

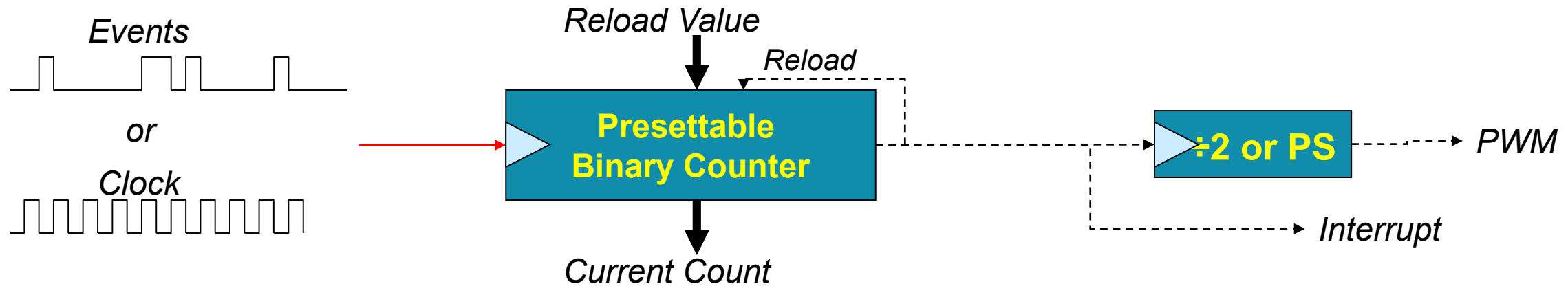
# Timer Peripherals

Ravi Suppiah  
Lecturer, NUS SoC

# KL25 Timer Peripherals

- **PIT - Periodic Interrupt Timer**
  - Can periodically generate interrupts or trigger DMA (direct memory access) transfers
- **TPM - Timer/PWM Module**
  - Connected to I/O pins, has input capture and output compare support
  - Can generate PWM signals
  - Can generate interrupts and DMA requests
- **LPTMR - Low-Power Timer**
  - Can operate as timer or counter in all power modes (including low-leakage modes)
  - Can wake up system with interrupt
  - Can trigger hardware
- **Real-Time Clock**
  - Powered by external 32.768 kHz crystal
  - Tracks elapsed time (seconds) in 32-bit register
  - Can set alarm
  - Can generate 1 Hz output signal and/or interrupt
  - Can wake up system with interrupt
- **SYSTICK**
  - Part of CPU core's peripherals
  - Can generate periodic interrupt

