Technology

Real Time & Embedded Systems

STM32 GPIO and Timers

GPIO

Memory Map of Cortex-M4



NVIC, System Timer, SCB, vendor-specific memory

Such as SD card

Off-chip memory for data

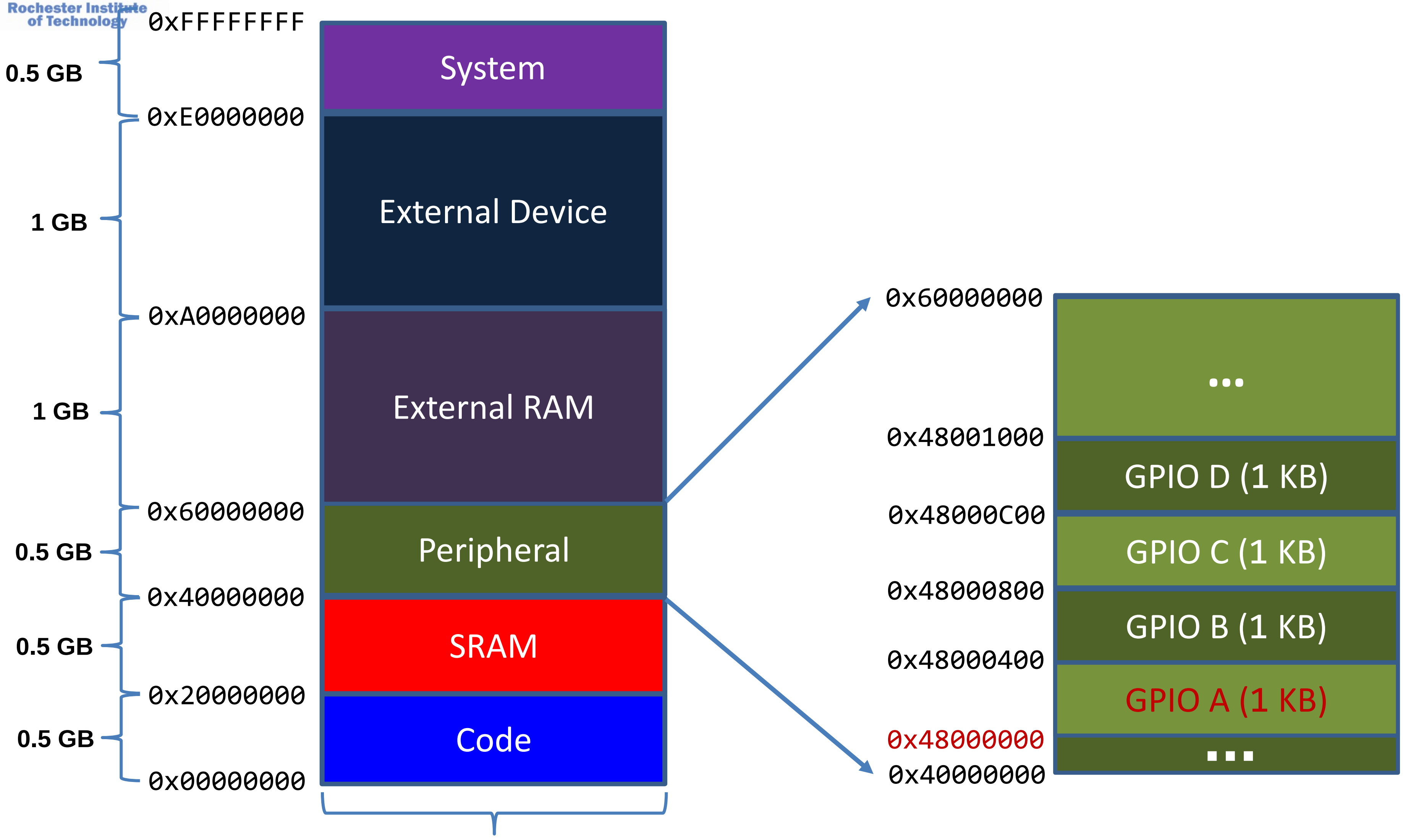
AHB & APB, such as timers, GPIO

On-chip RAM, for heap, stack, & coc

On-chip Flash, for code & data

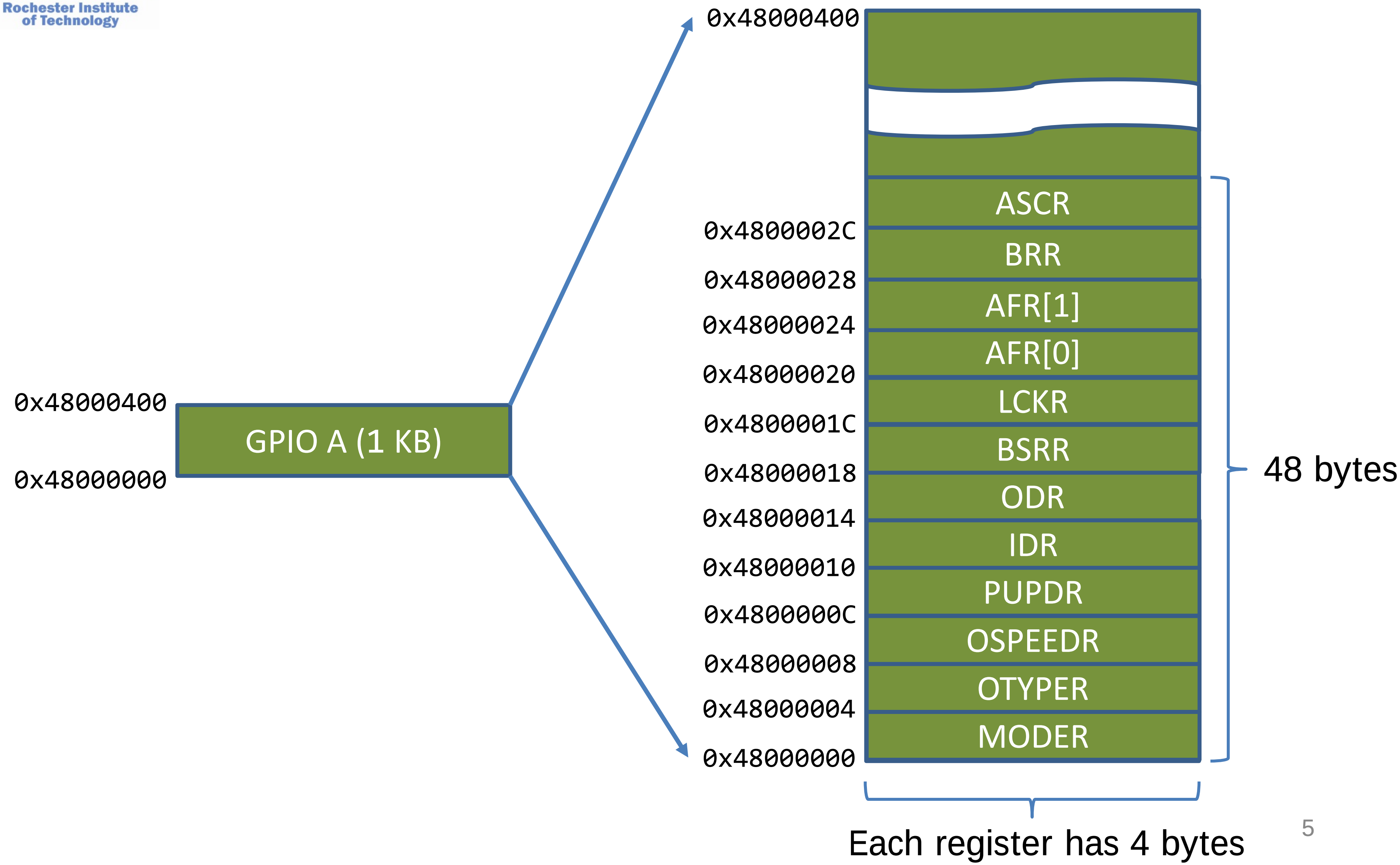
One Byte (8 bits)

Memory Map of STM32L4



One Byte (8 bits)

GPIO Memory Map



0x48000400

0x48000000

GPIO A (1 KB)

0x48000400

ASCR

0x4800002C

BRR

0x48000028

AFR[1]

0x48000024

AFR[0]

0x48000020

LCKR

0x4800001C

BSRR

0x48000018

ODR

0x48000014

IDR

0x48000010

PUPDR

0x4800000C

OSPEEDR

0x48000008

OTYPER

0x48000004

MODER

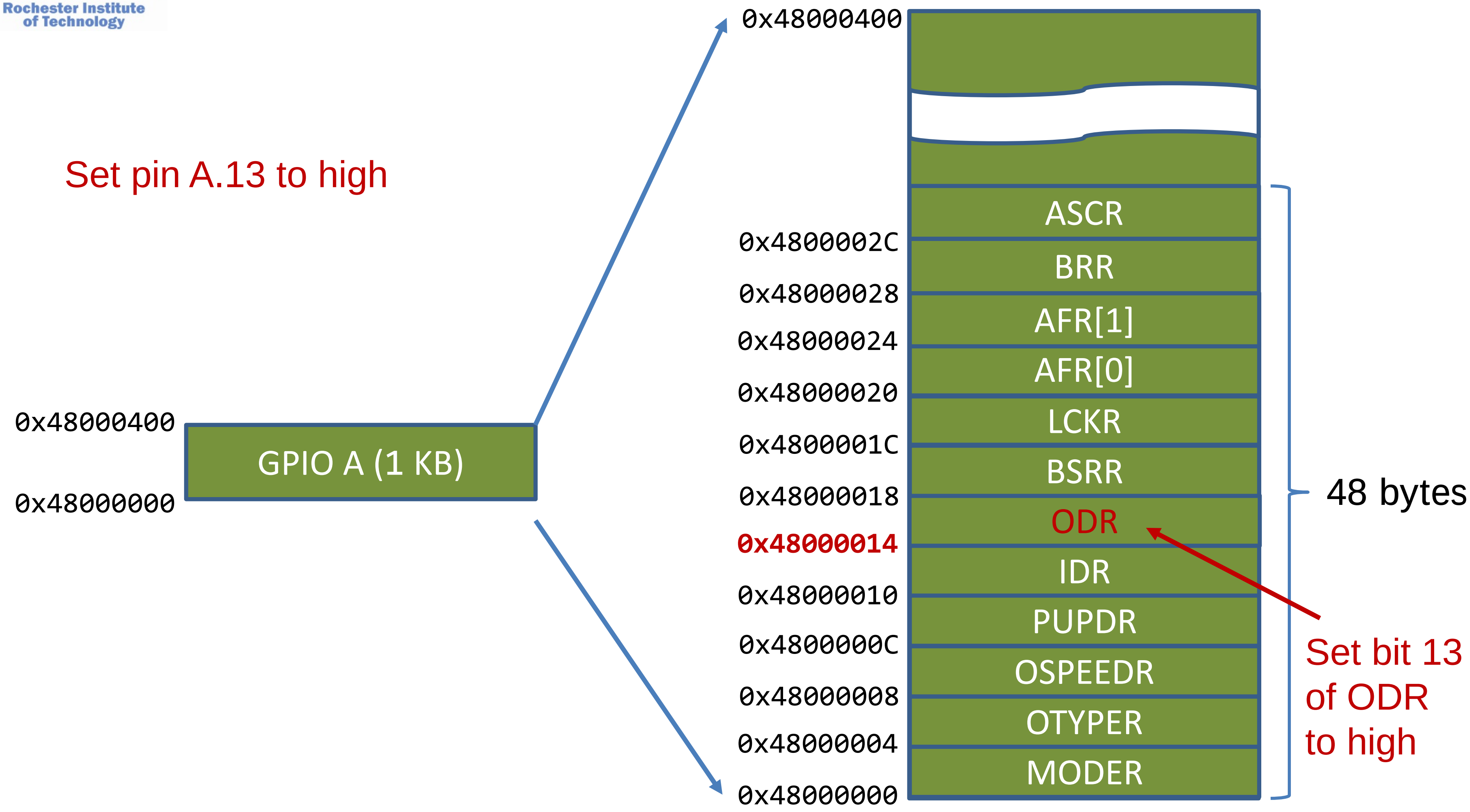
0x48000000

48 bytes

Each register has 4 bytes

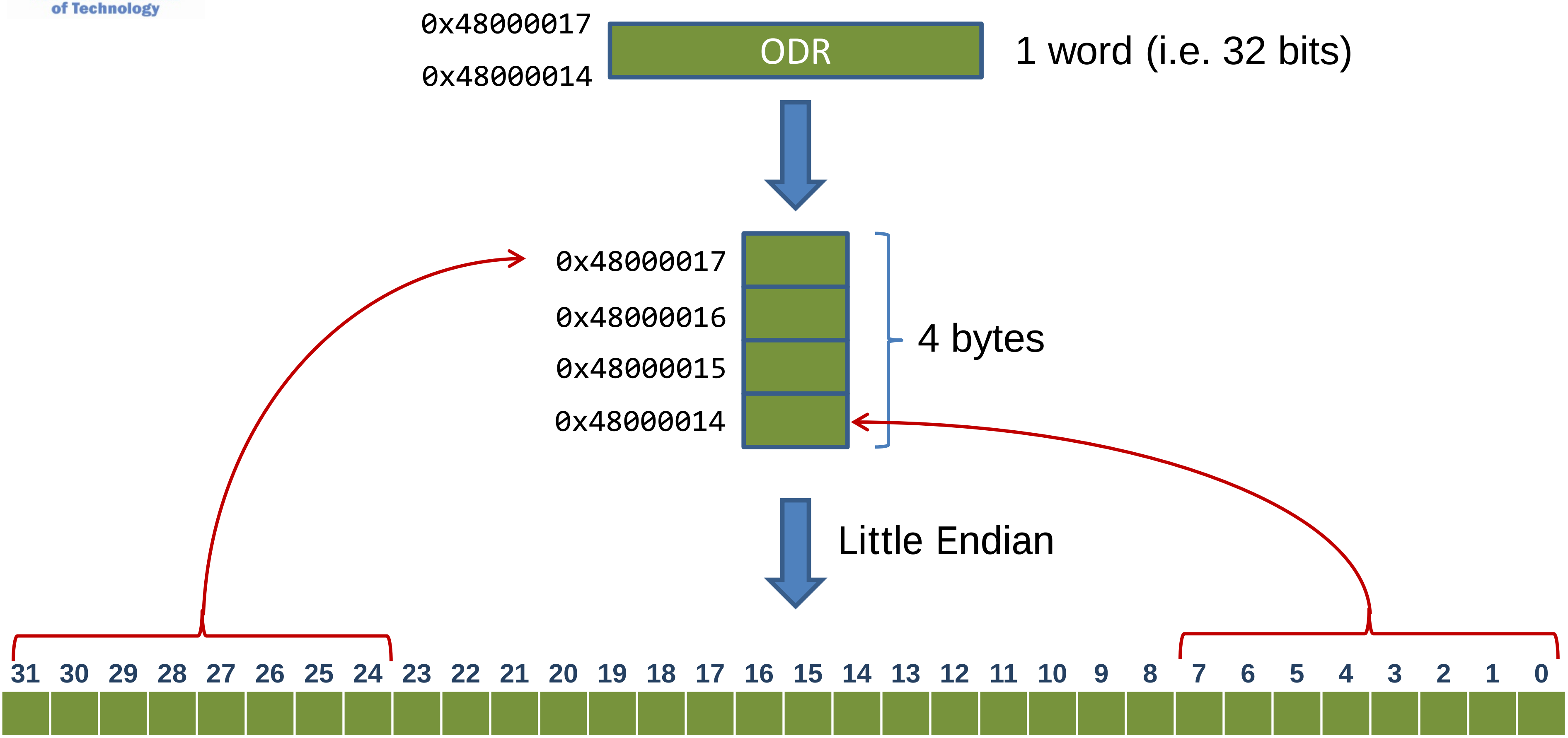
GPIO Memory Map

Set pin A.13 to high

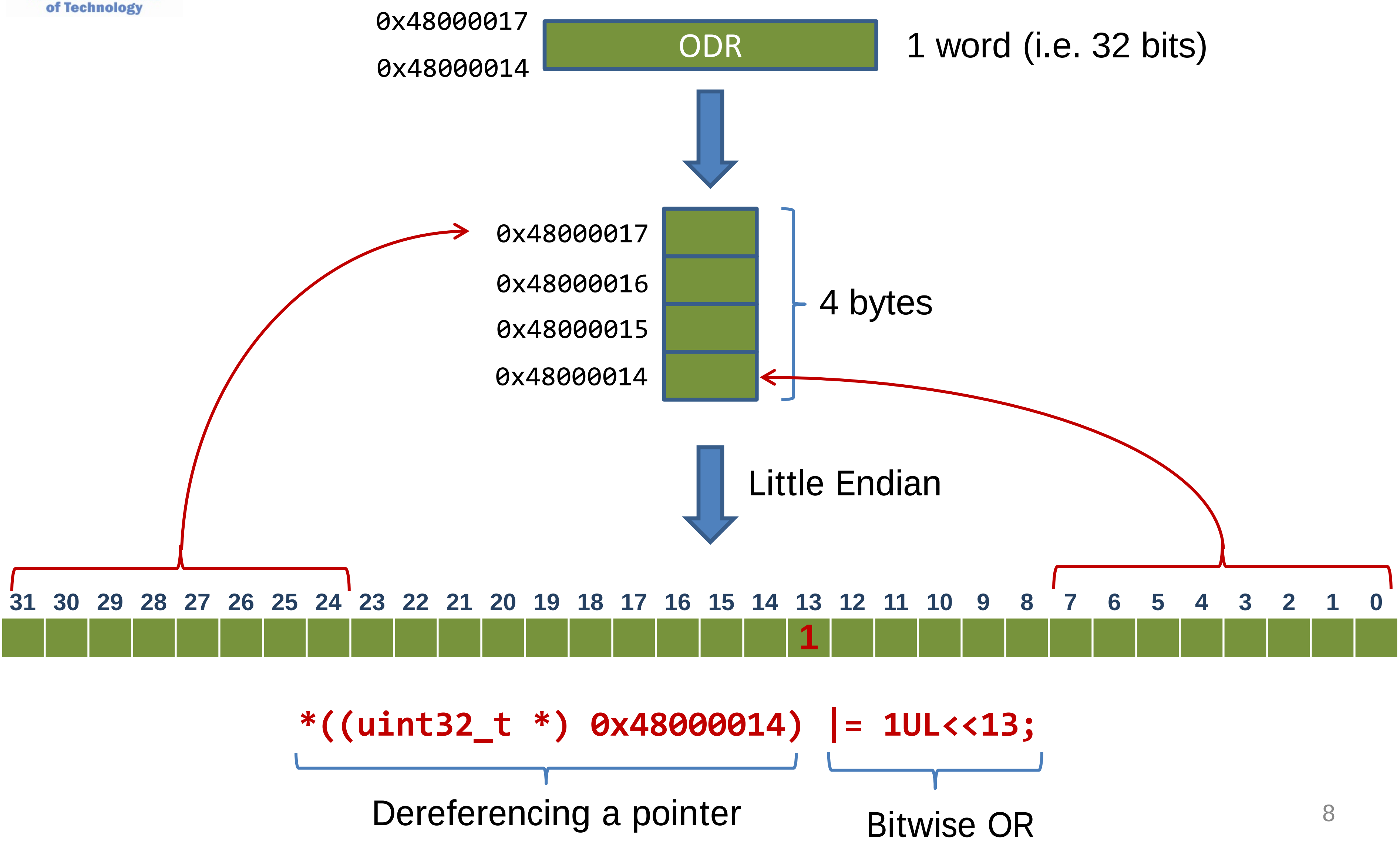


Set bit 13 of ODR to high

Output Data Register (ODR)



