Customer Training Workshop

Traveo™ II Timer/Counter/Pulse-Width Modulator (TCPWM)

Q1 2021

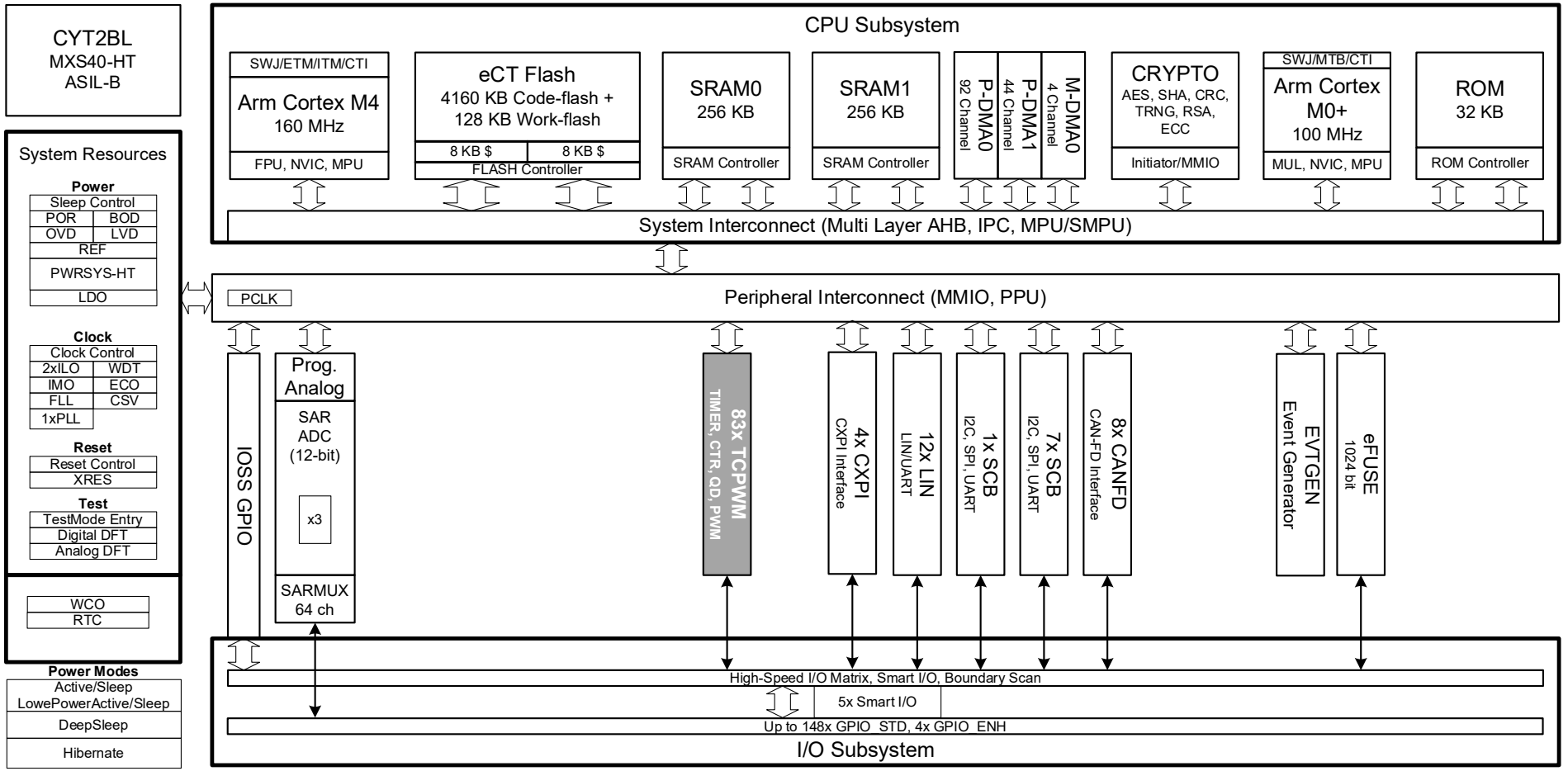
Target Products

› Target product list for this training material:

Family Category	Series	Code Flash Memory Size
Traveo™ II Automotive Body Controller Entry	CYT2B6	Up to 576 KB
Traveo II Automotive Body Controller Entry	CYT2B7	Up to 1088 KB
Traveo II Automotive Body Controller Entry	CYT2B9	Up to 2112 KB
Traveo II Automotive Body Controller Entry	CYT2BL	Up to 4160 KB
Traveo II Automotive Body Controller High	CYT3BB/ CYT4BB	Up to 4160 KB
Traveo II Automotive Body Controller High	CYT4BF	Up to 8384 KB
Traveo II Automotive Cluster	CYT3DL	Up to 4160 KB
Traveo II Automotive Cluster	CYT4DN	Up to 6336 KB

Introduction to Traveo II Body Controller Entry

› TCPWM in Traveo II is located in the peripheral blocks.