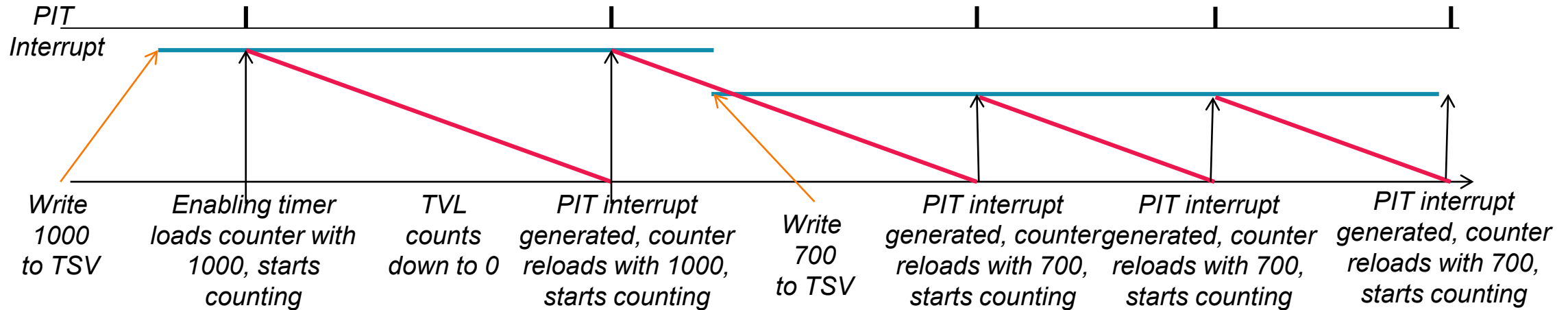
# Timer/Counter Peripheral Introduction



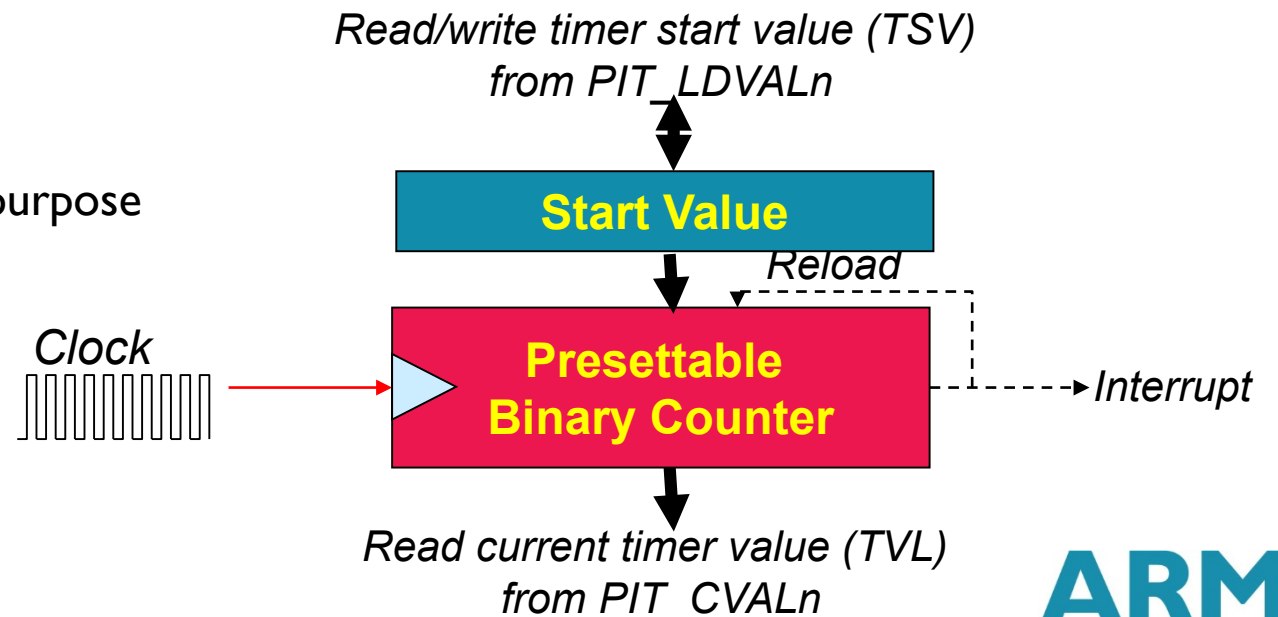
- Common peripheral for microcontrollers
- Based on presetable binary counter, enhanced with configurability
  - Count value can be read and written by MCU
  - Count **direction** can often be set to up or down
  - Counter's **clock source** can be selected
    - **Counter mode:** count **pulses** which indicate **events** (e.g. odometer pulses)
    - **Timer mode:** clock source is periodic, so counter value is proportional to **elapsed time** (e.g. stopwatch)
  - Counter's **overflow/underflow action** can be selected
    - Generate interrupt
    - Reload counter with special value and continue counting
    - Toggle hardware output signal
    - Stop!

# PERIODIC INTERRUPT TIMER

# Periodic Interrupt Timer



- Generates periodic interrupts using a 32-bit counter
- Load start value (32-bit) from LDVAL
- Counter decrements with each clock pulse
  - Fixed clock source for PIT - Bus Clock from Multipurpose Clock Generator - e.g. 24 MHz
- When timer value (CVAL) reaches zero
  - Generates interrupt
  - Reloads timer with start value