Output Data Register (ODR)



0x48000017

0x48000014

ODR

1 word (i.e. 32 bits)

0x48000017

0x48000016

0x48000015

0x48000014

4 bytes

Little Endian

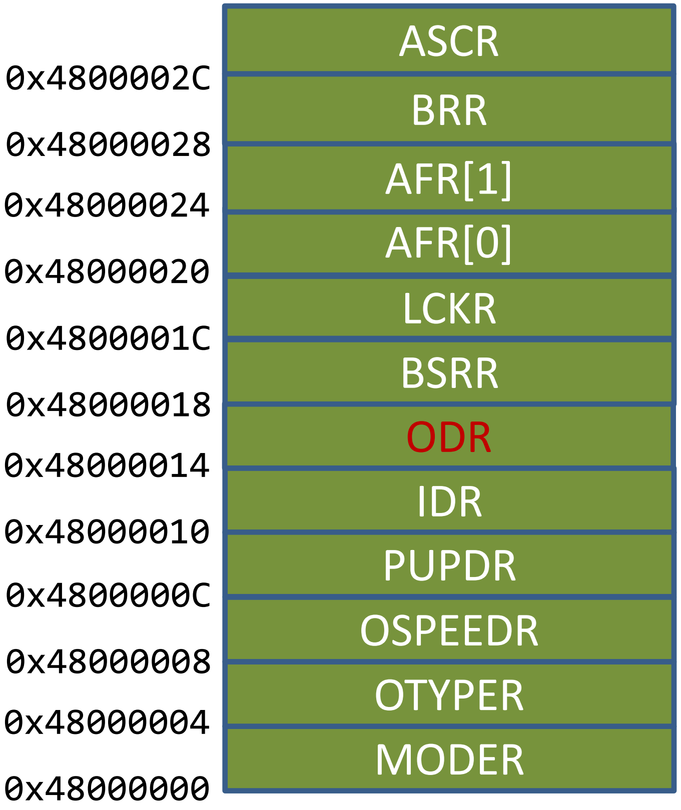
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																		1													

```
*((uint32_t *) 0x48000014) |= 1UL<<13;
```

Dereferencing a pointer

Bitwise OR

Dereferencing a Memory Address



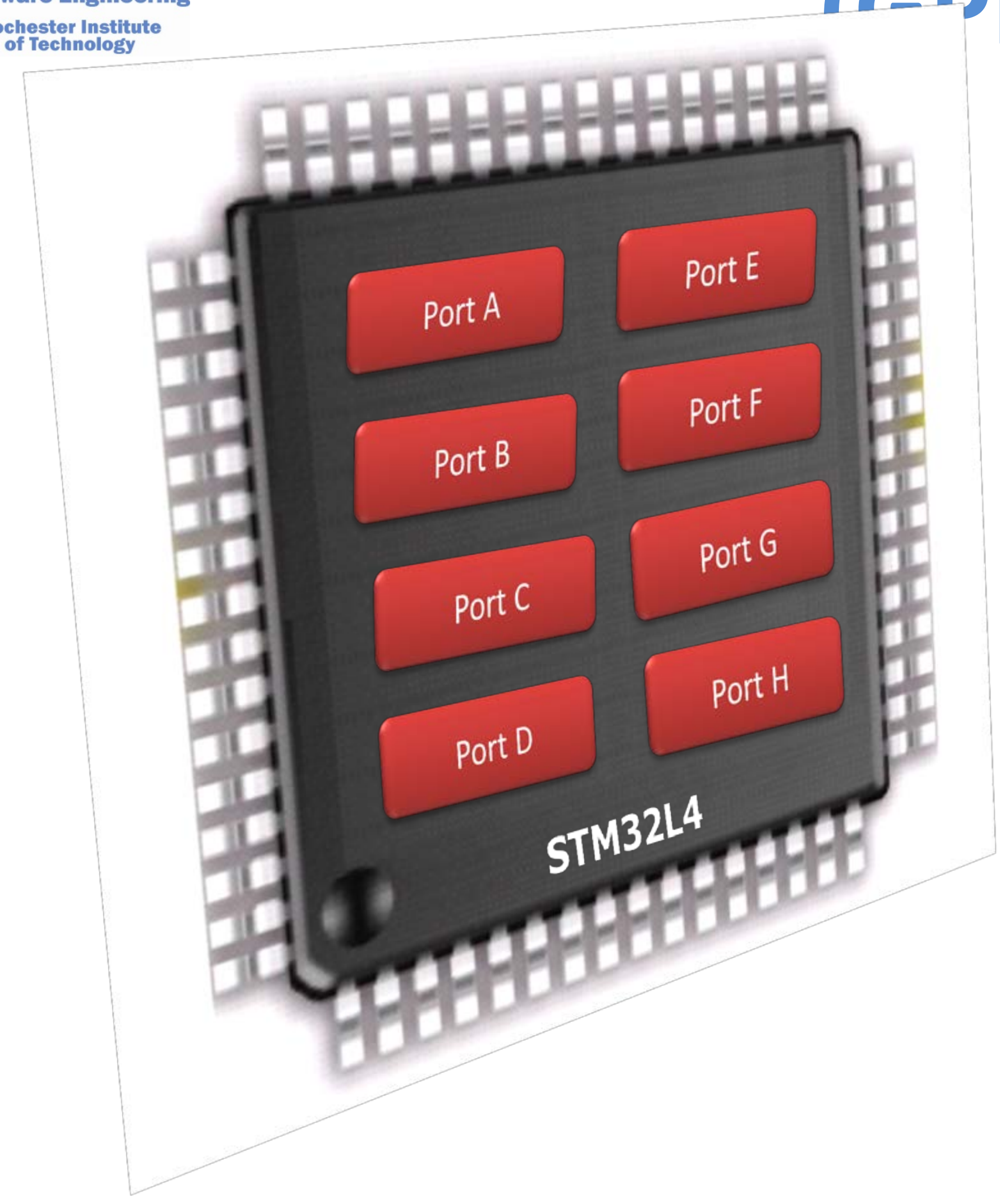
ASCR
0x4800002C
BRR
0x48000028
AFR[1]
0x48000024
AFR[0]
0x48000020
LCKR
0x4800001C
BSRR
0x48000018
ODR
0x48000014
IDR
0x48000010
PUPDR
0x4800000C
OSPEEDR
0x48000008
OTYPER
0x48000004
MODER
0x48000000

```
typedef struct {
  volatile uint32_t MODER;    // Mode register
  volatile uint32_t OTYPER;   // Output type register
  volatile uint32_t OSPEEDR;  // Output speed register
  volatile uint32_t PUPDR;    // Pull-up/pull-down register
  volatile uint32_t IDR;      // Input data register
  volatile uint32_t ODR;      // Output data register
  volatile uint32_t BSRR;     // Bit set/reset register
  volatile uint32_t LCKR;     // Configuration lock register
  volatile uint32_t AFR[2];   // Alternate function registers
  volatile uint32_t BRR;      // Bit Reset register
  volatile uint32_t ASCR;     // Analog switch control register
} GPIO_TypeDef;

// Casting memory address to a pointer
#define GPIOA ((GPIO_TypeDef *) 0x48000000)
```

```
GPIOA->ODR |= 1UL<<13;
```

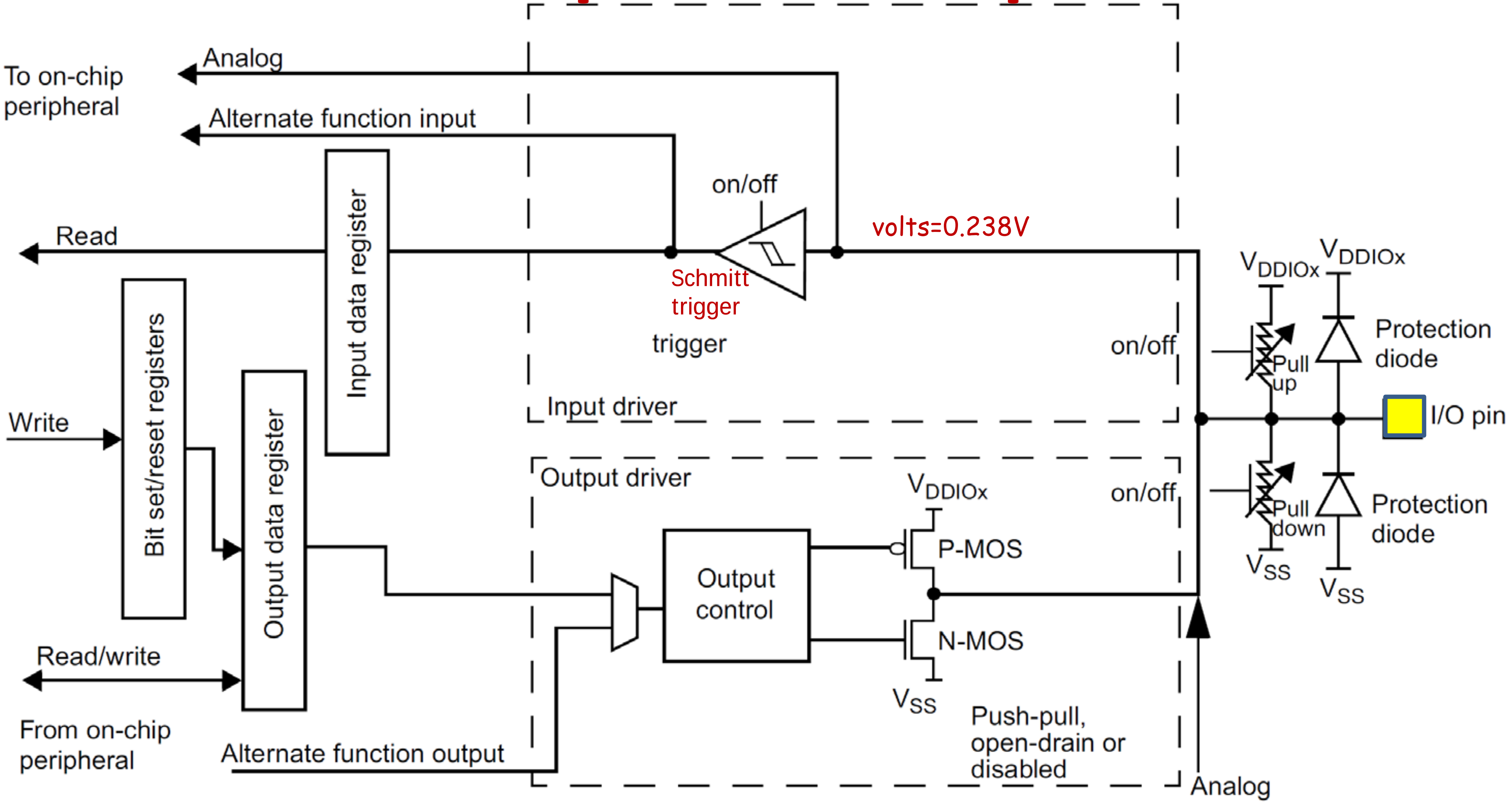
General Purpose Input/Output (GPIO)



- 8 GPIO Ports:
  A, B, C, D, E, F, G, H
- Up to 16 pins in each port

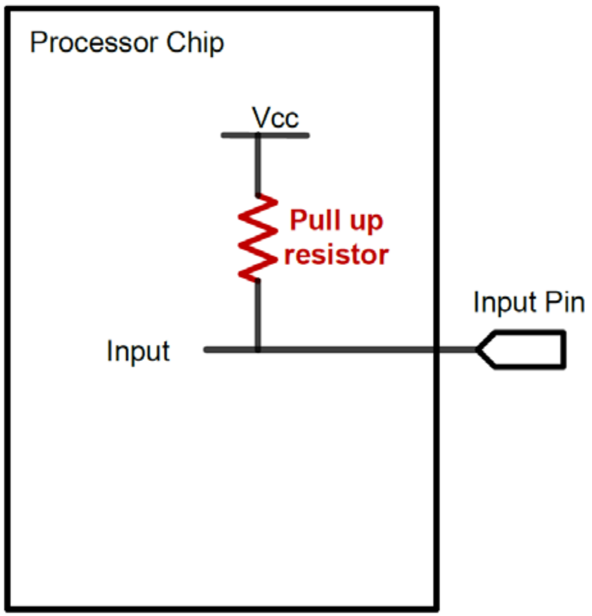
Basic Structure of an I/O Port Bit

Input and Output



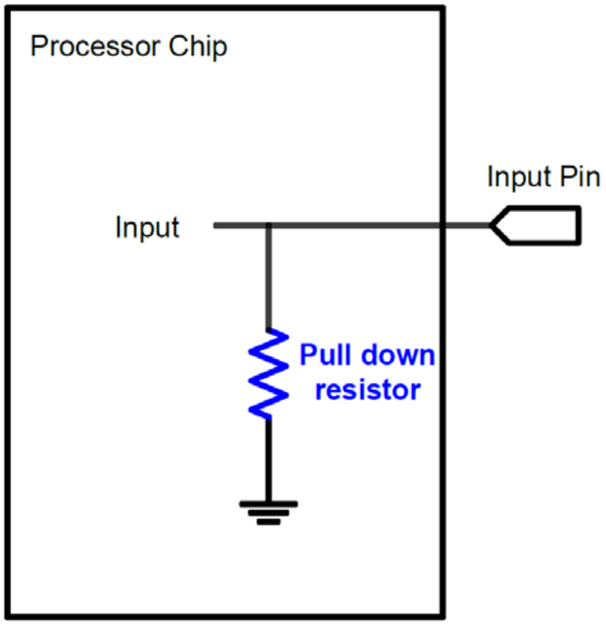
GPIO Input: Pull Up and Pull Down

- A digital input can have three states: High, Low, and High-Impedance (also called floating, tri-stated, HiZ)



Pull-Up

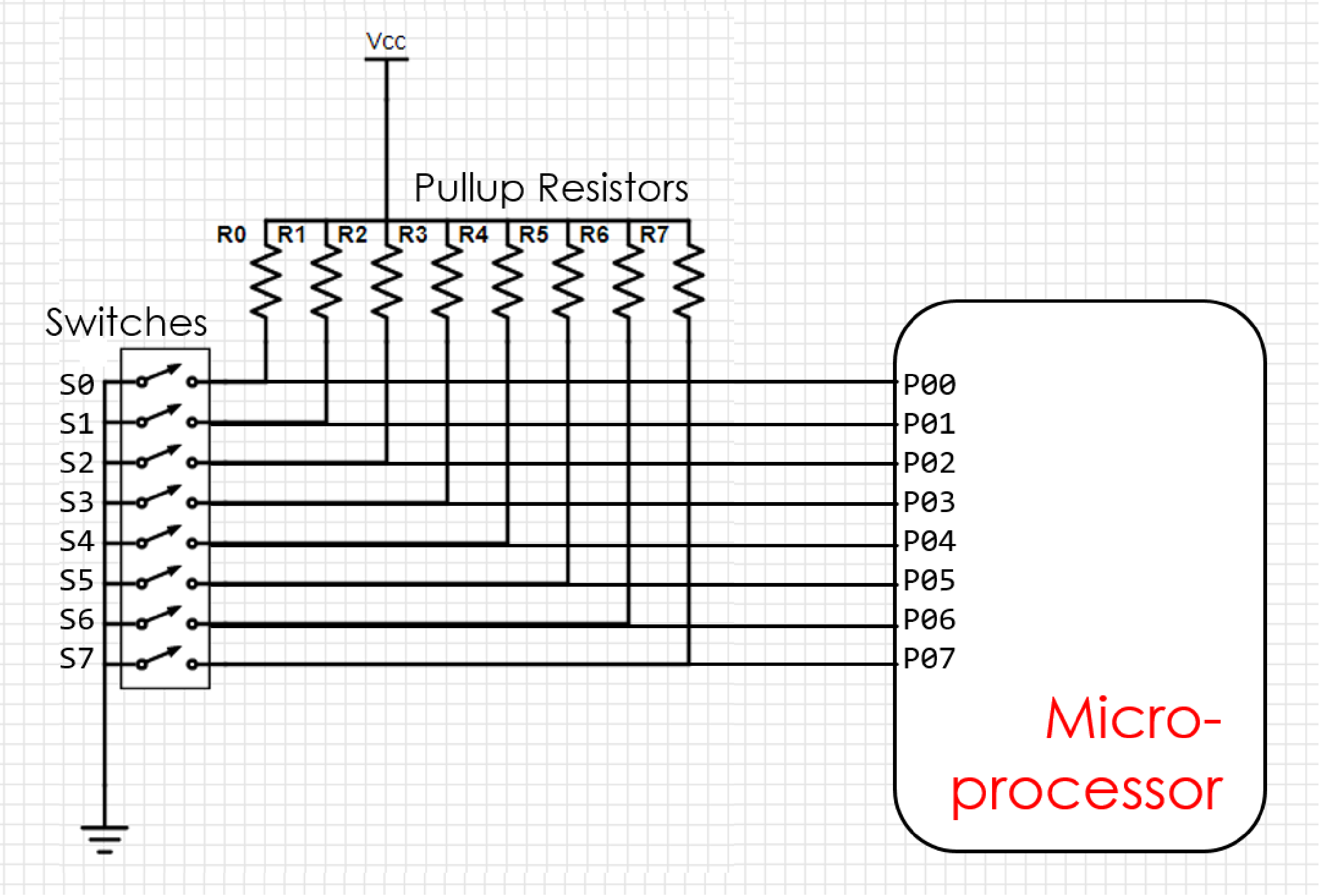
If external input is HiZ, the input is read as a valid HIGH.