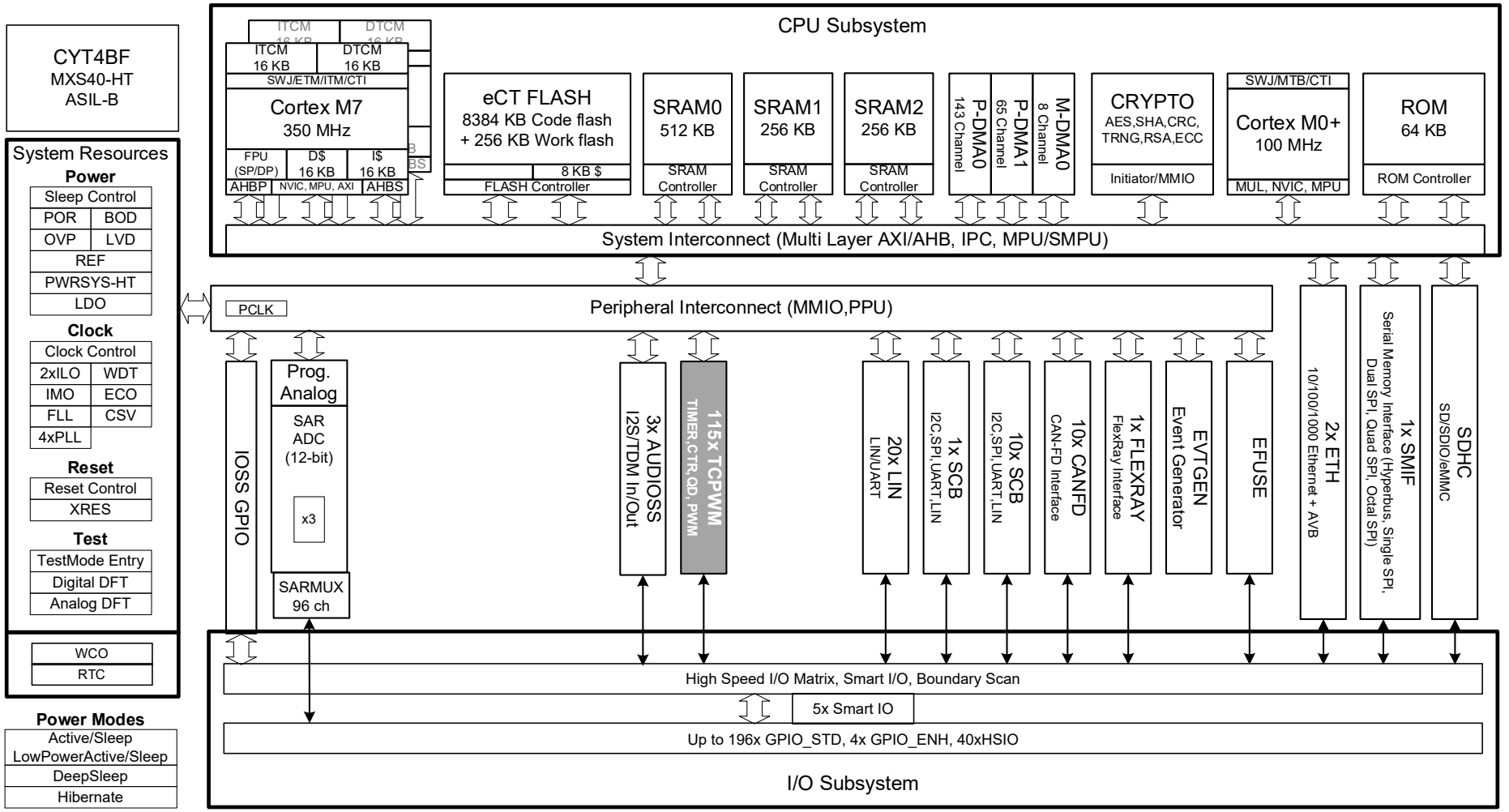


Hint Bar

Review TRM section 4.1 for additional details specific to the CPUSS

Introduction to Traveo II Body Controller High

› TCPWM in Traveo II is located in the peripheral blocks.

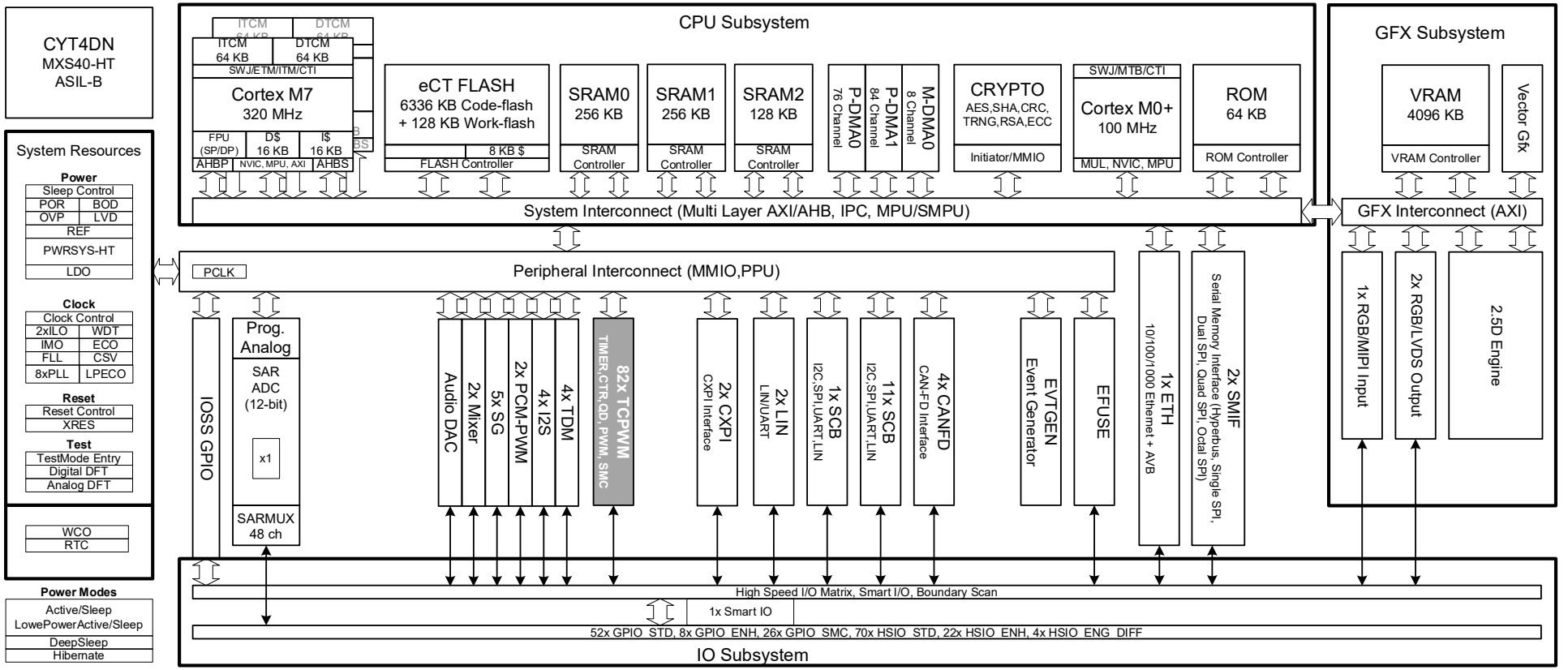


Hint Bar

Review TRM section 4.1 for additional details specific to the CPUSS

Introduction to Traveo II Cluster

› TCPWM in Traveo II is located in the peripheral blocks.