# TIMER/PWM MODULE (TPM)

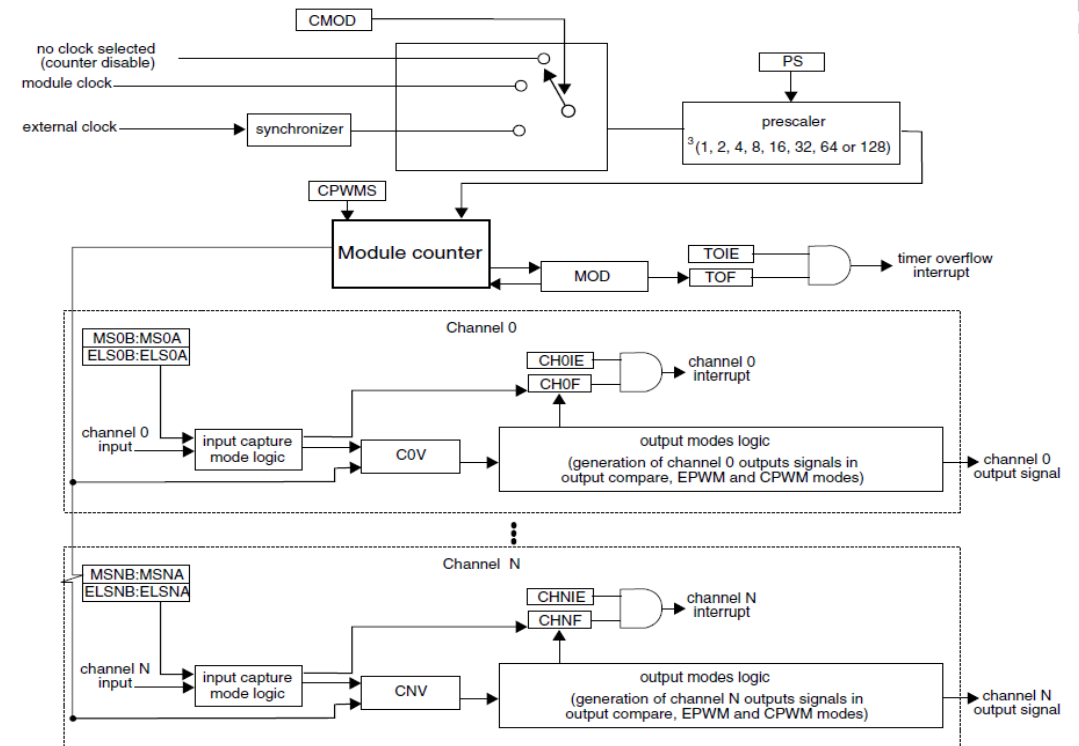
# TPM - Timer/PWM Module

## ■ Core: **Module counter**

- Two clock options - external or internal
  - no clock selected (counter disable)
  - module clock
  - external clock
- Prescaler to divide clock by 1 to 128
- 16-bit counter
  - Can count up or up/down
  - Can reload with set load value or wrap around (to FFFF or 0000)

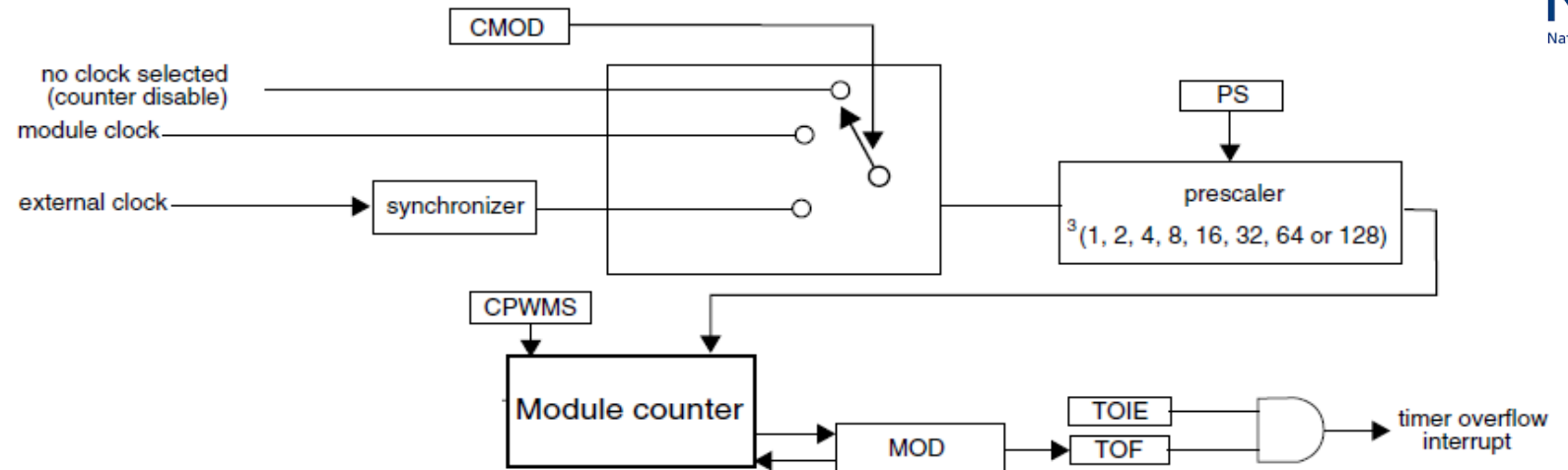
## ■ Six channels

- 3 modes
  - Capture Mode: capture timer's value when input signal changes
  - Output Compare: Change output signal when timer reaches certain value
  - PWM: Generate pulse-width-modulated signal. Width of pulse is proportional to specified value.
- Each channel can generate interrupt, DMA request, hardware trigger on overflow
- One I/O pin per channel TPM\_CHn