Pull-Down

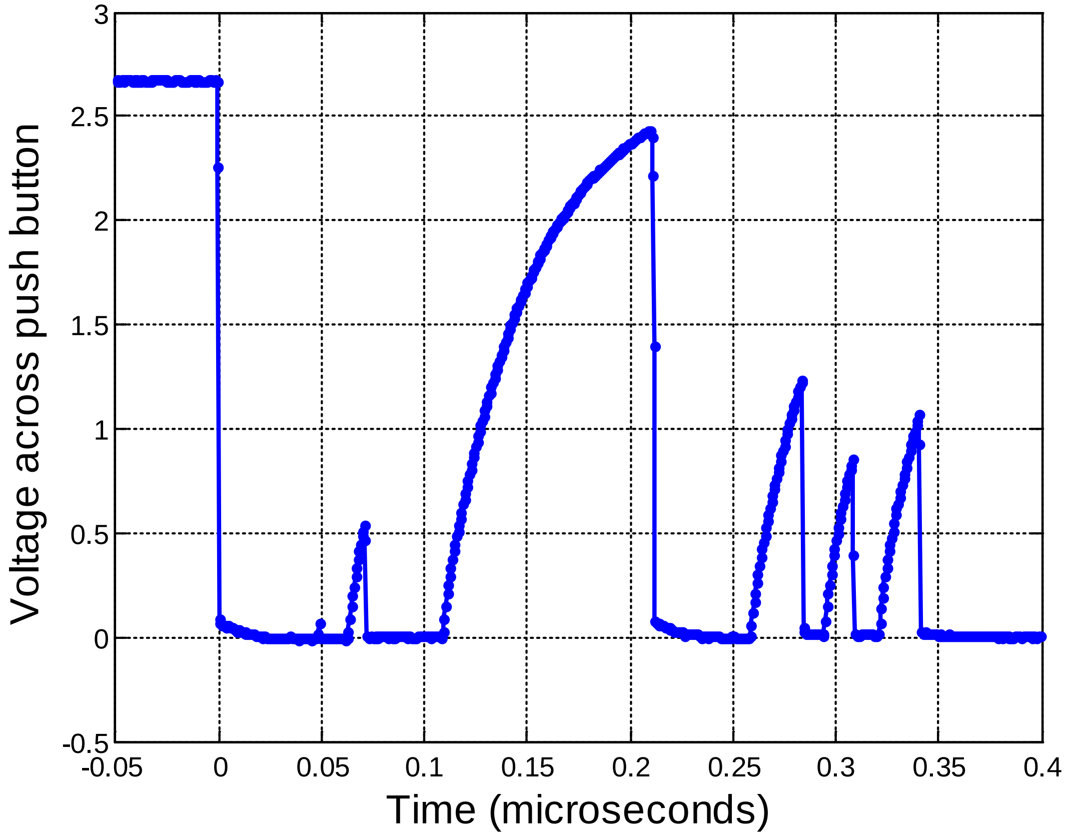
If external input is HiZ, the input is read as a valid LOW.

Buttons / switches

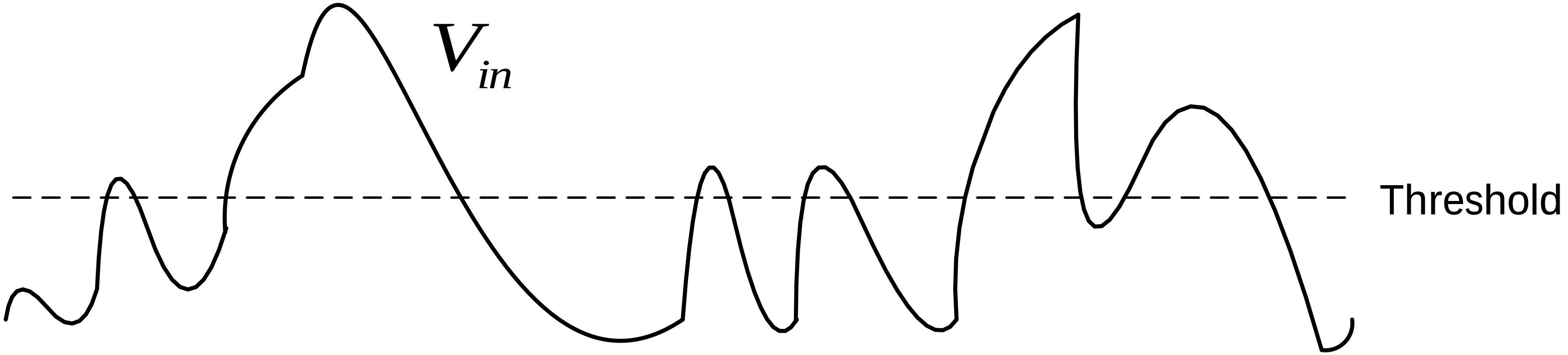


I/O Debouncing

- Example signal when a button is pressed



Noisy analog signals …

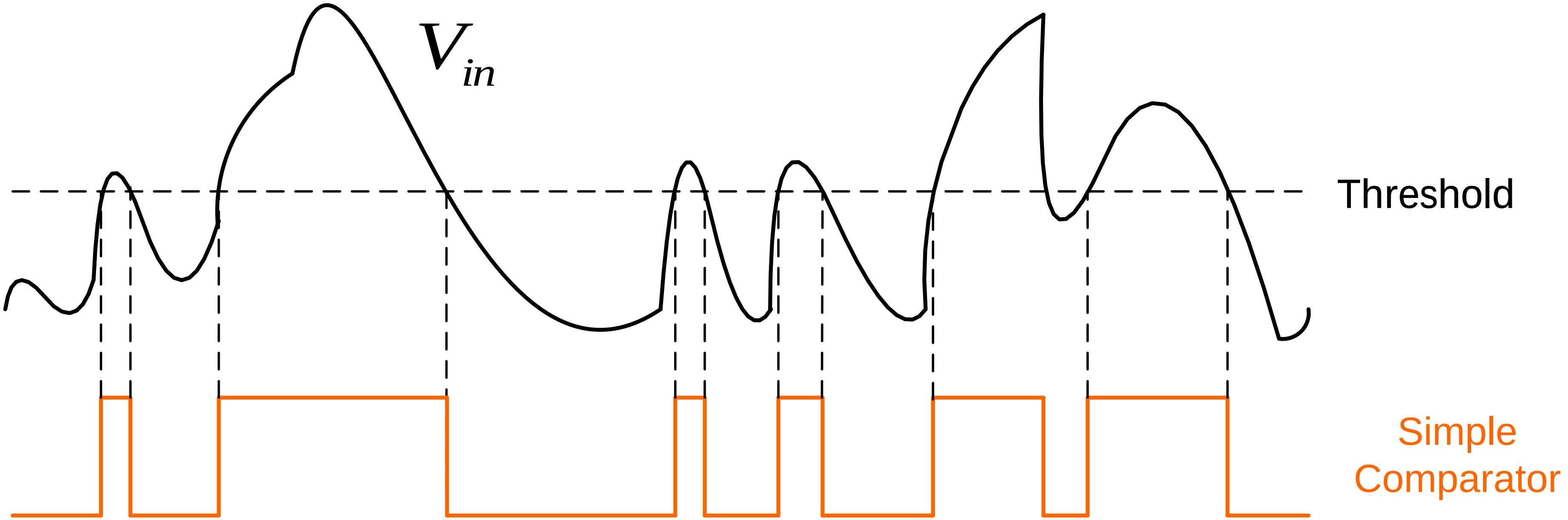


Analog signals

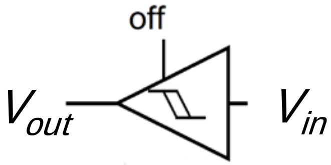
- Noisy
- Rise and fall slowly (small slew rate)
- A button press can generate a noisy transition for up to 20 msec if not debounced in hardware

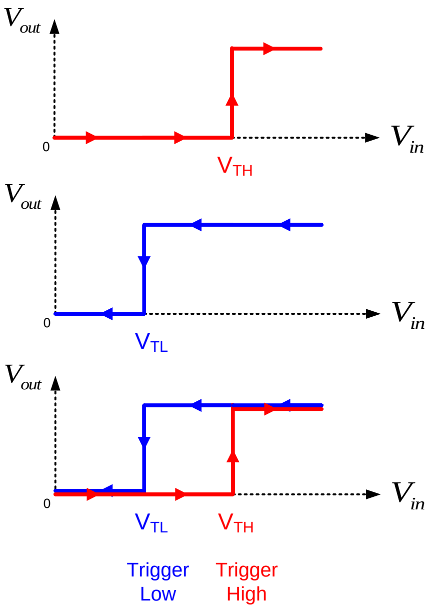
Software Engineering
Rochester Institute of Technology

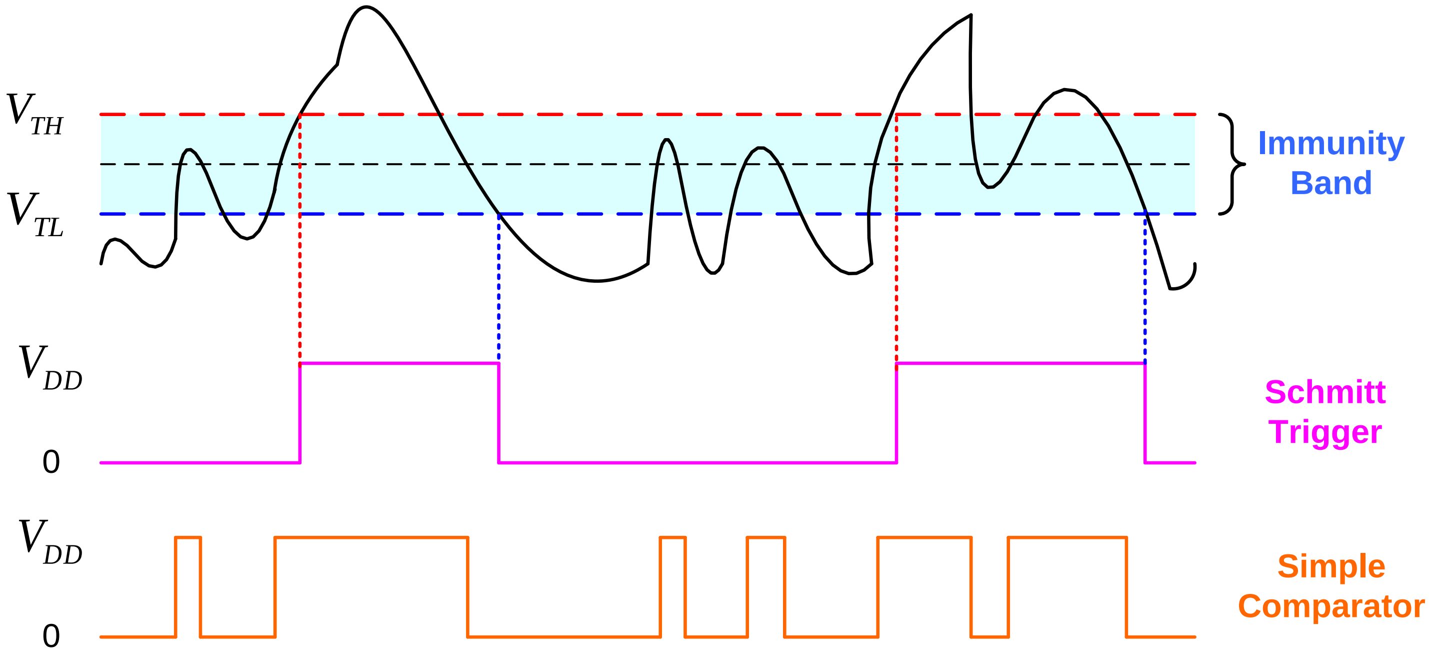
… produce noisy digital signals



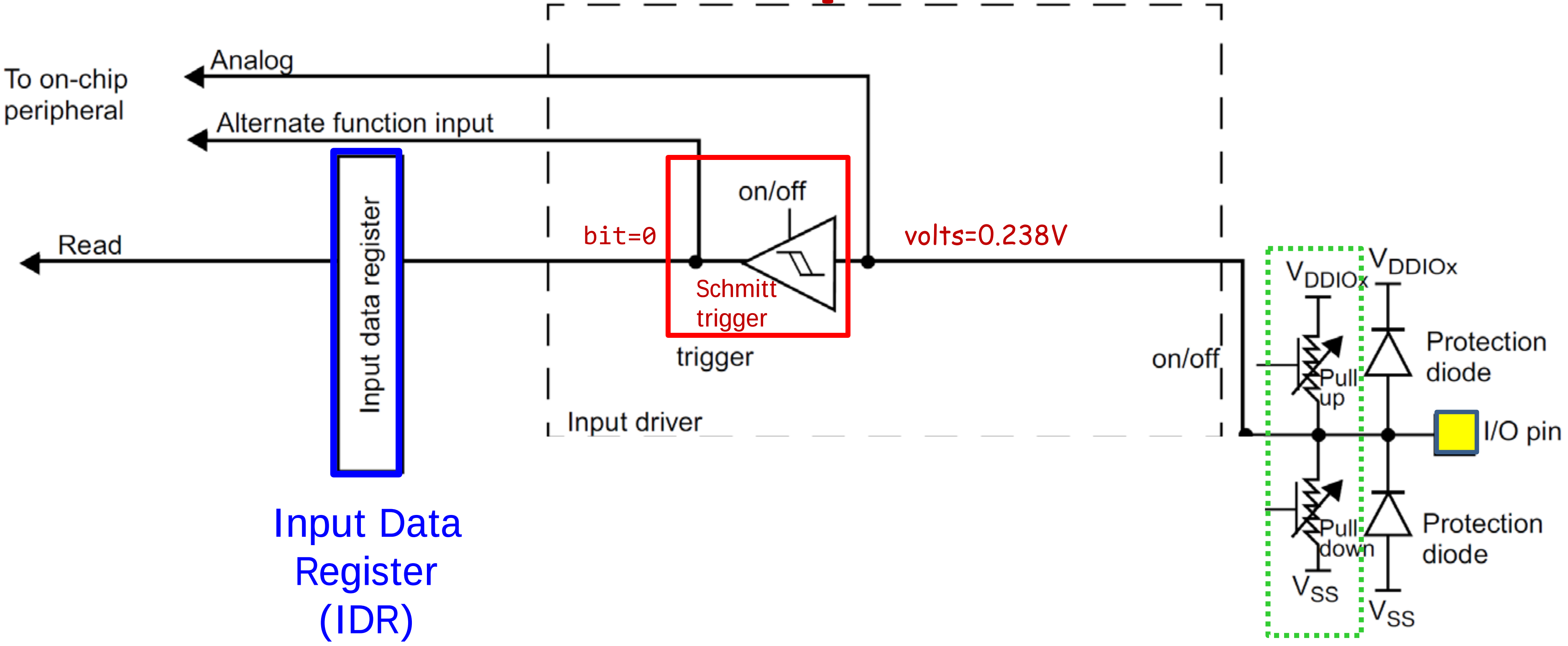
Schmitt Trigger







Basic Structure of an I/O Port Bit: Input



Input Data Register (IDR)

GPIO Pull-up/Pull-down Register (PUPDR)