Hint Bar

Review TRM section 4.1 for additional details specific to the CPUSS

TCPWM Overview

- The Timer/Counter/Pulse-Width Modulator (TCPWM) block in Traveo II implements the following functionality:
  - Timer
  - Capture
  - Quadrature Decoder
  - Pulse-Width Modulation (PWM)
  - PWM with Dead Time Insertion (PWM_DT)
  - PWM Pseudo-Random (PWM_PR)
  - Shift Register (SR)

Hint Bar

Review the TCPWM chapter in the TRM for additional details

TCPWM Features (1/2)

- Counter specification
  - Supports up to four counter groups (device-specific1)
  - Each counter group consists of up to 256 counters (counter group-specific1)
- Each counter
  - Can run in one of the seven function modes
  - Supports various counting modes
    - One-shot mode and Continuous mode (Up/Down/Up-down)
  - Selects input signals
    - Start, Reload, Stop, Count, and two Capture event signals
  - Generates output signals
    - Two output triggers, PWM output, and interrupt
  - Supports debug mode

Hint Bar

Review the TCPWM chapter in the TRM for additional details

TCPWM Features (2/2)

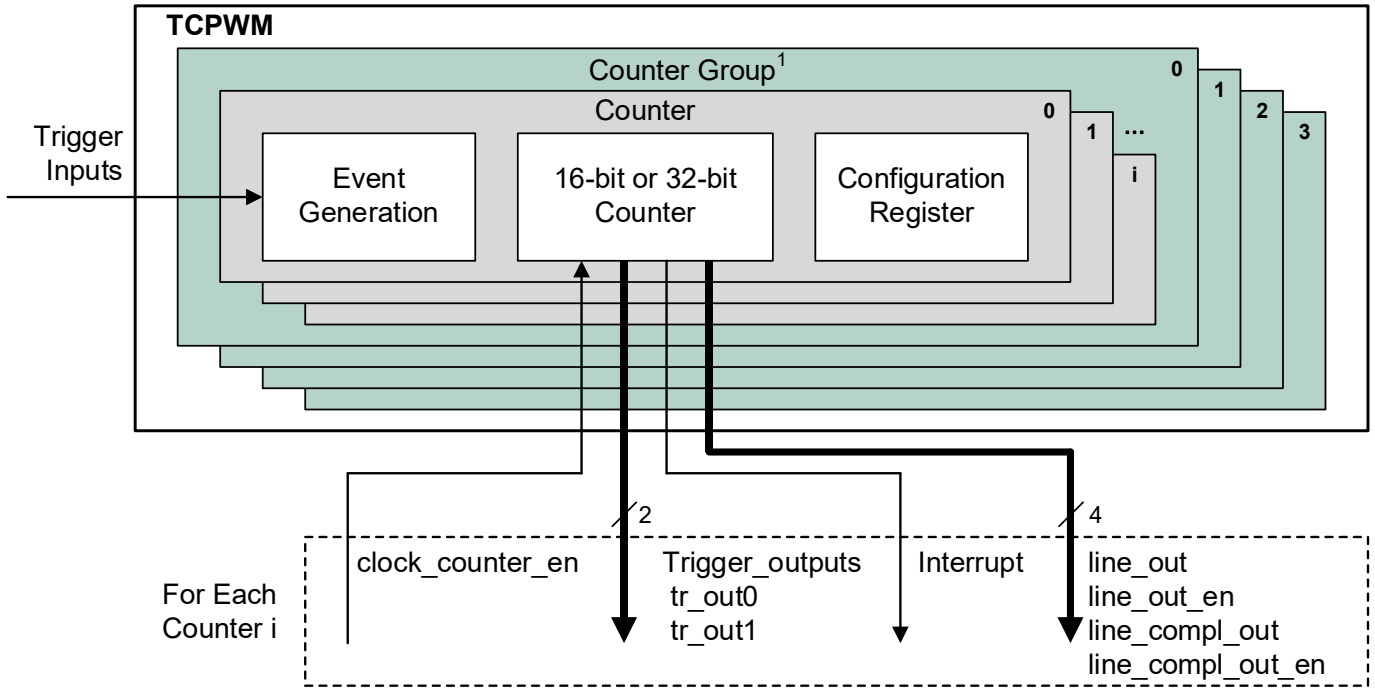
- Motor control functions are supported only in the “16-bit for motor” counter group
- Asymmetric PWM output control
  - This function has a different compare value when counting up and down
- PWM output selection for stepper motor
  - Two PWM output signals (line_out and line_compl_out) can be output by selecting one of the following values:
    - Constant low (0) or high (1)
    - PWM signal/Inverted PWM signal
    - Z (high impedance)
- Extend dead time
  - Dead time can be extended from 8 bits to 16 bits

Hint Bar

Review the TCPWM chapter in the TRM for additional details

TCPWM Block Diagram

- TCPWM components
  - Event generation
- 16-bit or 32-bit counter



[1] CYT2B5/B7 (176-pin) supports three counter groups. Refer to the respective device datasheet for the number of counter groups supported by each device.
- Counter Group 0: Up to 63 ch (16 bit)
- Counter Group 1: 12 ch (16 bit for motor)
- Counter Group 2: 4 ch (32 bit)
- Counter Group 3: Not available