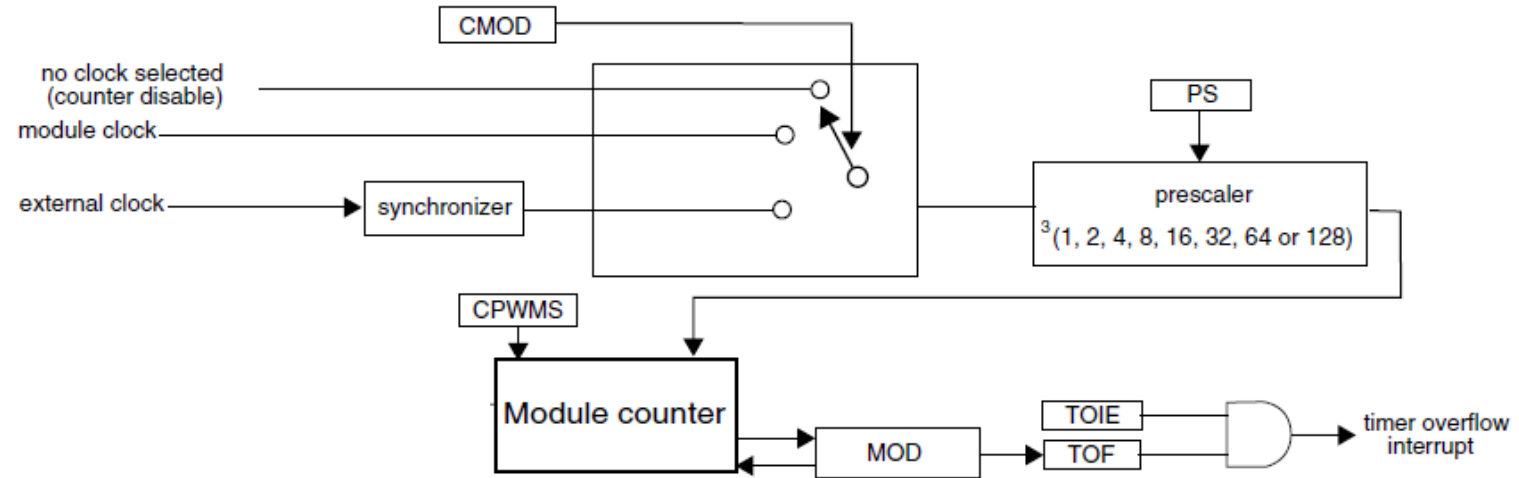
# Timer Configuration

- Clock source
  - CMOD: selects internal or external clock
- Prescaler
  - PS: divide selected clock by 1, 2, 4, 8, 16, 32, 64, 128
- Count Mode and Modulo
  - CPWMS: count up (0) or up and down (1)
  - MOD: 16-bit value up to which the counter counts
    - Up counting: 0, 1, 2, ... MOD, 0/Overflow, 1, 2, ... MOD
    - Up/down counting: 0, 1, 2, ... MOD, MOD-1/Interrupt, MOD-2, ... 2, 1, 0, 1, 2, ...
- Timer overflows when counter goes 1 beyond MOD value
- DMA: Enable DMA transfer on overflow
- TOF: Flag indicating timer has overflowed