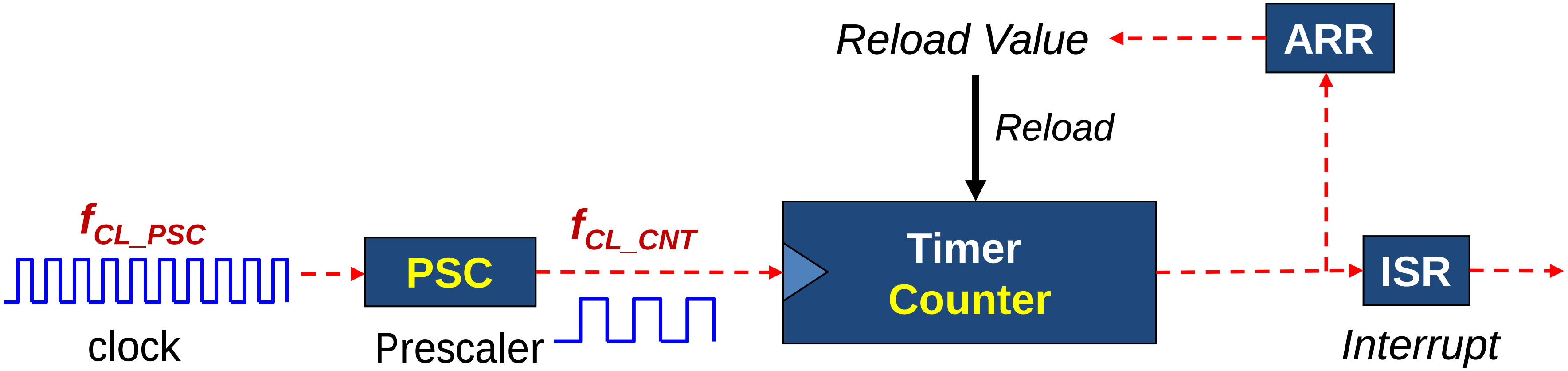
00 = No pull-up, pull-down	01 = Pull-up
10 = Pull-down	11 = Reserved

TIMERS

Timer

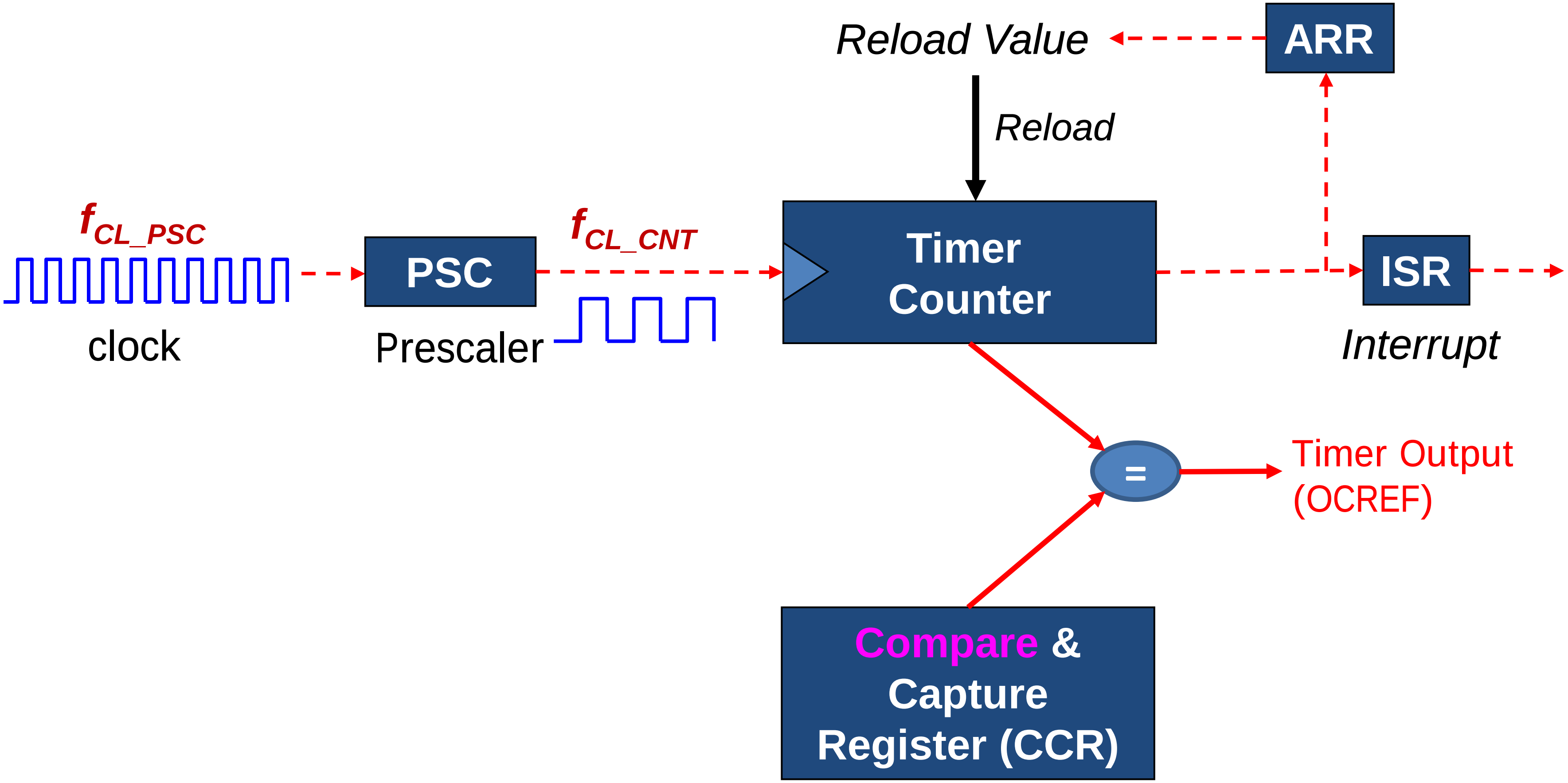
- Free-run counter (independent of processor)
- Functions
  - **Input capture**
  - Output compare
  - Pulse-width modulation (PWM) generation
  - One-pulse mode output
- STM has many application notes (on all aspects of the STM32)
  - App note AN4776 – General Purpose Timer Cookbook

Timer: Clock



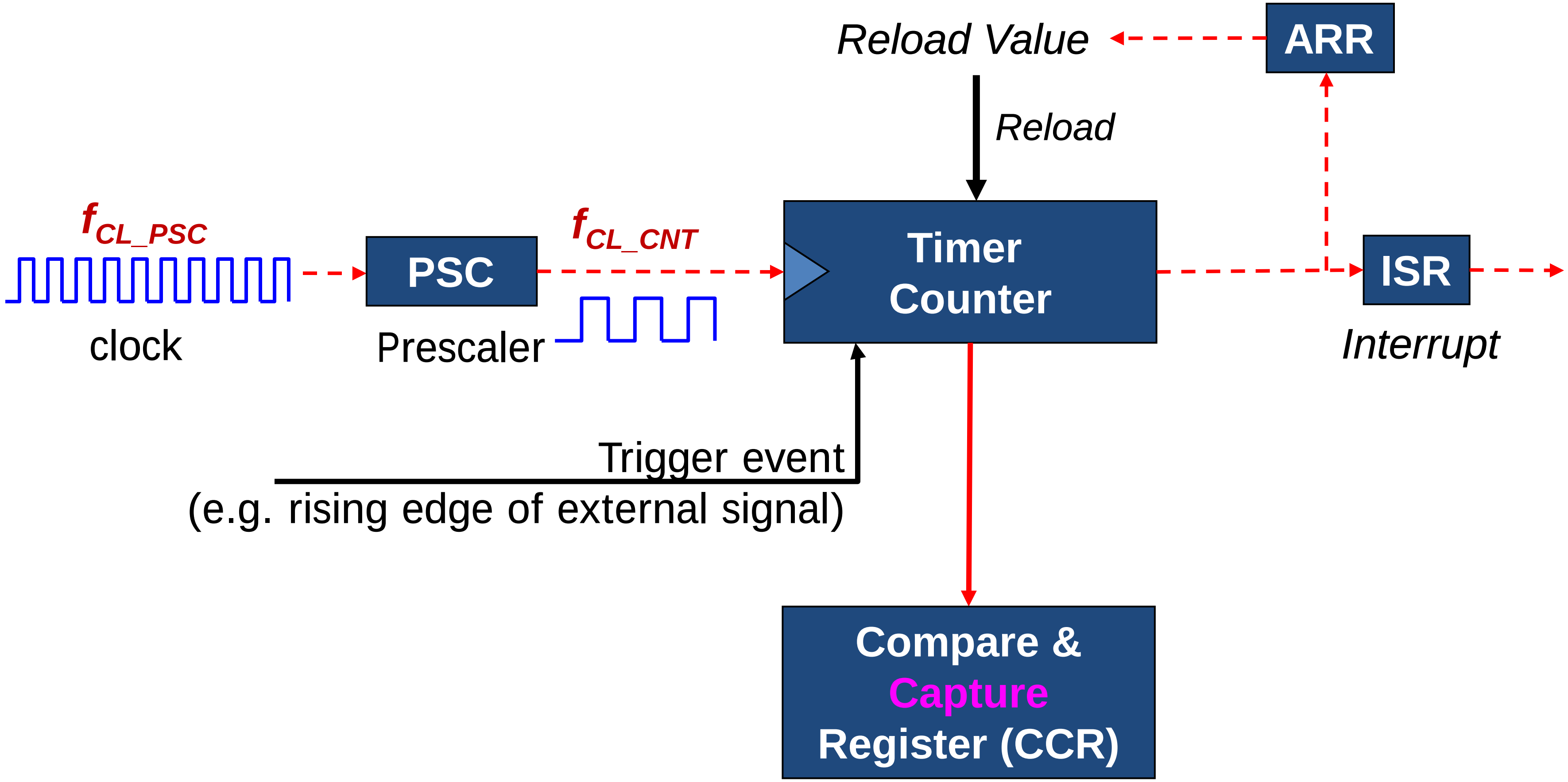
$$f_{CK\_CNT} = \frac{f_{CL\_PSC}}{PSC + 1}$$

Timer: Output

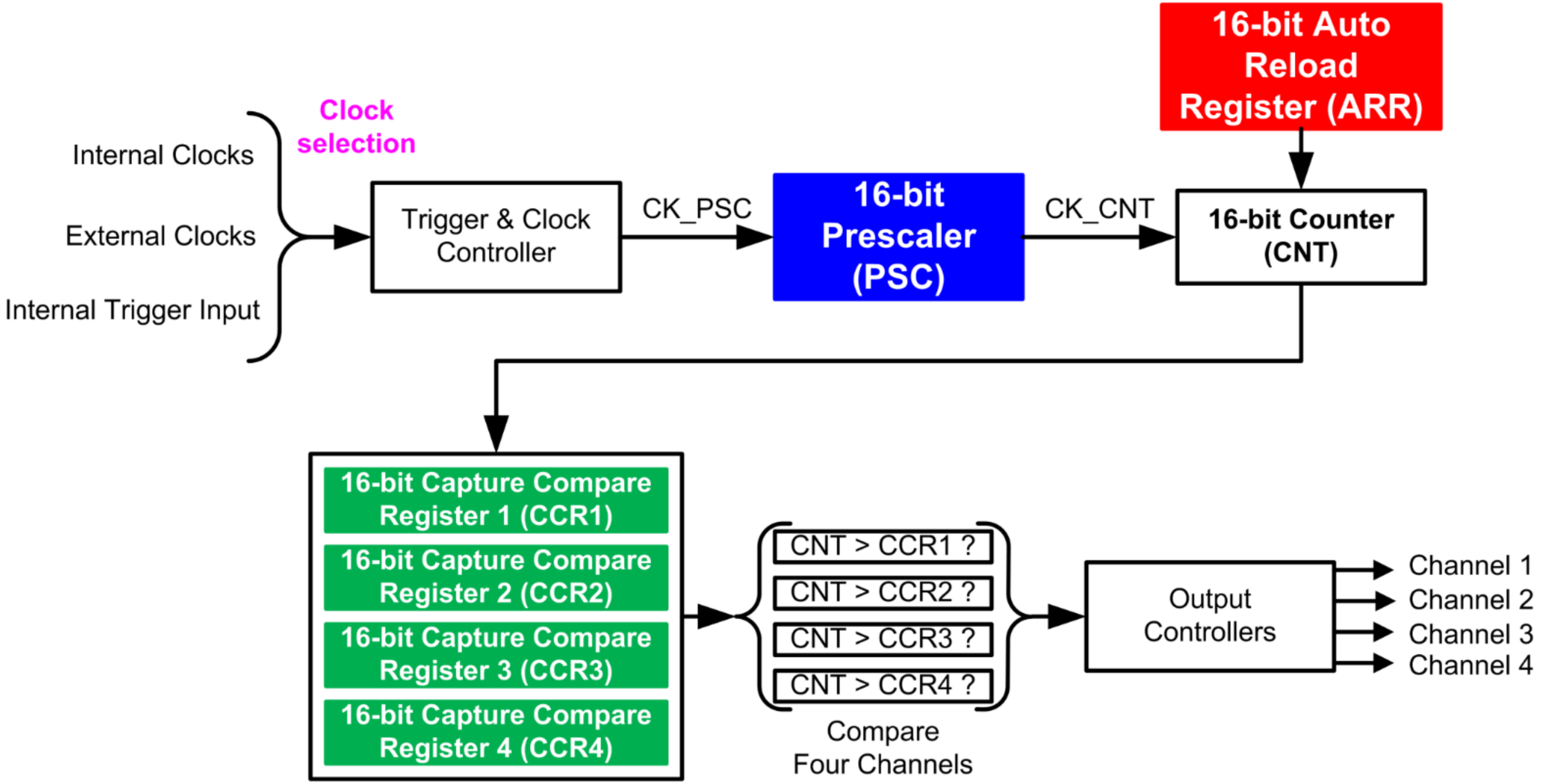


Software Engineering
Rochester Institute of Technology

Timer: Input Capture

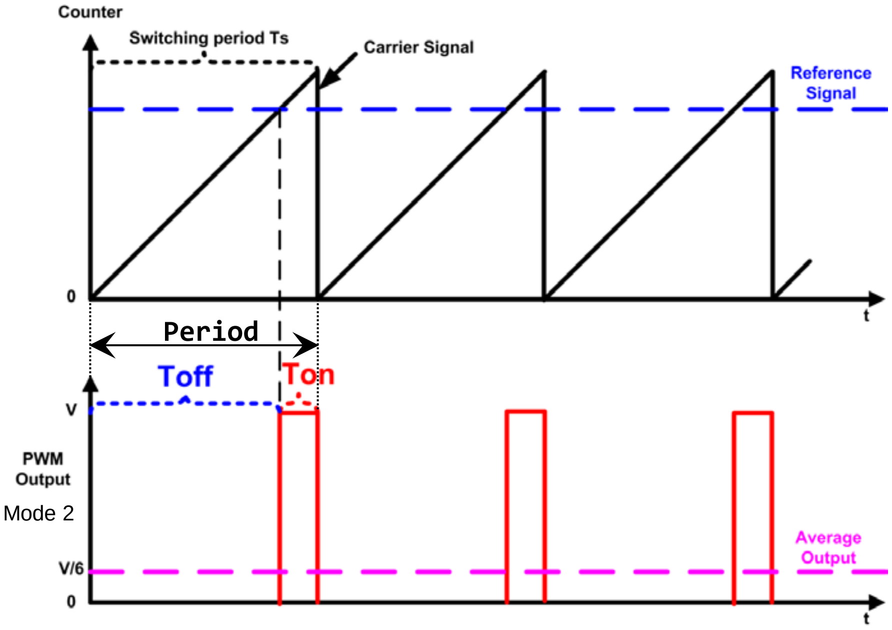


Multi-Channel Outputs



PWM Mode

(Pulse Width Modulation)



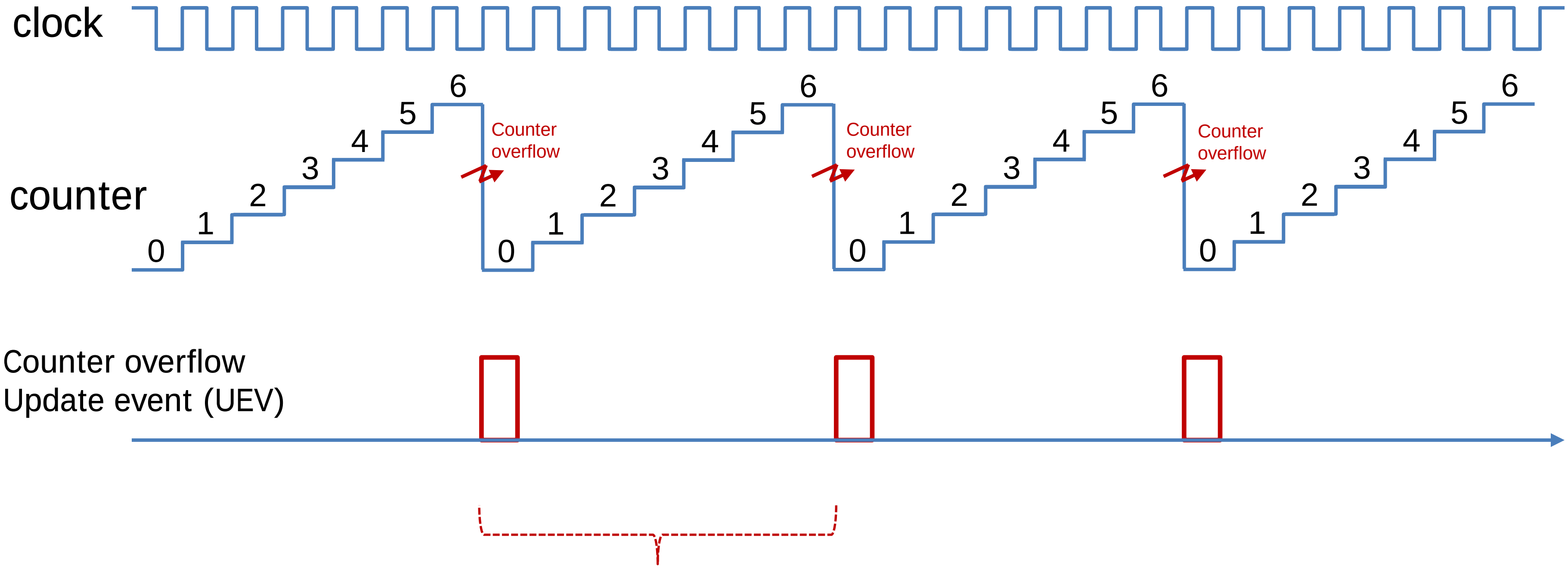
$$\text{Period} = T_{off} + T_{on}$$

$$\text{Duty Cycle} = \frac{T_{on}}{T_{off} + T_{on}}$$

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

Edge-aligned Mode (Up-counting)