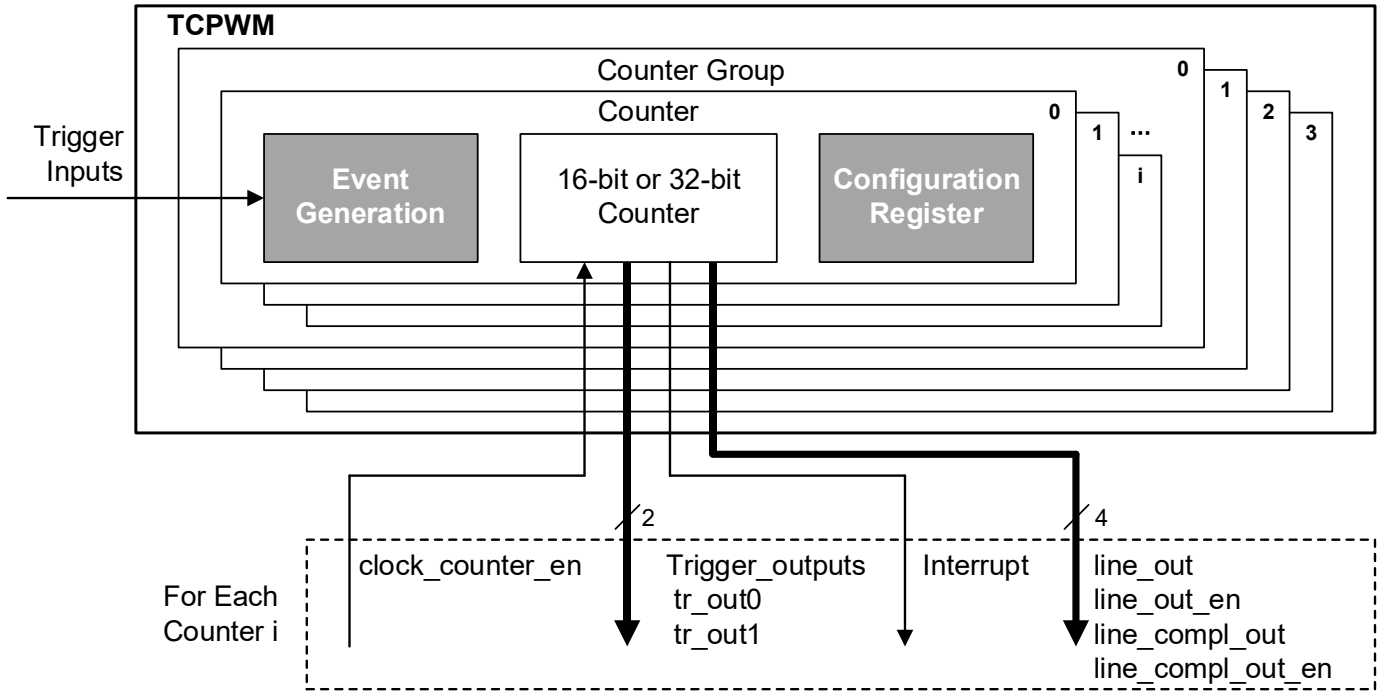
Hint Bar

Review the TCPWM chapter in the TRM for additional details

Refer to the Features List in the datasheet for additional details

Event Generation

- Event generation functions
  - Input trigger selection
  - Event detection

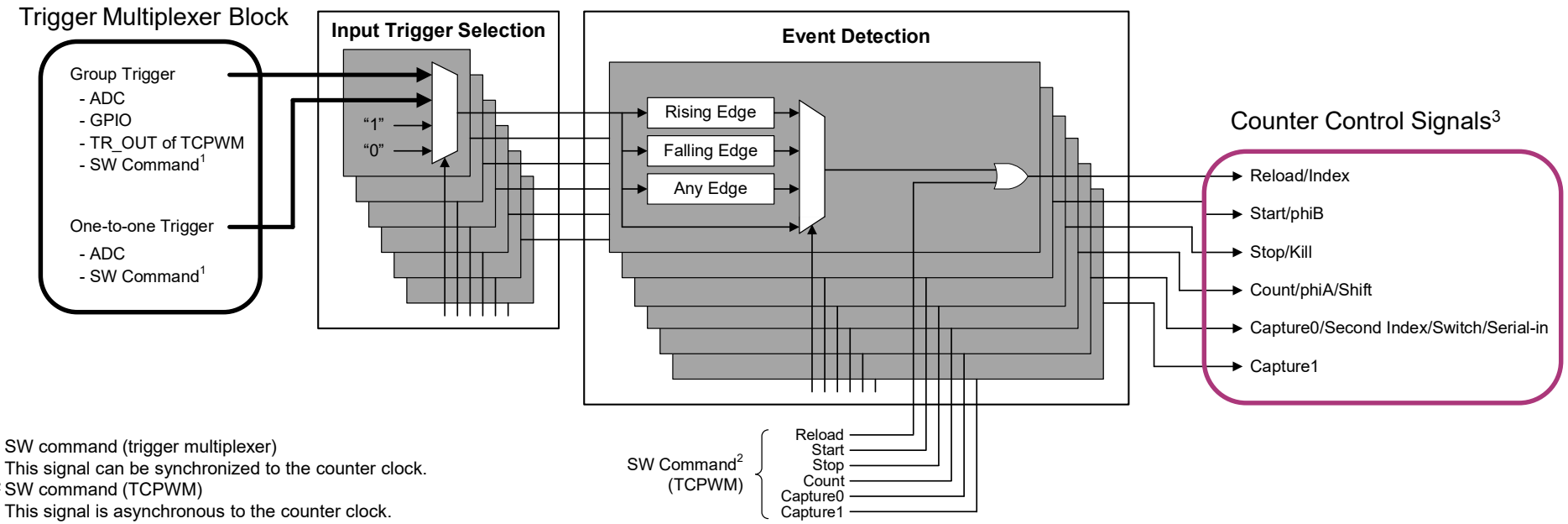


Hint Bar

Review the TCPWM chapter in the TRM for additional details

Input Trigger Selection and Event Detection

- The counter control signals have the following features:
  - Selects input signals from two types of trigger multiplexer signals
    - Group trigger and one-to-one trigger
    - The constant “1” and “0” are also used as input triggers
  - Detects the selected edge from four types
    - Rising edge, falling edge, any edge, and no edge
  - Supports software (SW) commands
    - SW command of trigger multiplexer/SW command of TCPWM