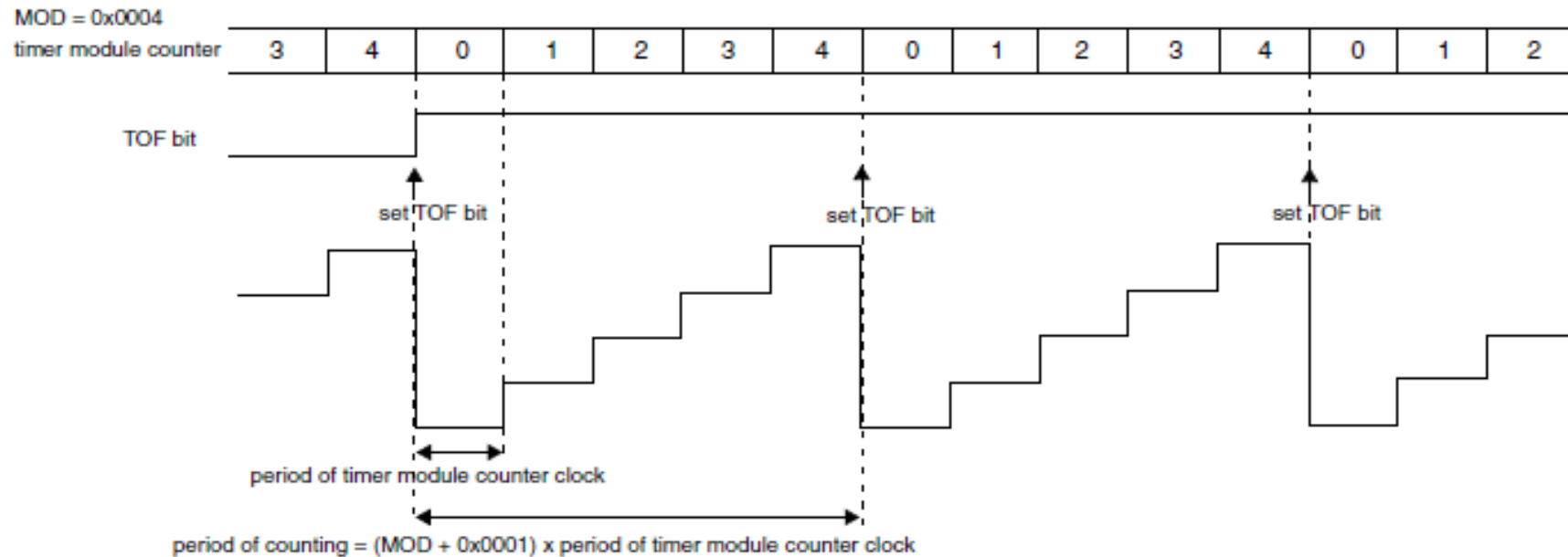
# Basic Counter Mode



- Count external events applied on input pin
  - Set CMOD = 10 to select external input
  - Set PS = 000 (unless division needed)
- Timer overflow flag TOF set to 1 when counter goes by 1 beyond MOD value
- Can generate interrupt if TOIE is set

| 2-0 PS | Prescaler Factor |
|--------|------------------|
| 000    | 1                |
| 001    | 2                |
| 010    | 4                |
| 011    | 8                |
| 100    | 16               |
| 101    | 32               |
| 110    | 64               |
| 111    | 128              |

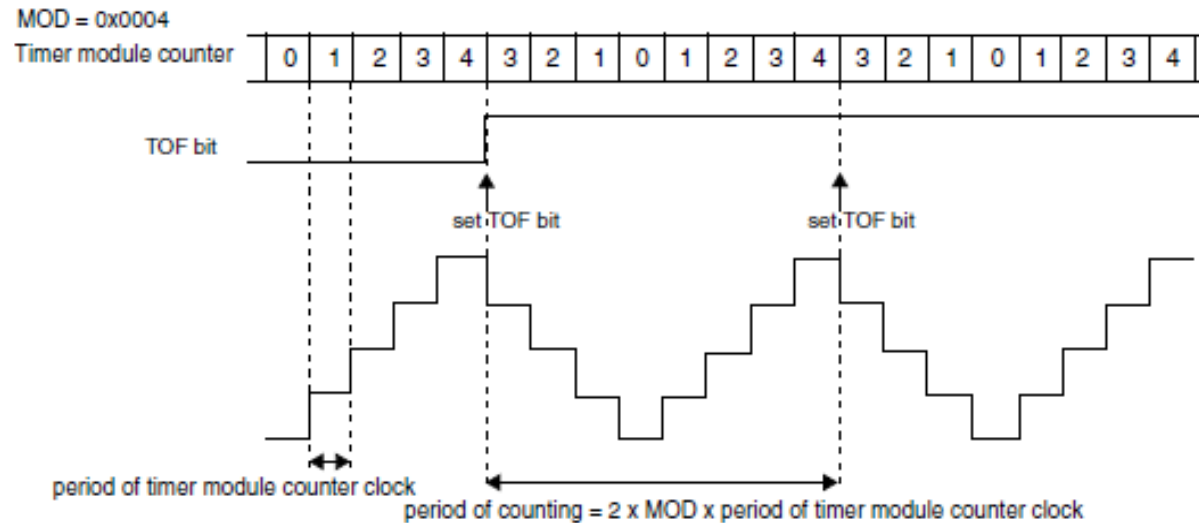
# Count Mode and Modulo - Counting Up



**Figure 31-79. Example of TPM Up Counting**

- Counter increments with each clock tick
- When counter reaches MOD, at the end of the tick,
  - set TOF bit (timer overflow)
  - reset counter value to 0
- Frequency of overflows is timer clock frequency / (1 + MOD)