ARR = 6, RCR = 0

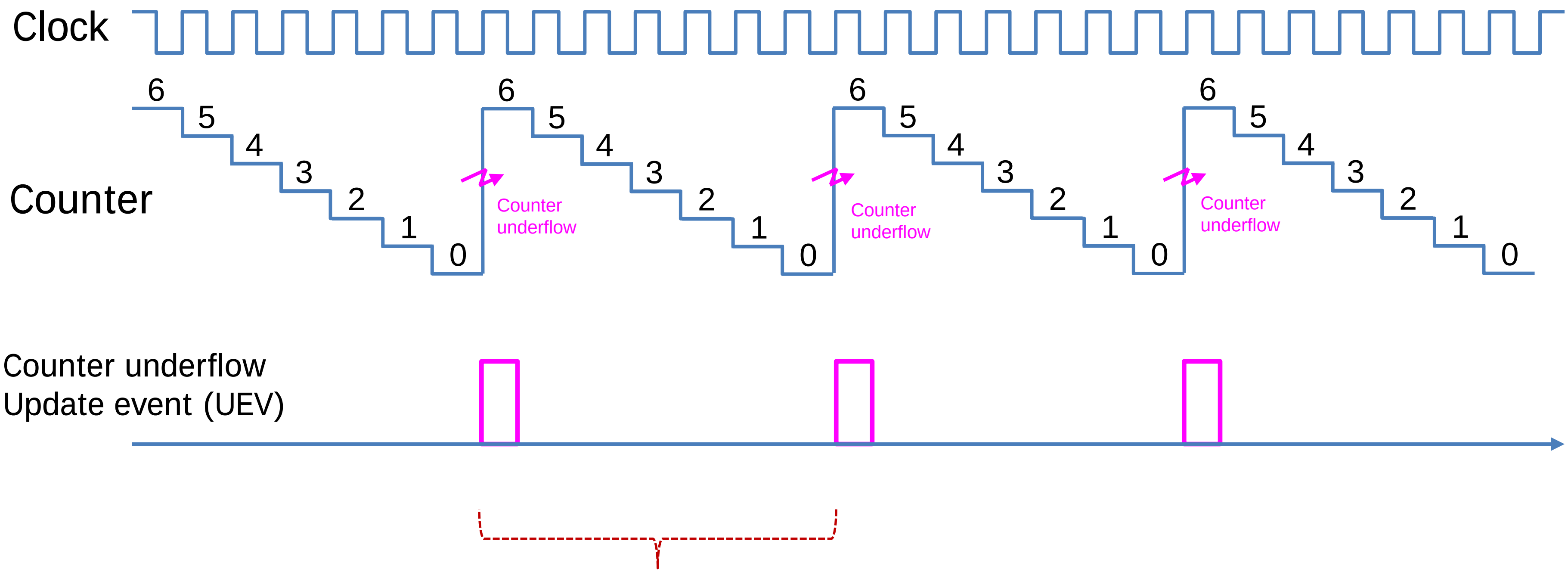


```
Period = (1 + ARR) * Clock Period
       = 7 * Clock Period
```

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

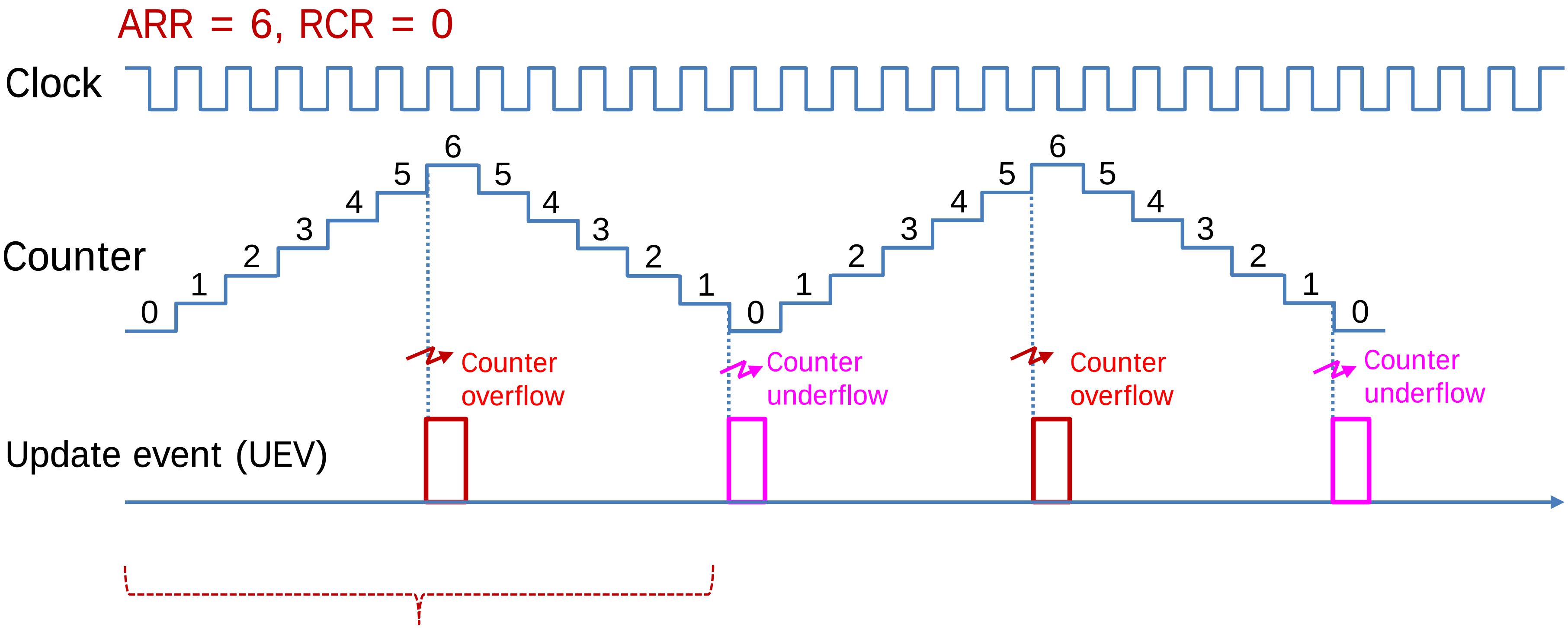
Edge-aligned Mode (down-counting)

ARR = 6, RCR = 0



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

Center-aligned Mode

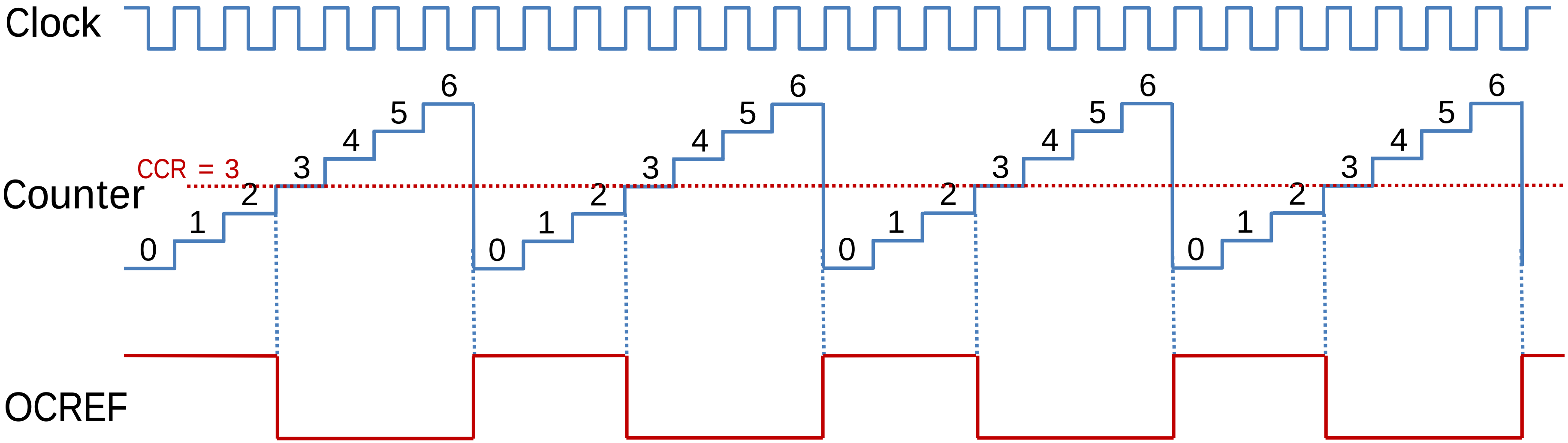


```
Period = (2 * ARR) * Clock Period
       = 12 * Clock Period
```

PWM
Mode 1 (Low-True)

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

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



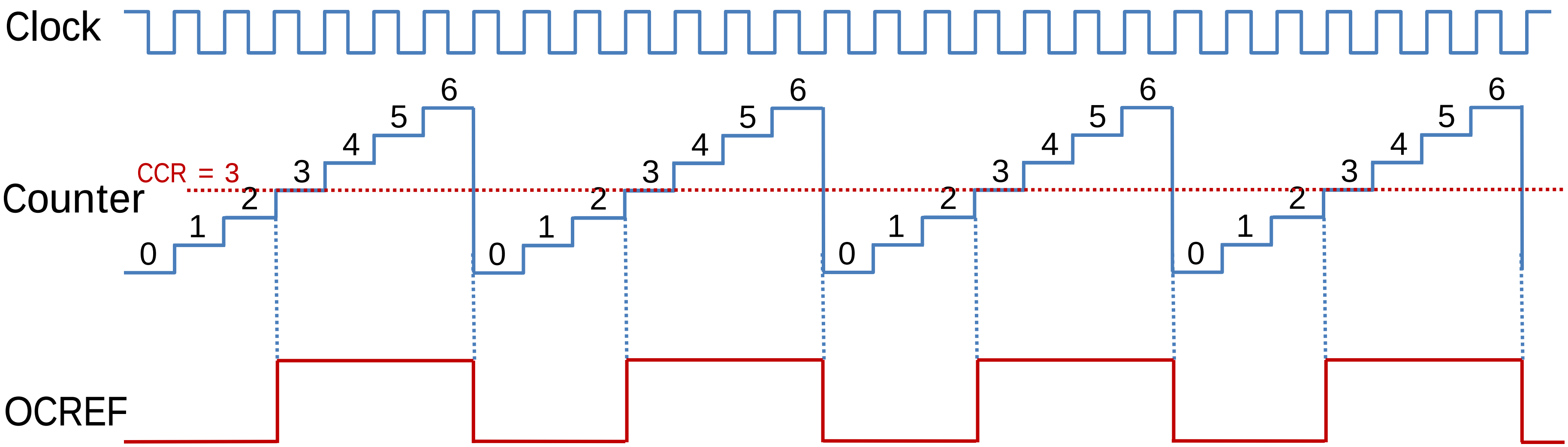
$$\text{Duty Cycle} = \frac{\text{CCR}}{\text{ARR} + 1}$$
$$= \frac{3}{7}$$

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

PWM
Mode 2 (High-True)

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

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



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

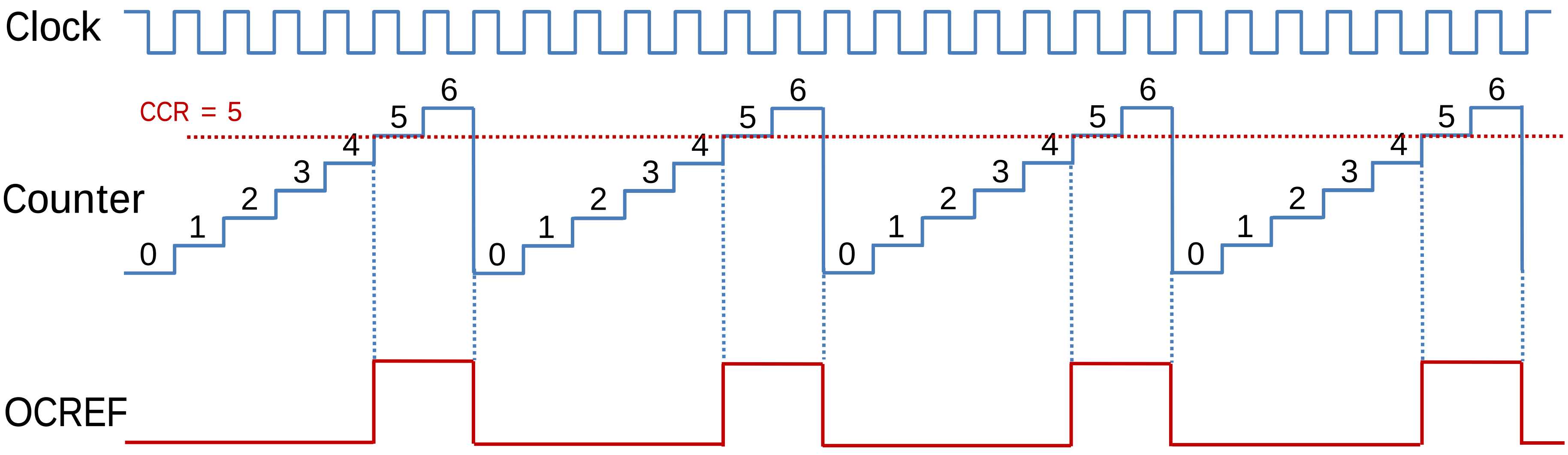
$$= \frac{4}{7}$$

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

PWM
Mode 2 (High-True)

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

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



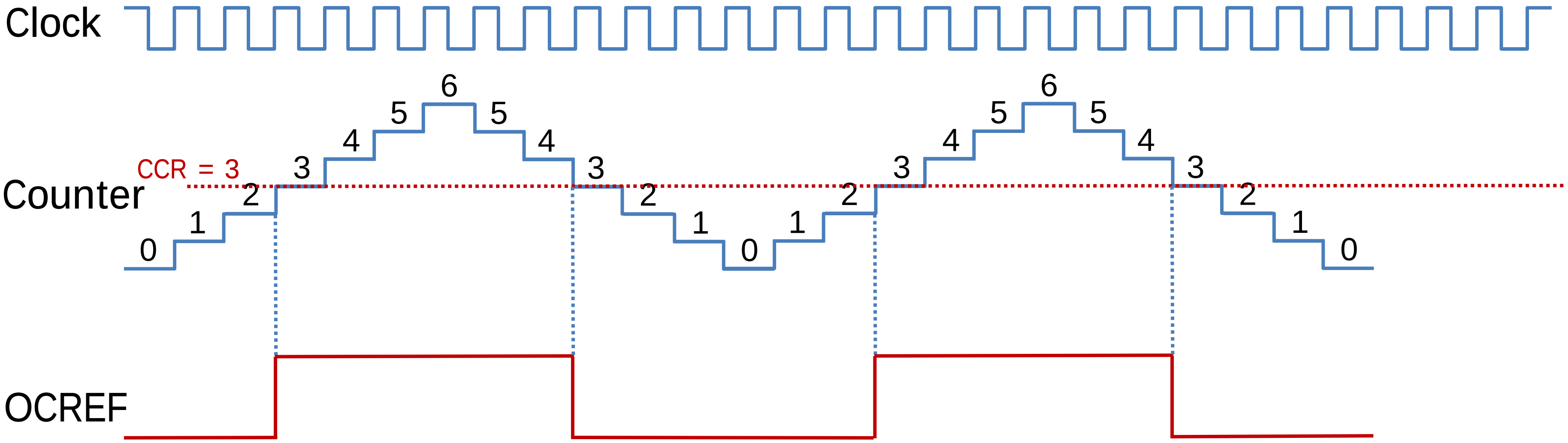
$$\text{Duty Cycle} = 1 - \frac{\text{CCR}}{\text{ARR} + 1} = \frac{2}{7}$$

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

PWM
Mode 2 (High-True)

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

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



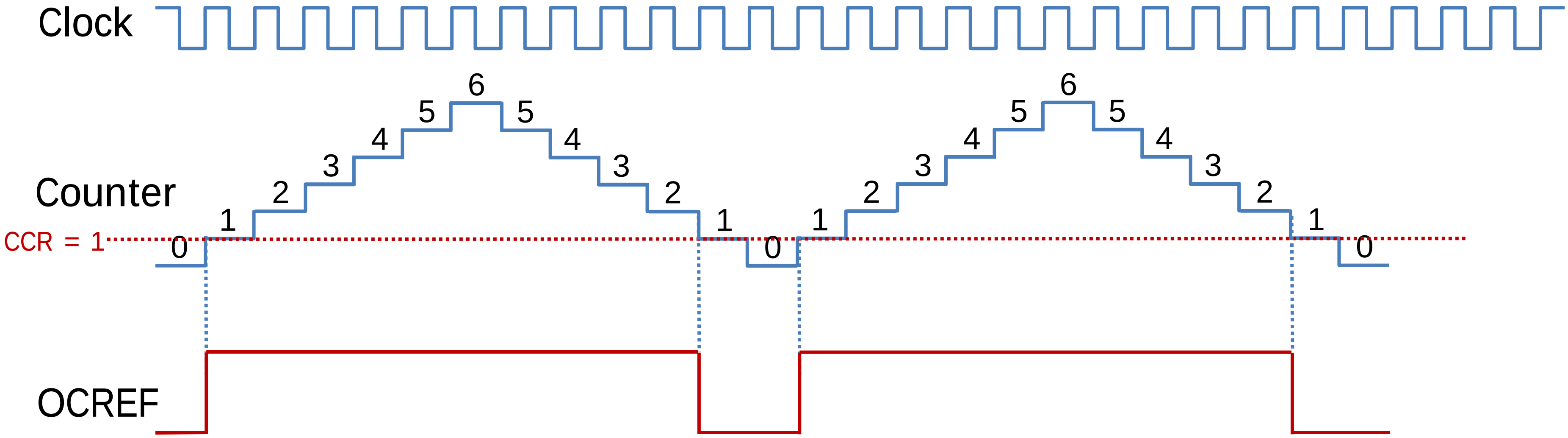
$$\text{Duty Cycle} = 1 - \frac{\text{CCR}}{\text{ARR}} = \frac{1}{2}$$