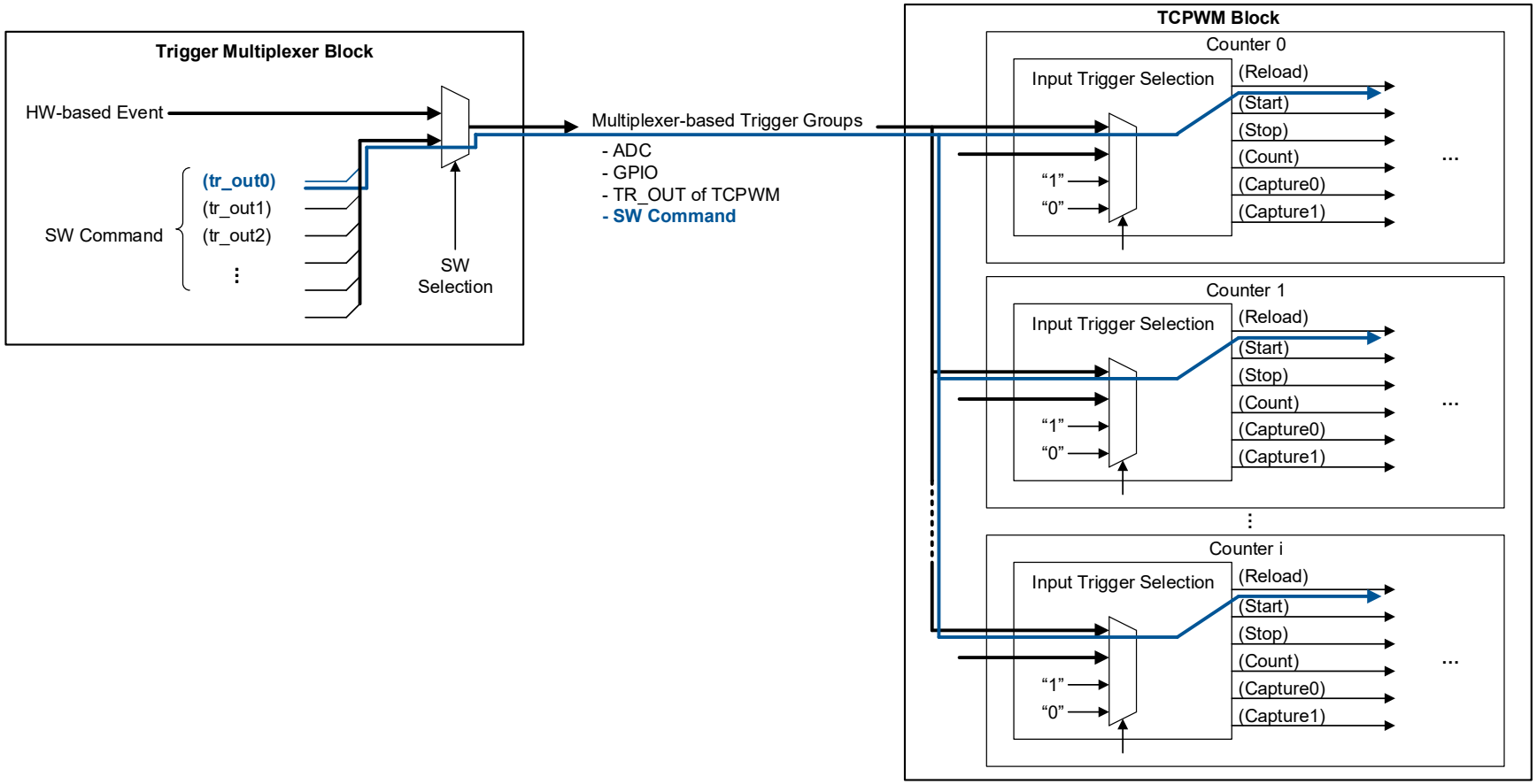
[1] SW command (trigger multiplexer)
This signal can be synchronized to the counter clock.
[2] SW command (TCPWM)
This signal is asynchronous to the counter clock.
[3] The name of the counter control signal is changed by the counter function.

Hint Bar

Review the TCPWM chapter in the TRM for additional details

Simultaneous Activation

- Advantage
  - When a trigger multiplexer block SW command is used, the TCPWM counters can also be activated at the same time