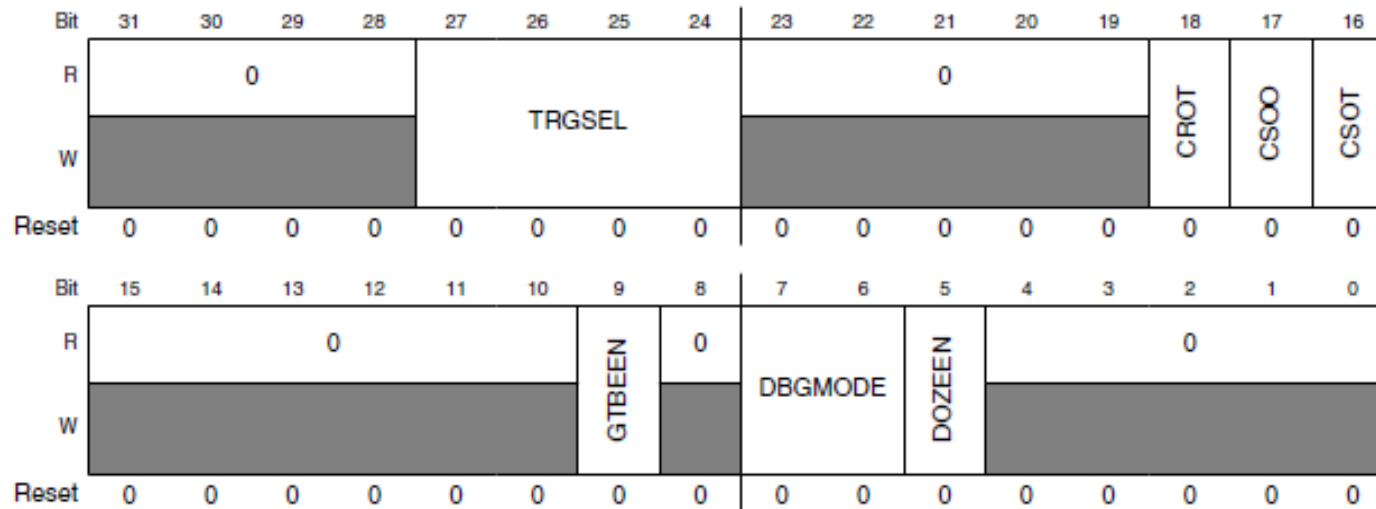
# Count Mode and Modulo - Counting Up and Down



**Figure 31-80. Example of Up-Down Counting**

- Two modes
  - Up-counting
  - Counter increments with each clock tick
  - When counter reaches MOD, set TOF bit (timer overflow) at the end of the tick, set to down-count mode
  - Down-counting
  - Counter decrements with each clock tick
  - When counter reaches 0, set to up-count mode
- Frequency of overflows is timer clock frequency / (2 \* MOD)

# TPM Configuration (TPMx\_CONF)



- TRGSEL - input trigger select
- CROT - counter reload on trigger
- CSOO - counter stop on overflow
- CSOT - counter start on trigger
- GTBEEN - external global time base enable (rather than LPTPM counter)
- DBGMODE - let LPTPM counter increment during debug mode
- DOZEEN - pause LPTPM when in doze mode

# TPM Status (TPMx\_STATUS)

| 8   | 7 | 6 | 5    | 4    | 3    | 2    | 1    | 0    |
|-----|---|---|------|------|------|------|------|------|
| TOF | 0 |   | CH5F | CH4F | CH3F | CH2F | CH1F | CH0F |
| w1c |   |   | w1c  | w1c  | w1c  | w1c  | w1c  | w1c  |

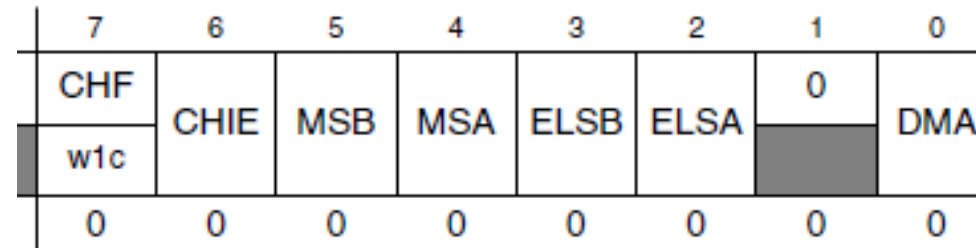
- TOF - Counter has overflowed
- CHxF - Channel event has occurred (event depends on mode)

# Major Channel Modes

- Input Capture Mode
  - Capture timer's value when input signal changes
    - Rising edge, falling edge, both
  - How long after I started the timer did the input change?
    - Measure time delay
- Output Compare Mode
  - Modify output signal when timer reaches specified value
    - Set, clear, pulse, toggle (invert)
  - Make a pulse of specified width
  - Make a pulse after specified delay
- Pulse Width Modulation
  - Make a series of pulses of specified width and frequency