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

PWM
Mode 2 (High-True)

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

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



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

$$= \frac{5}{6}$$

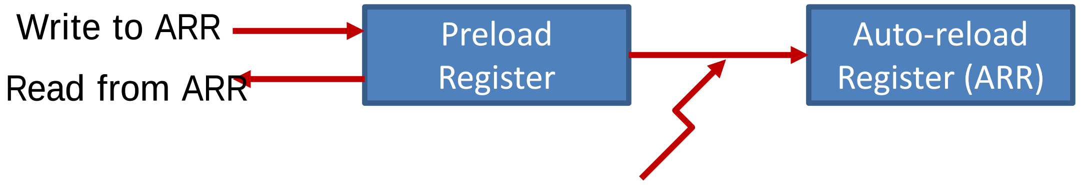
$$\text{Period} = 2 * \text{ARR} * \text{Clock Period}$$

$$= 12 * \text{Clock Period}$$

Auto-Reload Register (ARR)

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

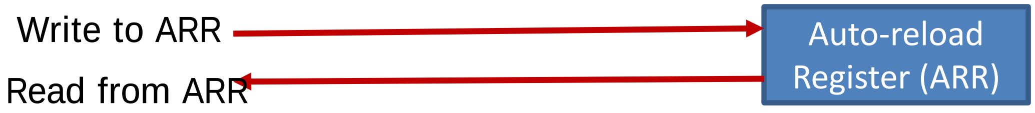
ARPE = 1 (Synchronous Update)



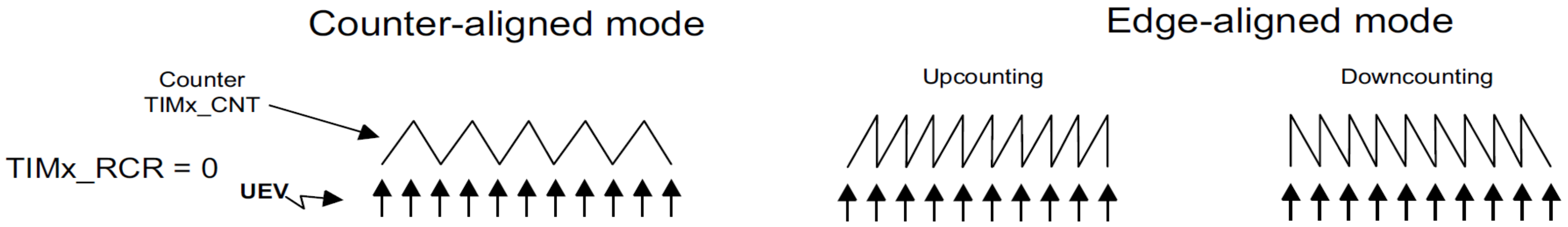
Triggered by Update Event (UEV)

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

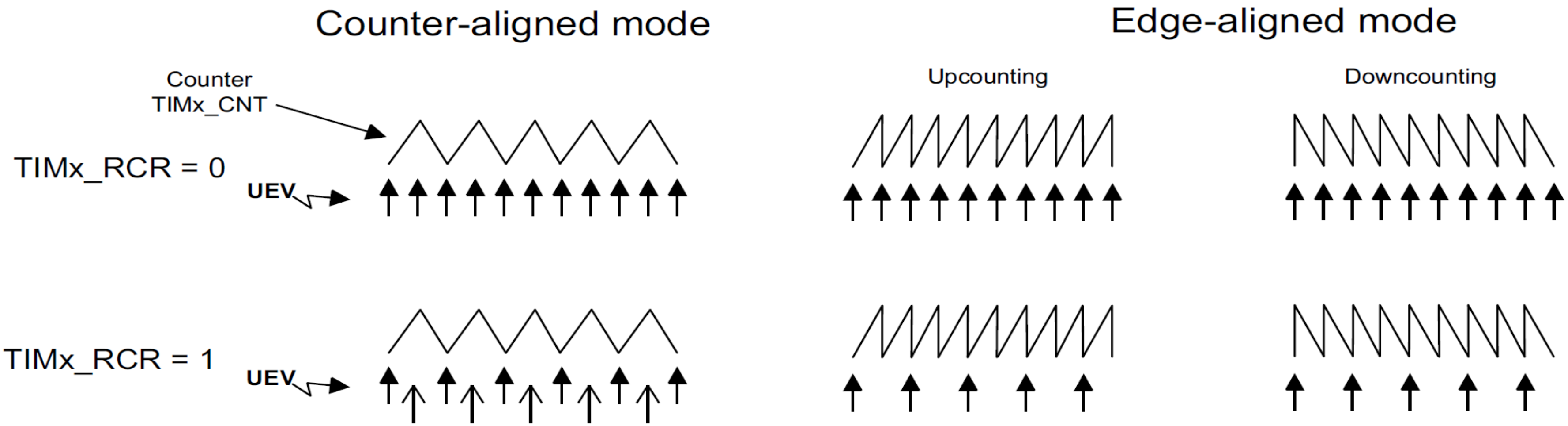
ARPE = 0 (Asynchronous Update)



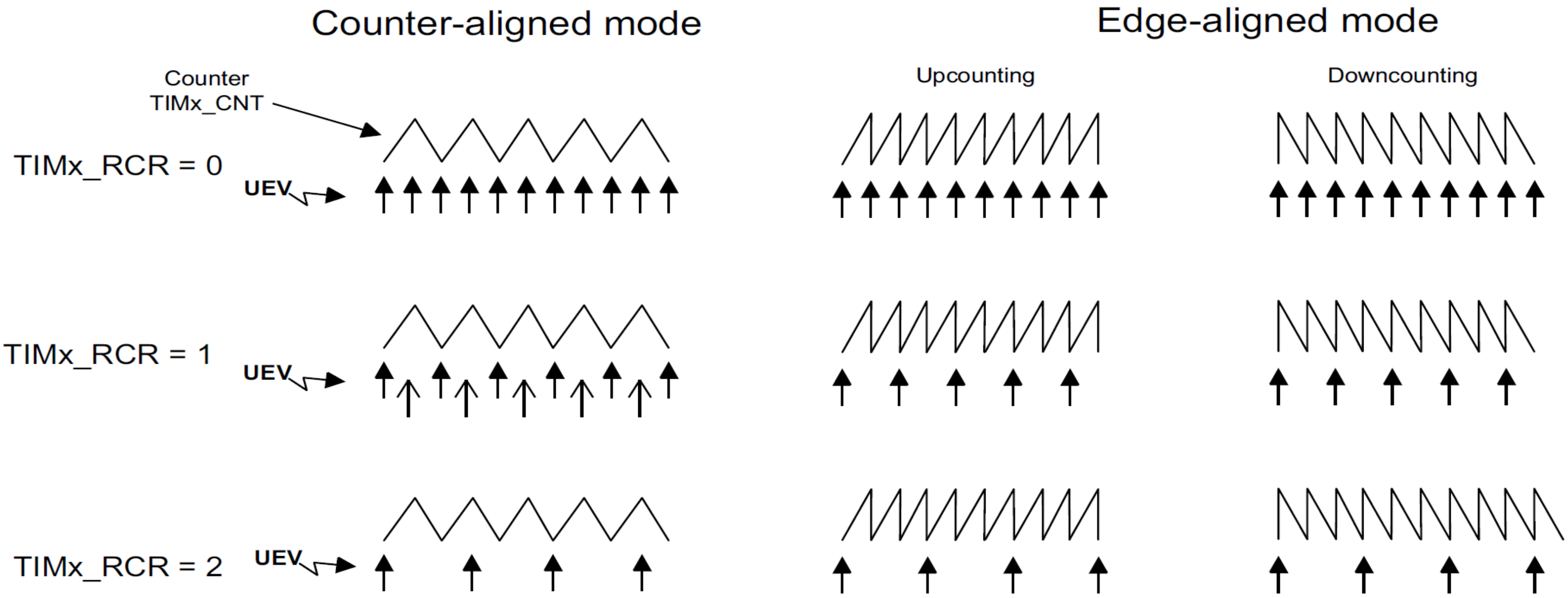
Repetition Counter Register (RCR)



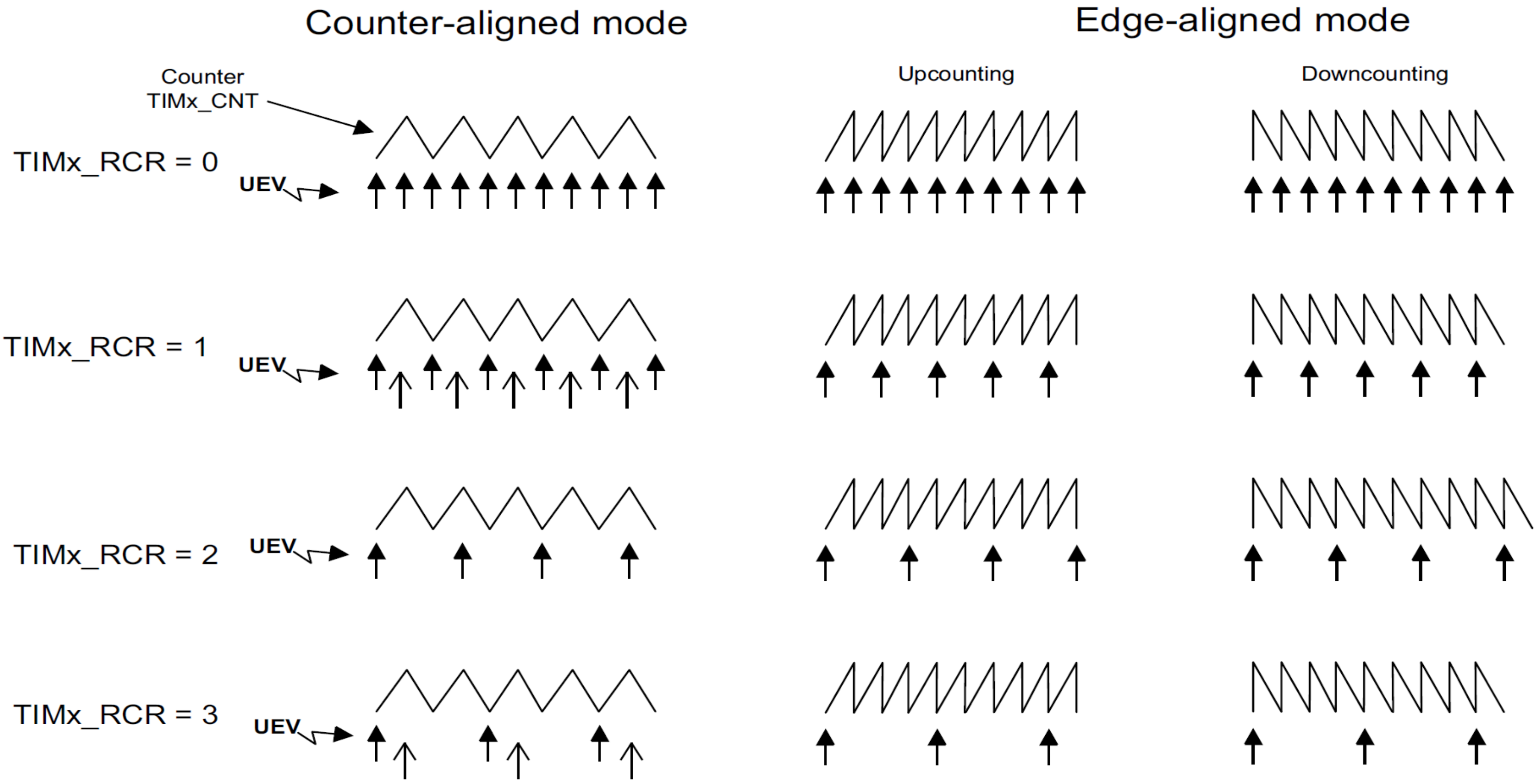
Repetition Counter Register (RCR)



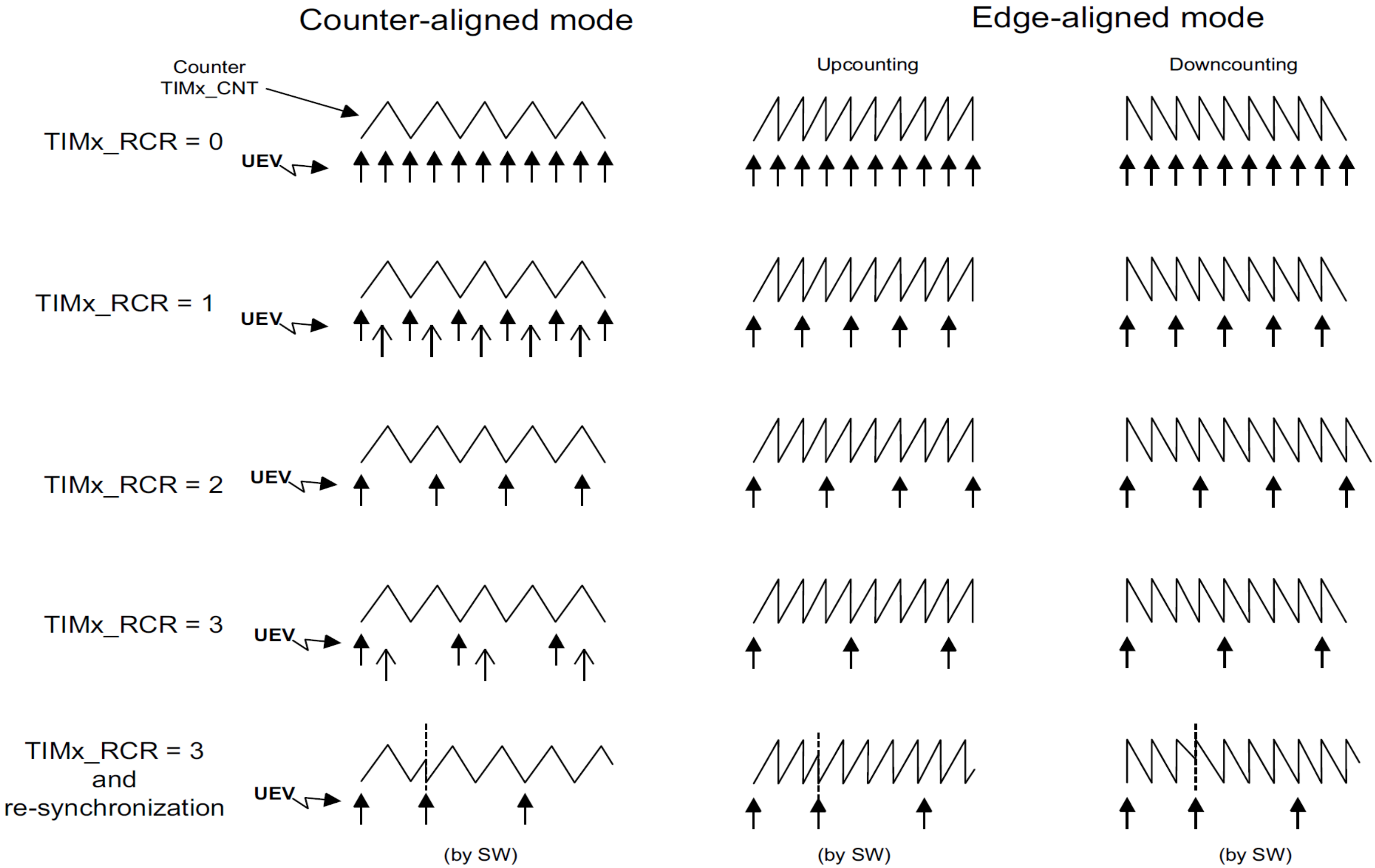
Repetition Counter Register (RCR)



Repetition Counter Register (PCR)



Repetition Counter Register (PCR)