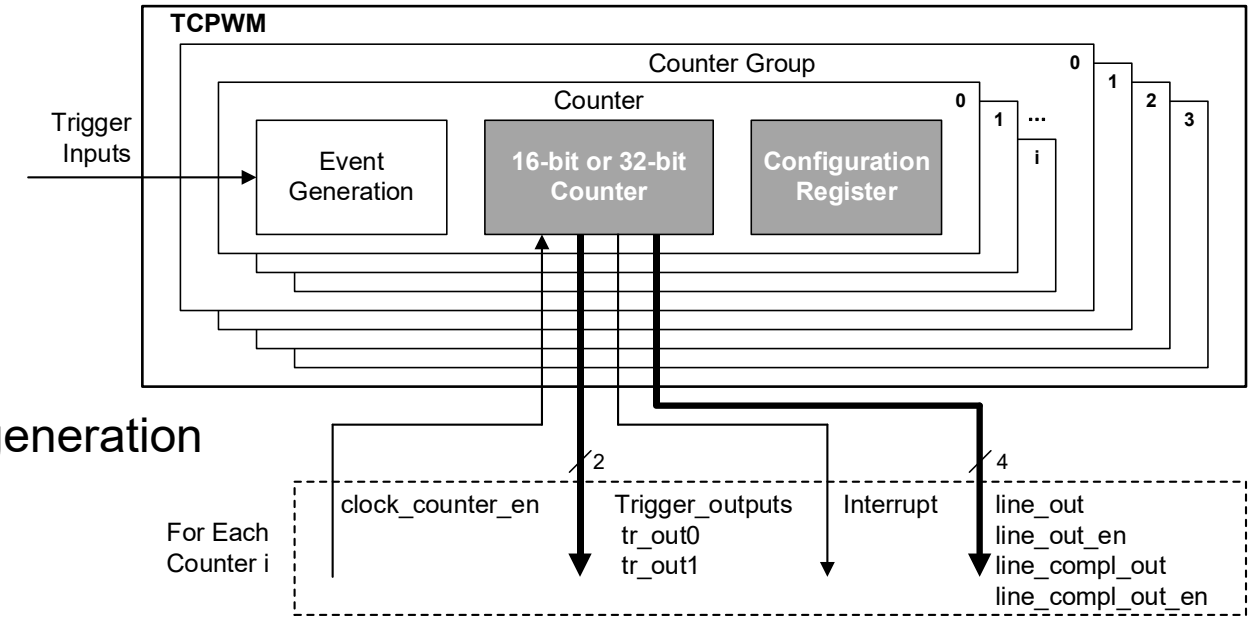


Hint Bar

Review the TCPWM chapter in the TRM for additional details

16-bit or 32-bit Counter

- Functions
  - Timer
  - Capture
  - Quadrature Decode
  - PWM
  - PWM_DT
  - PWM_PR
  - SR
- Trigger output and interrupt generation
- Clock enable
- Line output
- Debug mode

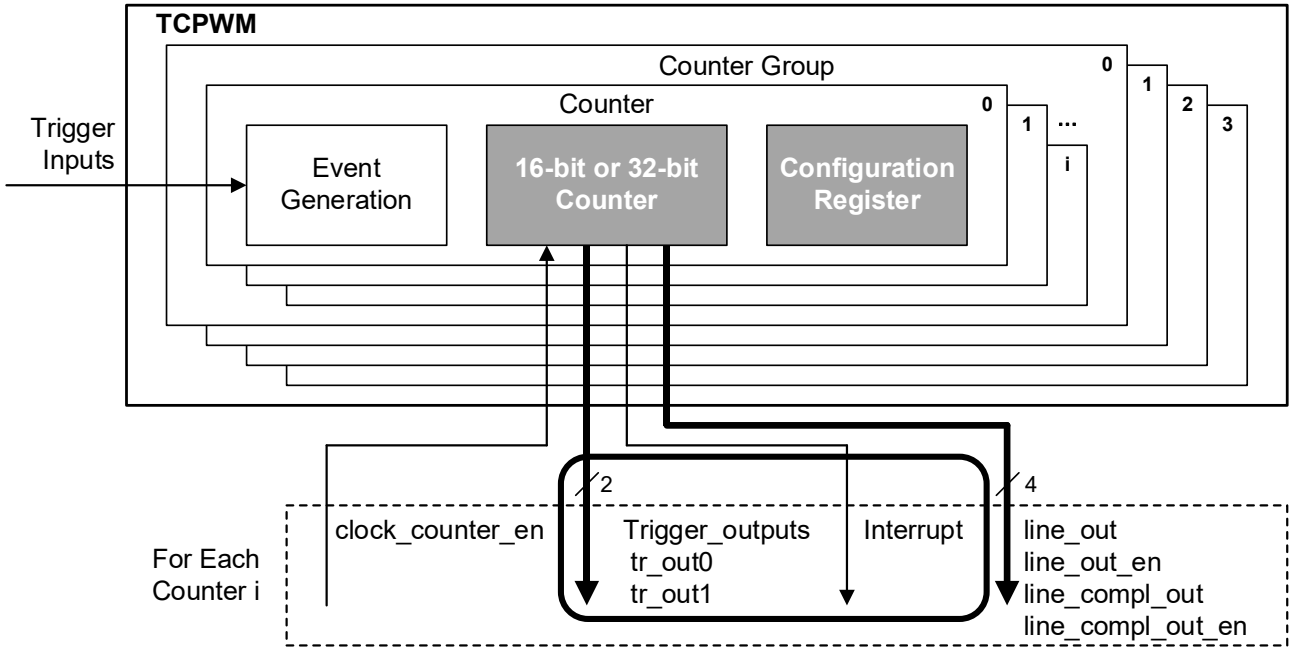


Hint Bar

Review the TCPWM chapter in the TRM for additional details

Trigger Output and Interrupt Generation Features

- Each counter has trigger outputs (TR_OUT0/1) and an interrupt
  - Overflow (OV)
  - Underflow (UN)
  - Terminal Count (TC)
  - CC0/1_MATCH
  - LINE_OUT



Hint Bar

Review the TCPWM chapter in the TRM for additional details

Trigger Output and Interrupt Generation

- TR_OUT 0/1 and the interrupt are selected by SW

Factor	Description	TR_OUT0/1[1]	Interrupt
Overflow (OV)	An overflow event indicates that in an up counting mode, COUNTER equals PERIOD register, and is changed to a different value	✔	–
Underflow (UN)	An underflow event indicates that in a down counting mode, COUNTER equals 0, and is changed to a different value	✔	–
Terminal Count (TC)	A TC event is the logical OR of the underflow and overflow events	✔	✔
CC0/1_MATCH	This event is generated when the counter is running and one of the following conditions occur: - Counter equals the compare value - A capture event occurs and the CC0/1 and CC0/1_BUFF registers are updated	✔	✔
LINE_OUT	PWM output signal	✔	–