# Channel Configuration and Value

## ■ Configuration:TPMx\_CnSC



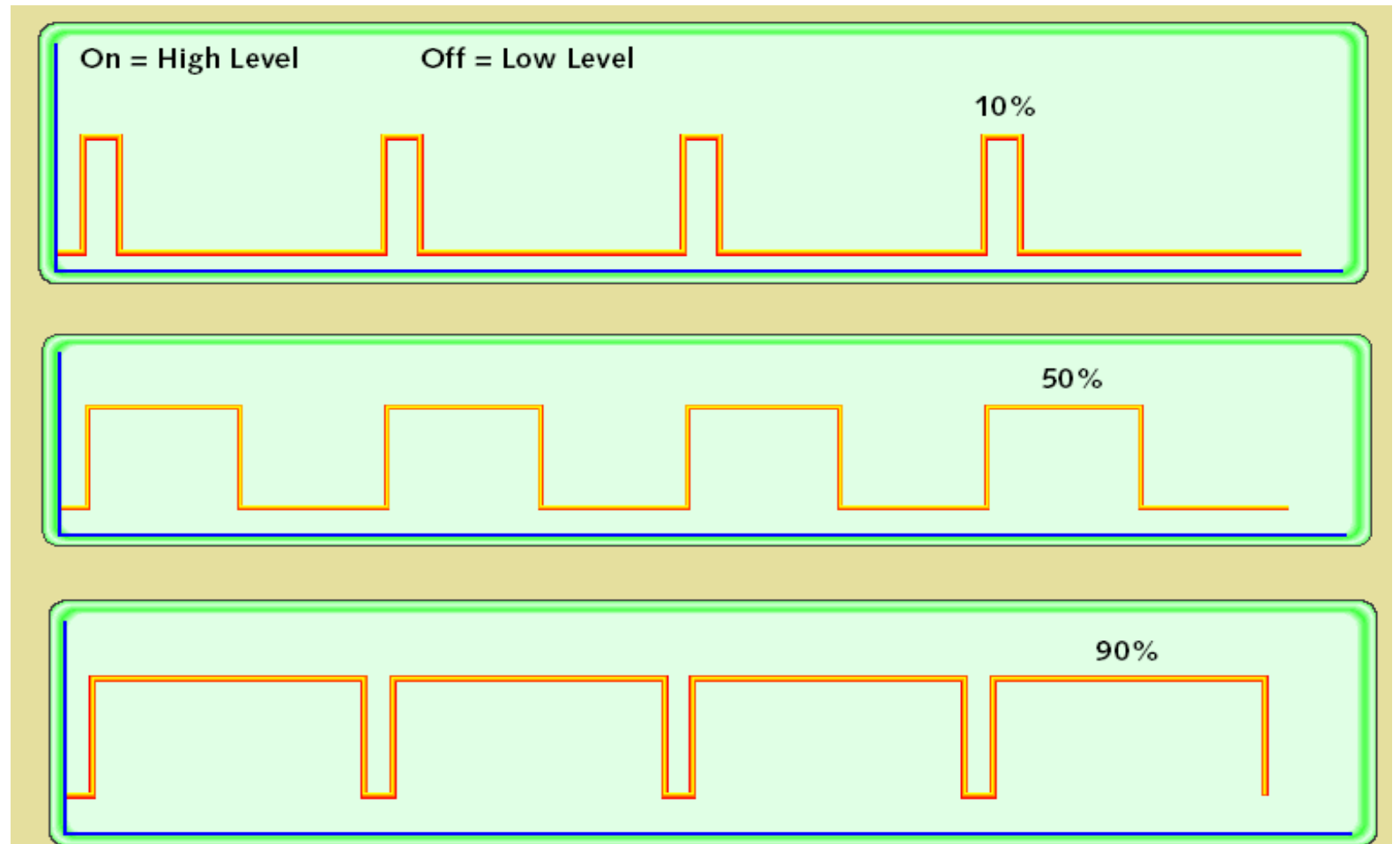
- CHF - set when event occurs
- CHIE - enable channel to generate an interrupt
- MSB:MSA - mode select
- ELSB:ELSA - edge or level select
- DMA - enable DMA transfers