PWM Output Polarity

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

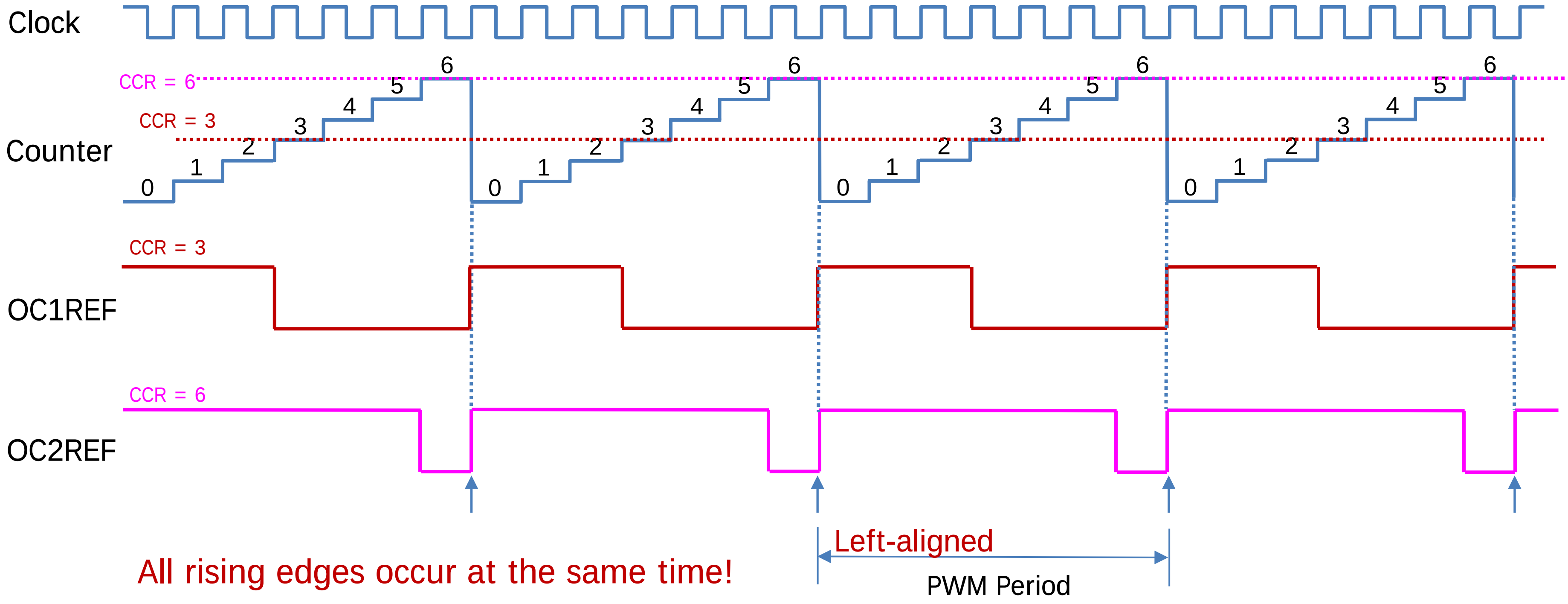
Output Polarity:

- 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

Up-Counting: Left Edge-aligned

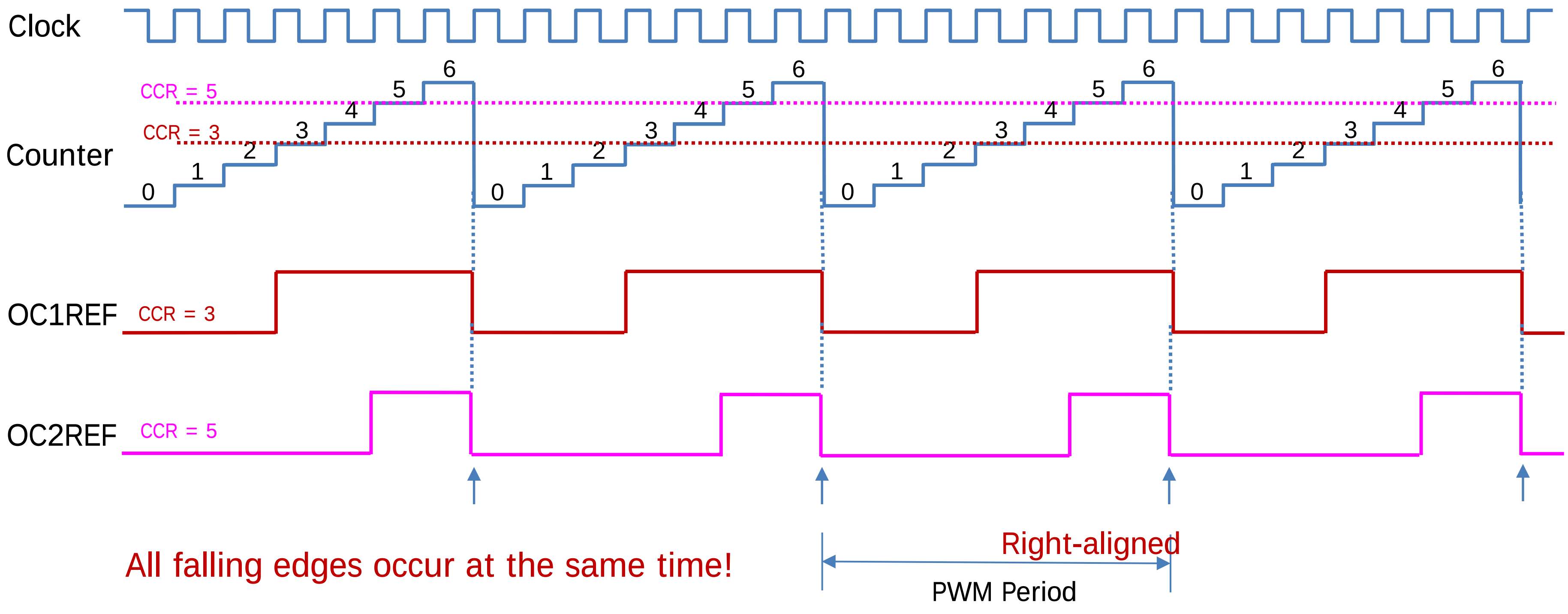
Upcounting mode, ARR = 6, CCR = 3, RCR = 0



All rising edges occur at the same time!

PWM Mode 2: Right Edge-aligned

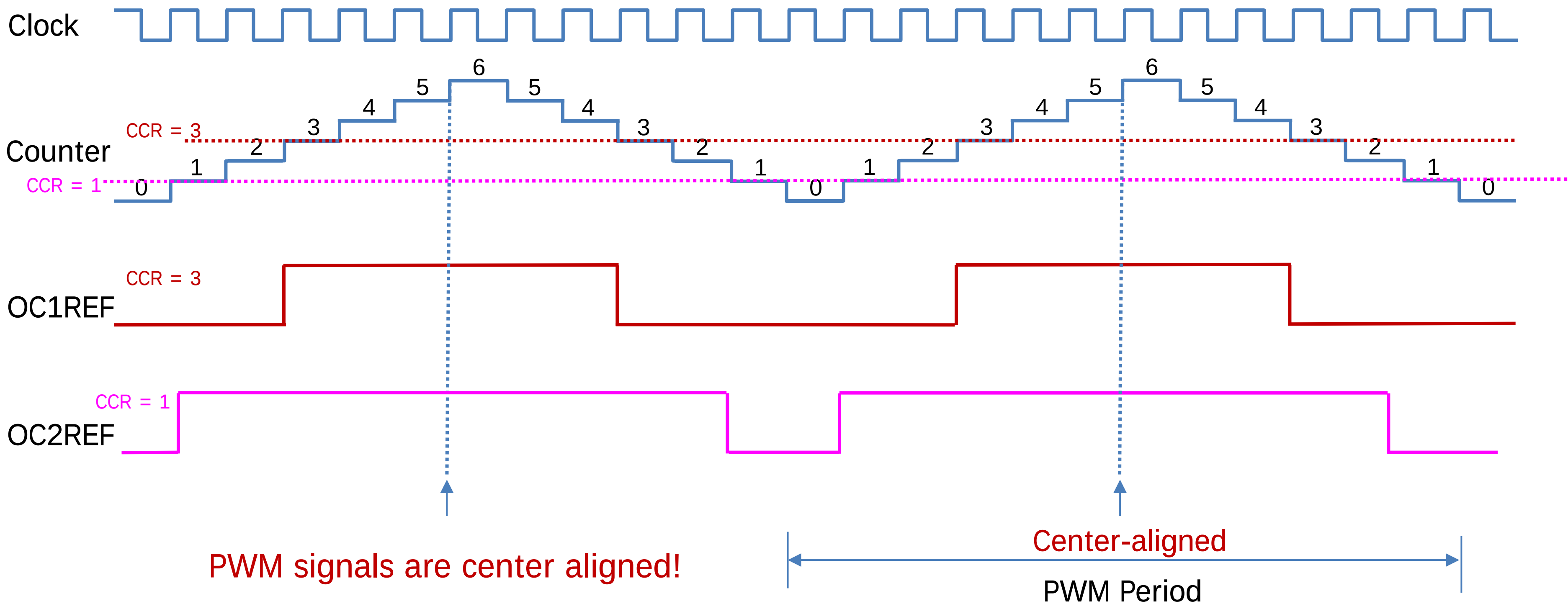
Upcounting mode, ARR = 6, CCR = 3, RCR = 0



All falling edges occur at the same time!

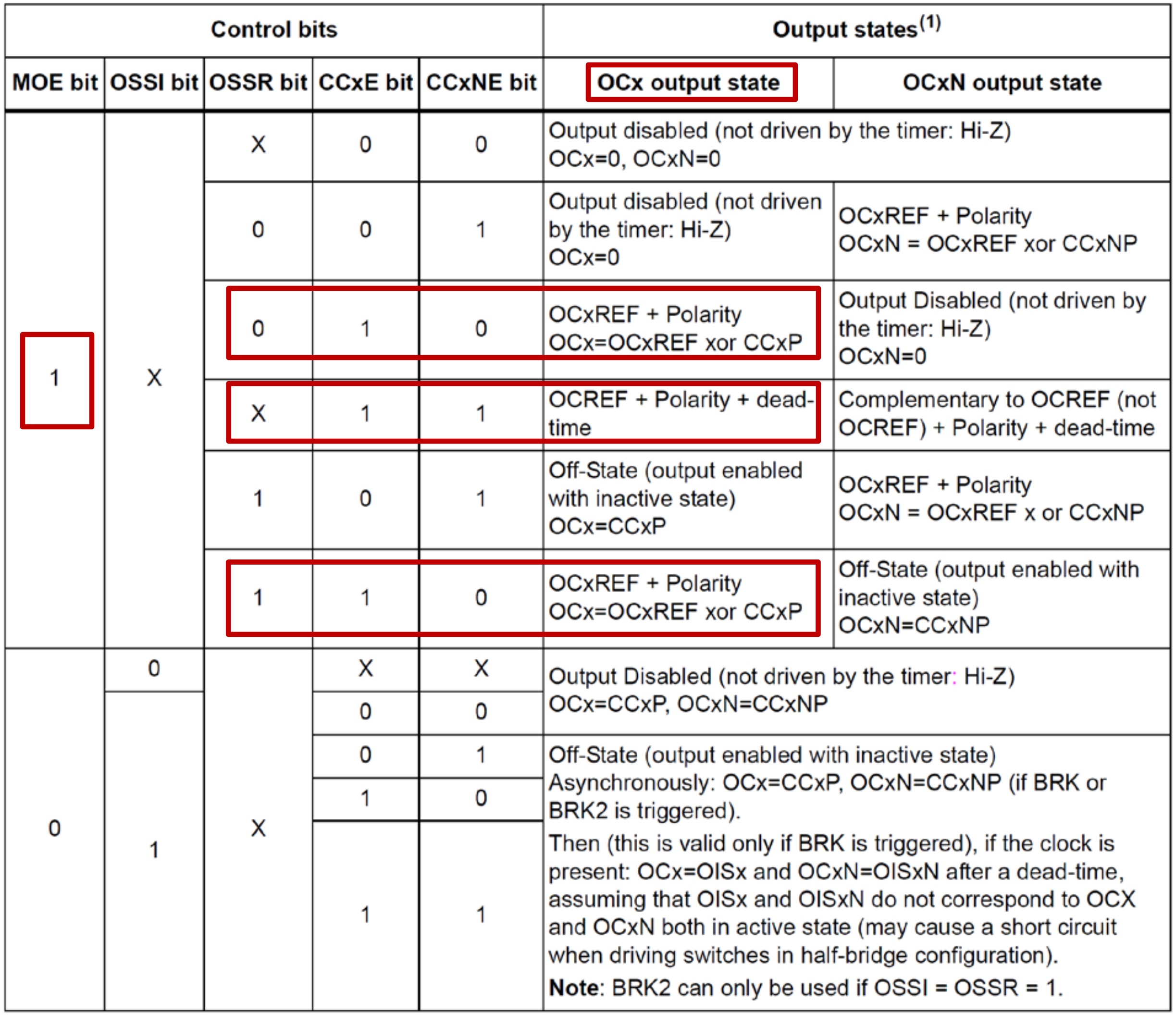
PWM Mode 2: Center Aligned

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



The devil is in the detail

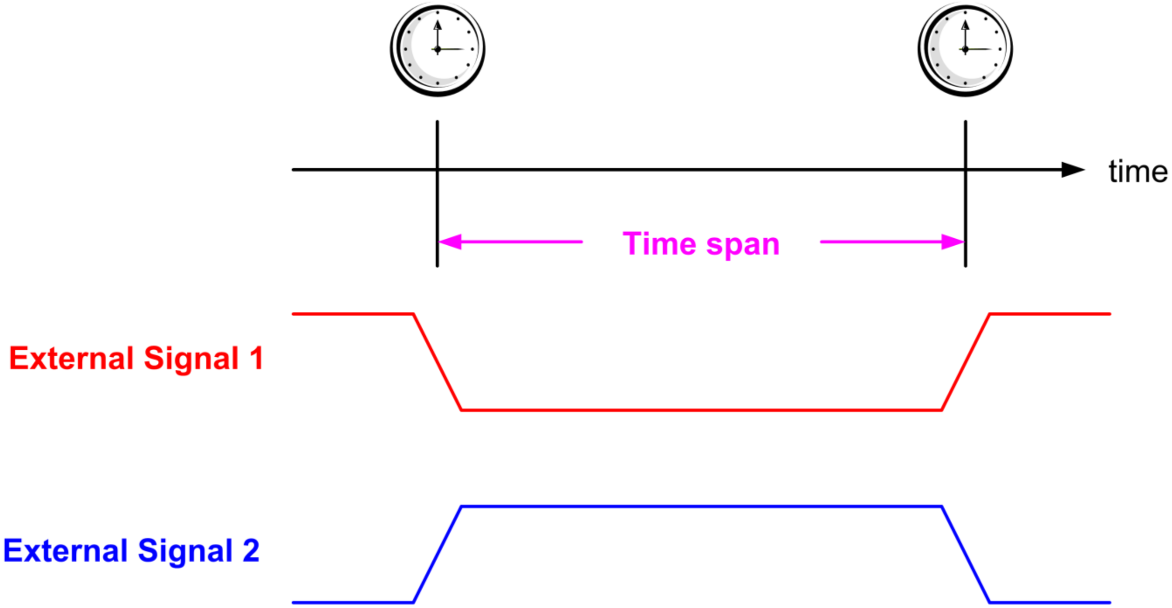
- Timer output control
- Enable Timer Output
  - MOE: Main output enable
  - OSSI: Off-state selection for Idle mode
  - OSSR: Off-state selection for Run mode
  - CCxE: Enable of capture/compare output for channel x
  - CCxNE: Enable of capture/compare complementary output for channel x