- TR_OUT can be routed through the trigger multiplexer to other peripherals on the device

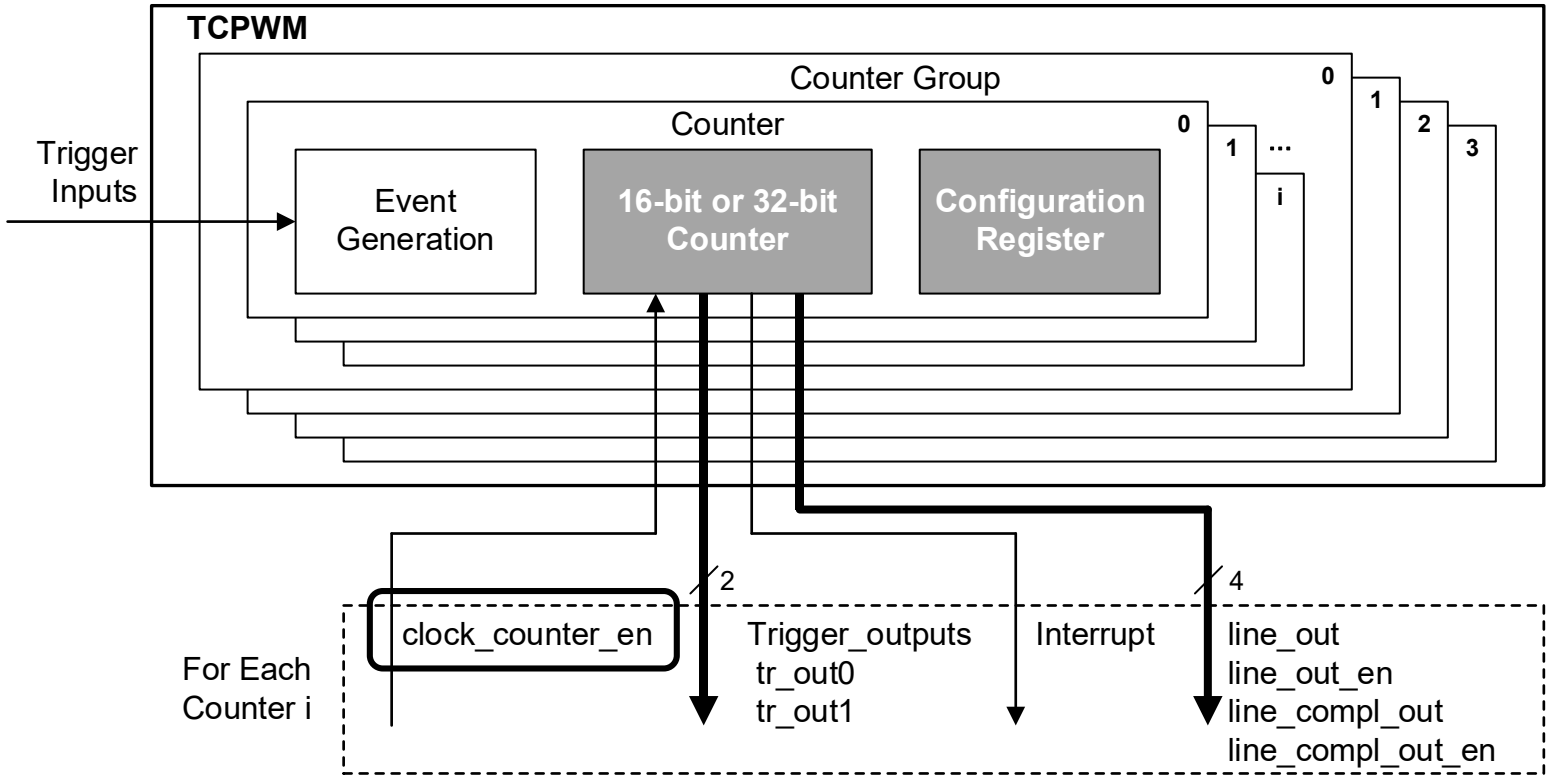
Hint Bar

Review the TCPWM chapter in the TRM for additional details

[1] TR_OUT0/1 can also be disabled.

Clock Enable

- A counter increments or decrements by ‘1’ for every counter clock cycle in which a count event is detected