## ■ Value:TPMx\_CnV

- 16-bit value for output compare or input capture

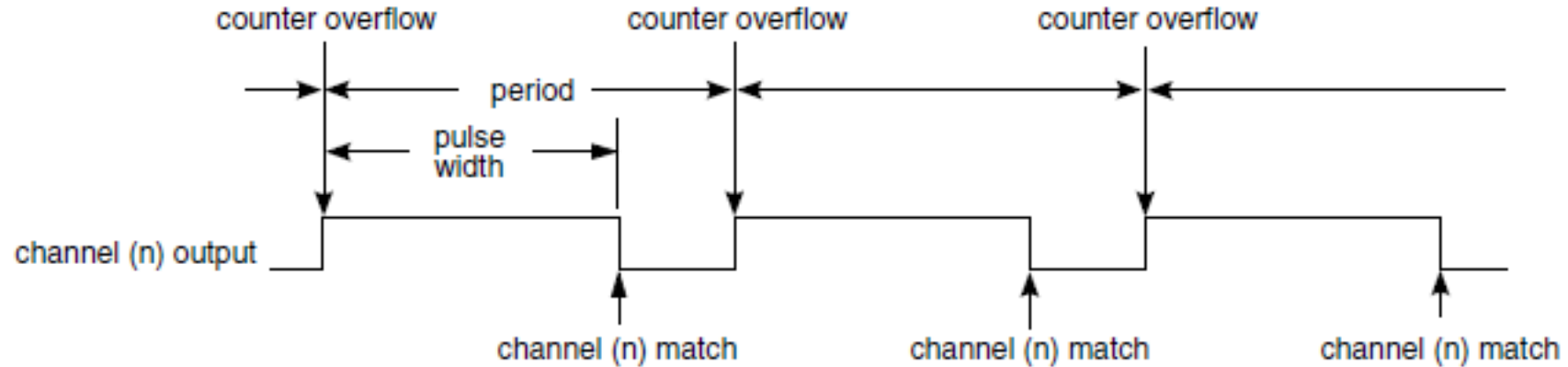
# Pulse-Width Modulation

- PWM signal characteristics
  - Period –  $1/(\text{modulation frequency})$
  - On-time – amount of time that each pulse is on (asserted)
  - Duty-cycle – on-time/period
  - Adjust *on-time* (hence *duty cycle*) to represent the analog value



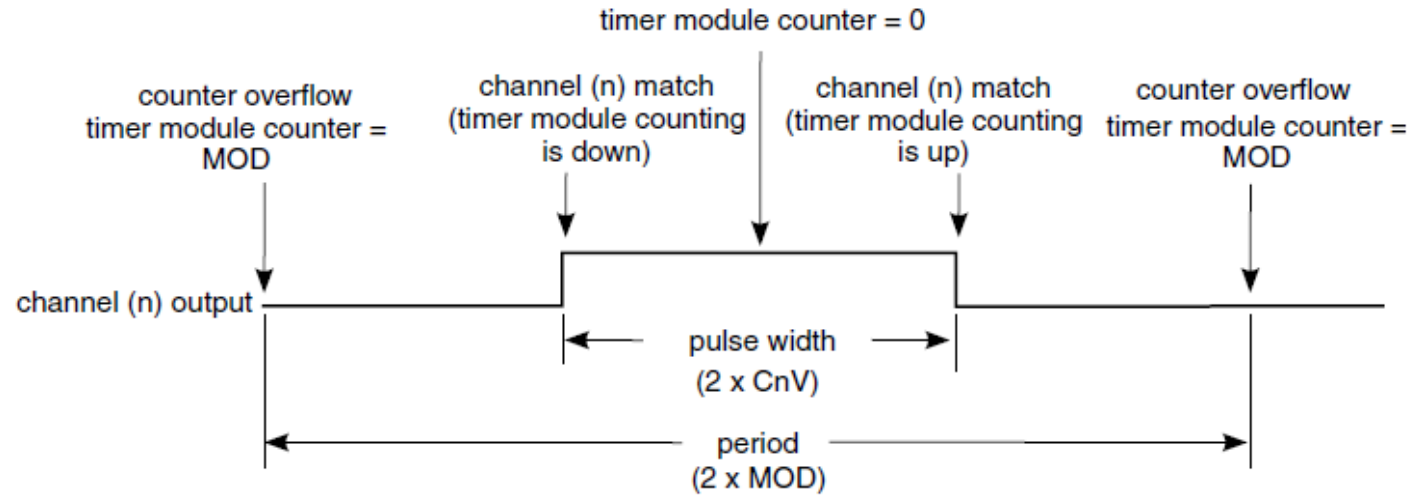


# TPM Channel for PWM Mode



- Edge-aligned - leading edges of signals from all PWM channels are aligned
  - Uses count up mode
  - Period = (MOD + 1) cycles
  - Pulse width = (CnV) cycles
- MSnB:MSnA = 10, CPWMS = 0
  - ELSnB:ELSnA = 10 - high-true pulses
  - ELSnB:ELSnA = x1 - low-true pulses

# TPM Channel for PWM Mode



- Center-aligned - centers of signals from all PWM channels are aligned
  - Uses count up/down mode
  - Period =  $2 \times \text{MOD}$  cycles.
  - Pulse width =  $2 \times \text{CnV}$  cycles
- MSnB:MSnA = 10, CPWMS = 1
  - ELSnB:ELSnA = 10 - high-true pulses
  - ELSnB:ELSnA = x1 - low-true pulses

# Let's Code It!

- We will review this week's Lab so you know what to prepare!

# The End!

- Now Let's Communicate!