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		

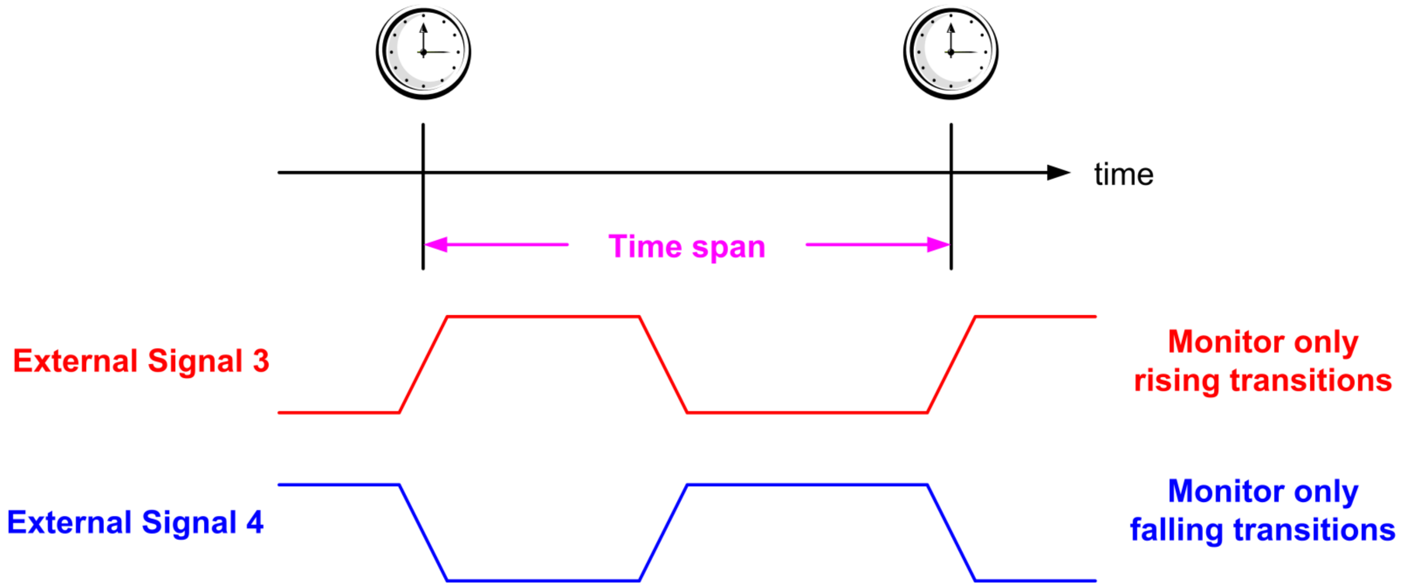
Input Capture

- Monitor both rising and falling edge

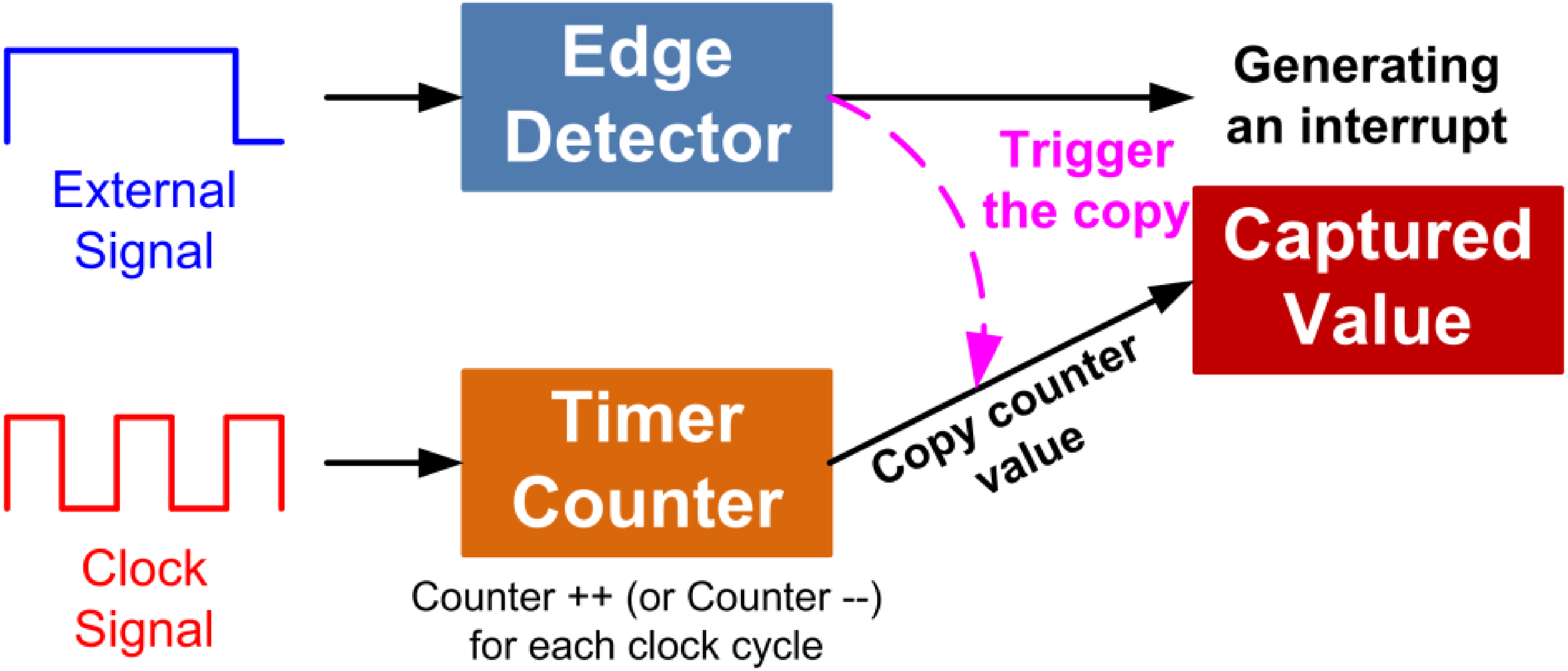


Input Capture

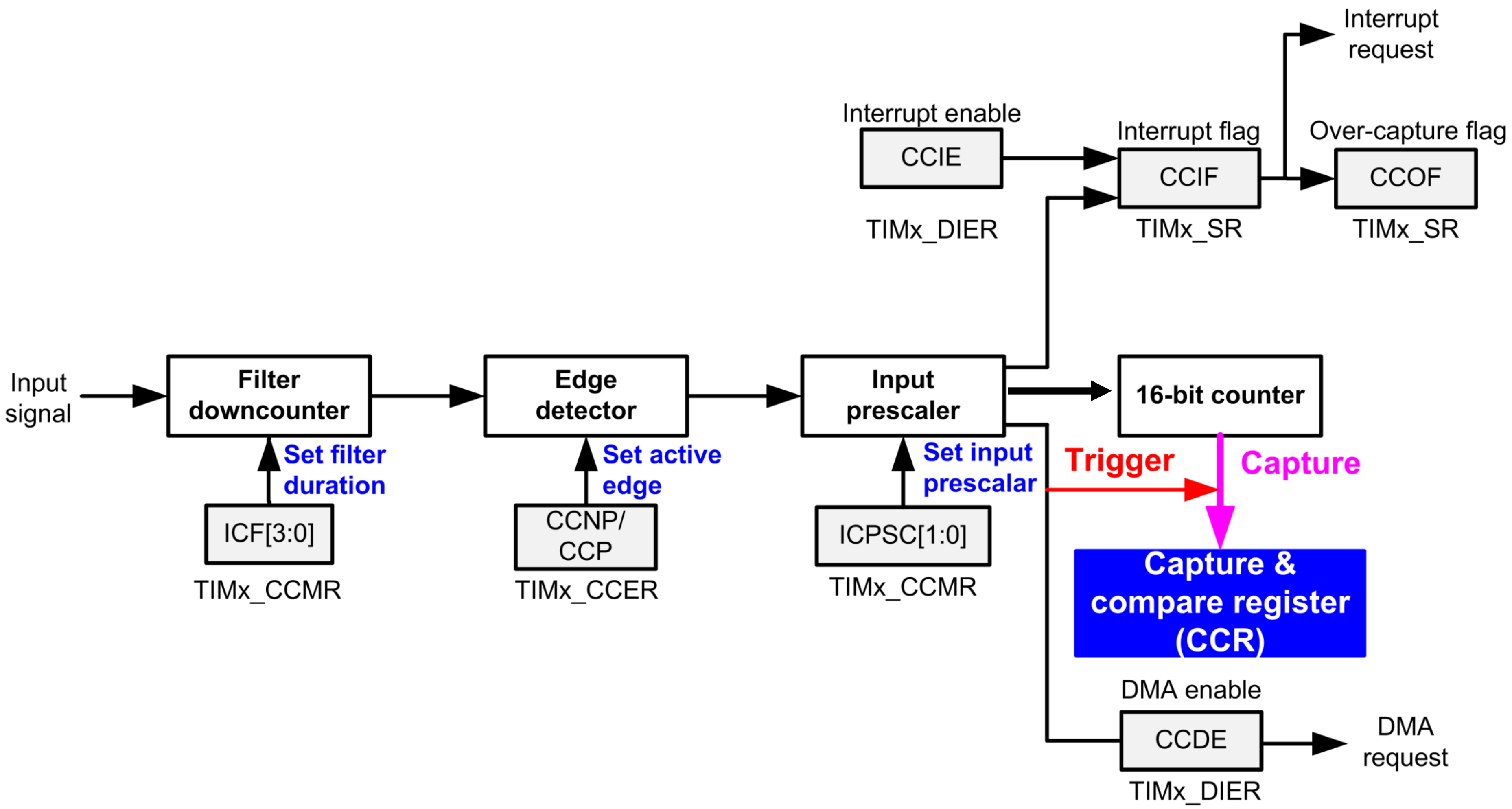
- Monitor only rising edges or only falling edge


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

Input Capture

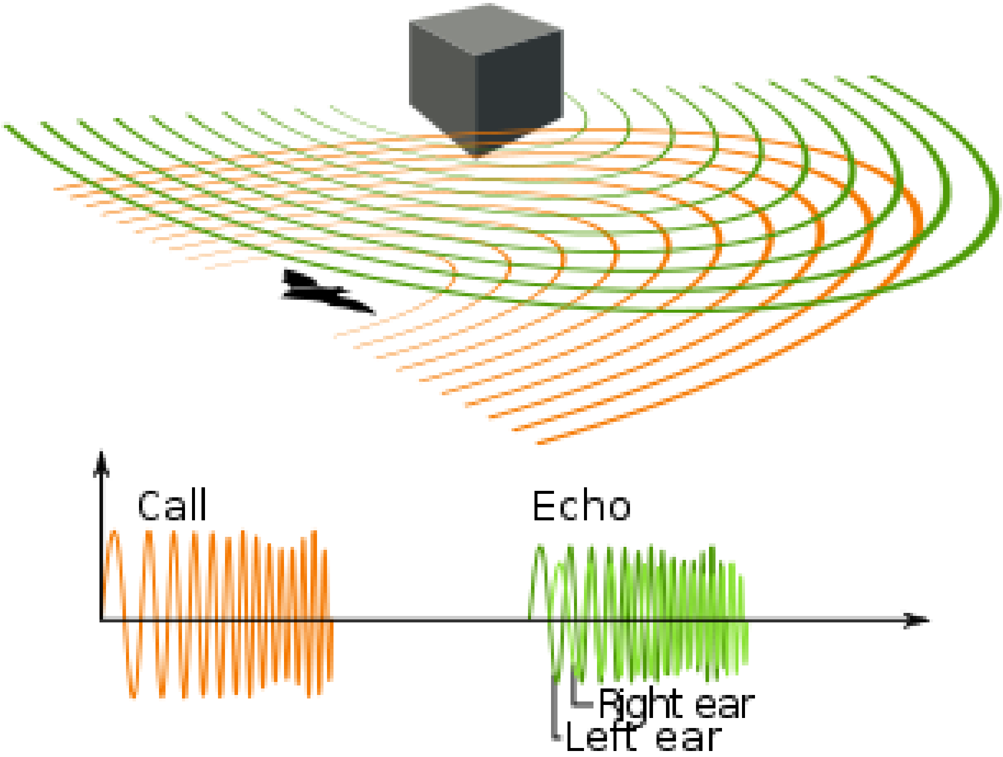


Input Capture Diagram

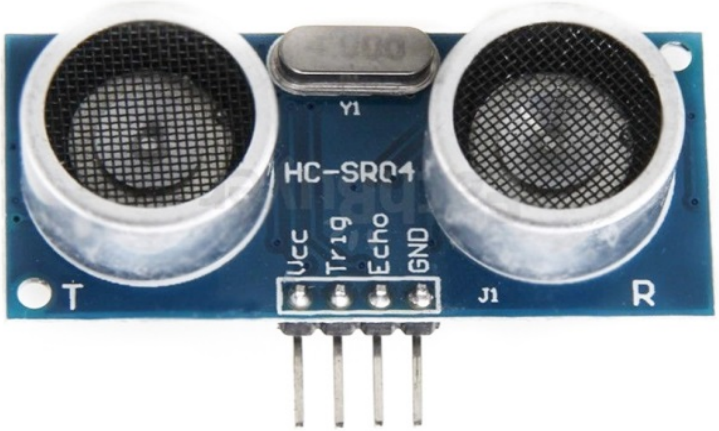


WHY IS THE MEASUREMENT OF TIME INTERESTING???

Bats use echolocation to map their surroundings?



Ultrasonic Distance Sensor

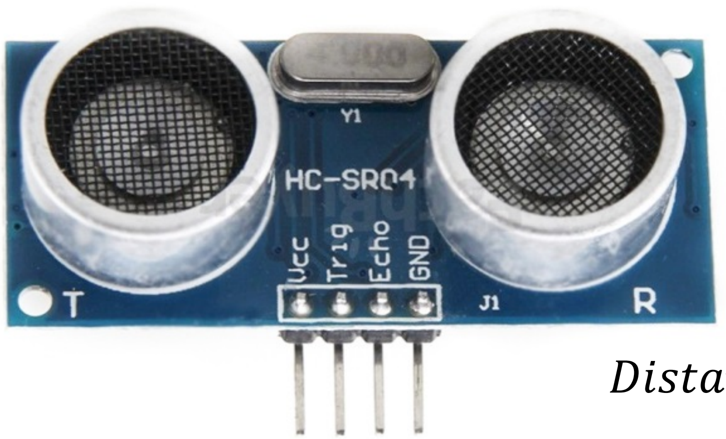


$$Distance = \frac{Round\ Trip\ Time \times Speed\ of\ Sound}{2}$$

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

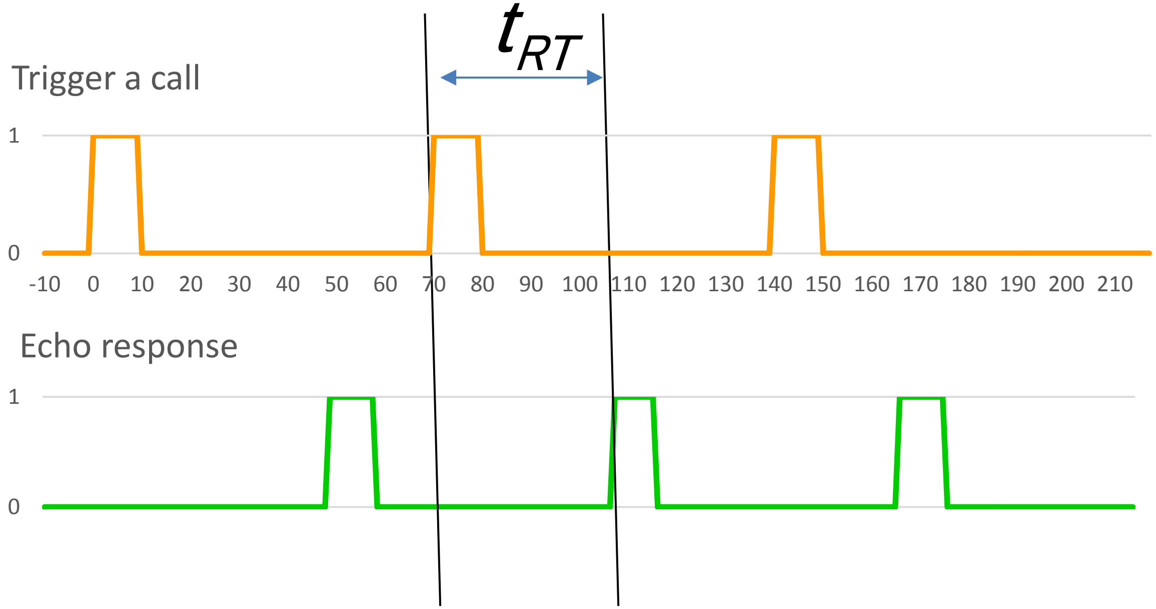
$$= \frac{Round\ Trip\ Time(\mu s)}{58}$$

Ultrasonic Distance Sensor



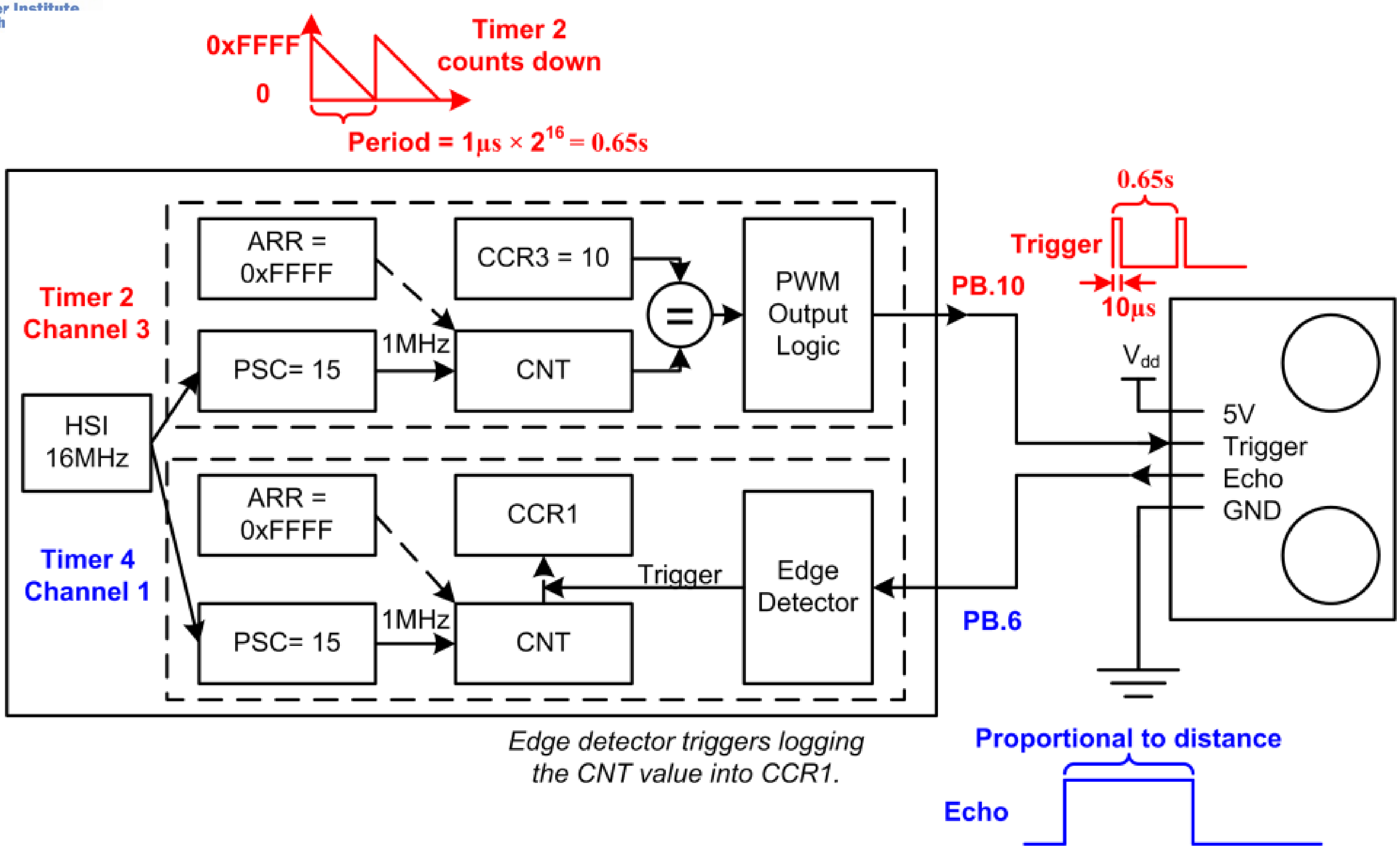
The delay from triggered chirp to echo response is the round-trip time t_{RT}.

$$Distance\ (cm) = \frac{Round\ trip\ time\ (\mu s)}{58}$$



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

Ultrasonic Distance Sensor



Edge detector triggers logging the CNT value into CCR1.