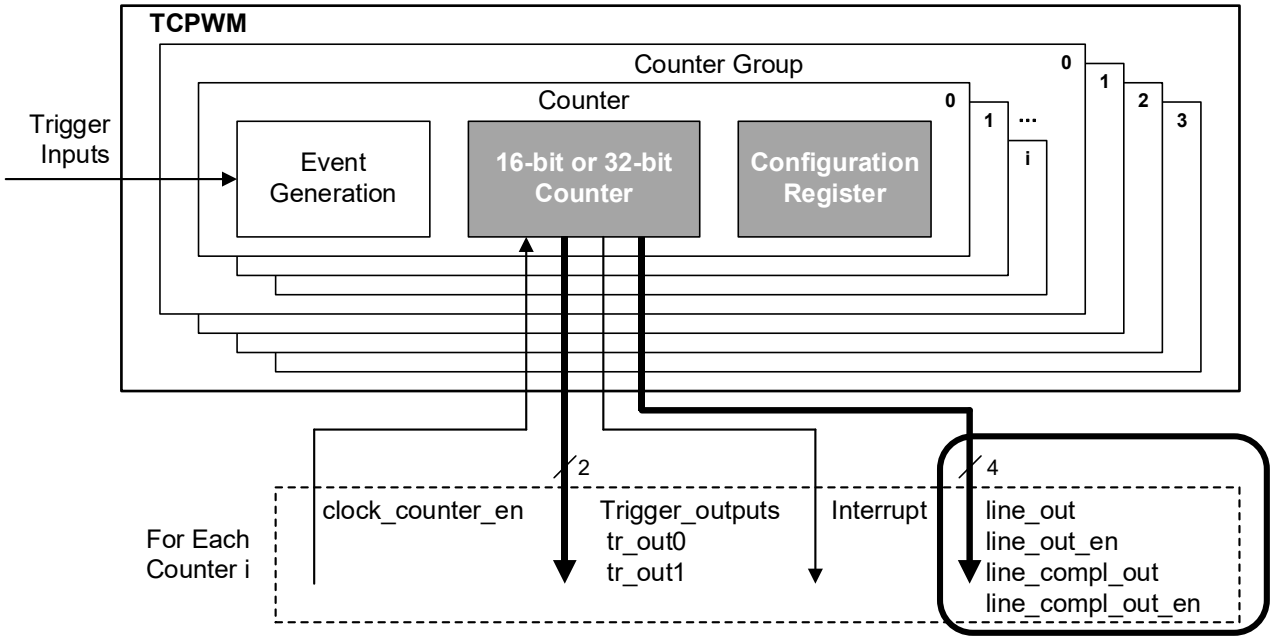
Hint Bar

Review the TCPWM chapter in the TRM for additional details

Line Output

- Each counter can produce two output signals
  - Line_out
  - Line_compl_out

    These signals are supported in PWM, PWM_DT, PWM_PR, and SR functionality
- Output signal condition depends on two signals
  - Line_out_en
  - Line_compl_out_en



Hint Bar

Review the TCPWM chapter in the TRM for additional details

Timer

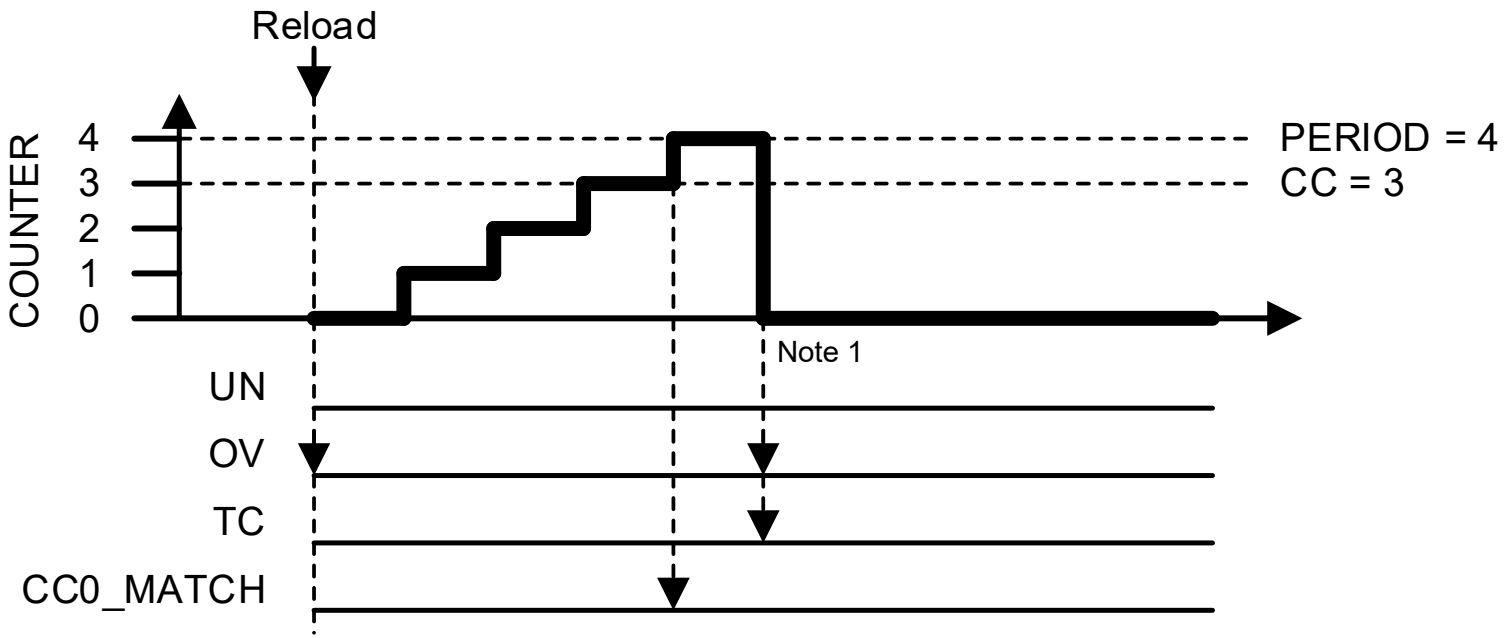
- The timer mode is commonly used to measure the time an event occurs or to measure the time difference between two events
  - Event generation
    - SW selects the input triggers (Reload[1], Start[2], Stop[3], Count[4])
  - 16-bit or 32-bit counter
    - Auto reload
      - CC0/1 and CC0/1_BUFF are exchanged on a CC0/1_MATCH event
  - Counter operation
    - One-shot mode and Continuous mode
    - Up/Down modes (COUNT_UP/COUNT_DOWN/COUNT_UPDN1/2)
  - Signal output
    - TR_OUT0: OV, UN, TC, CC0_MATCH, CC1_MATCH
    - TR_OUT1: OV, UN, TC, CC0_MATCH, CC1_MATCH
    - Interrupt: TC, CC0_MATCH, CC1_MATCH

Hint Bar

Review the TCPWM chapter in the TRM for additional details

[1] Reload: Initializes and starts the counter.
[2] Start: Starts the counter.
[3] Stop: Stops the counter.
[4] Count: Count event (increments/decrements the counter).

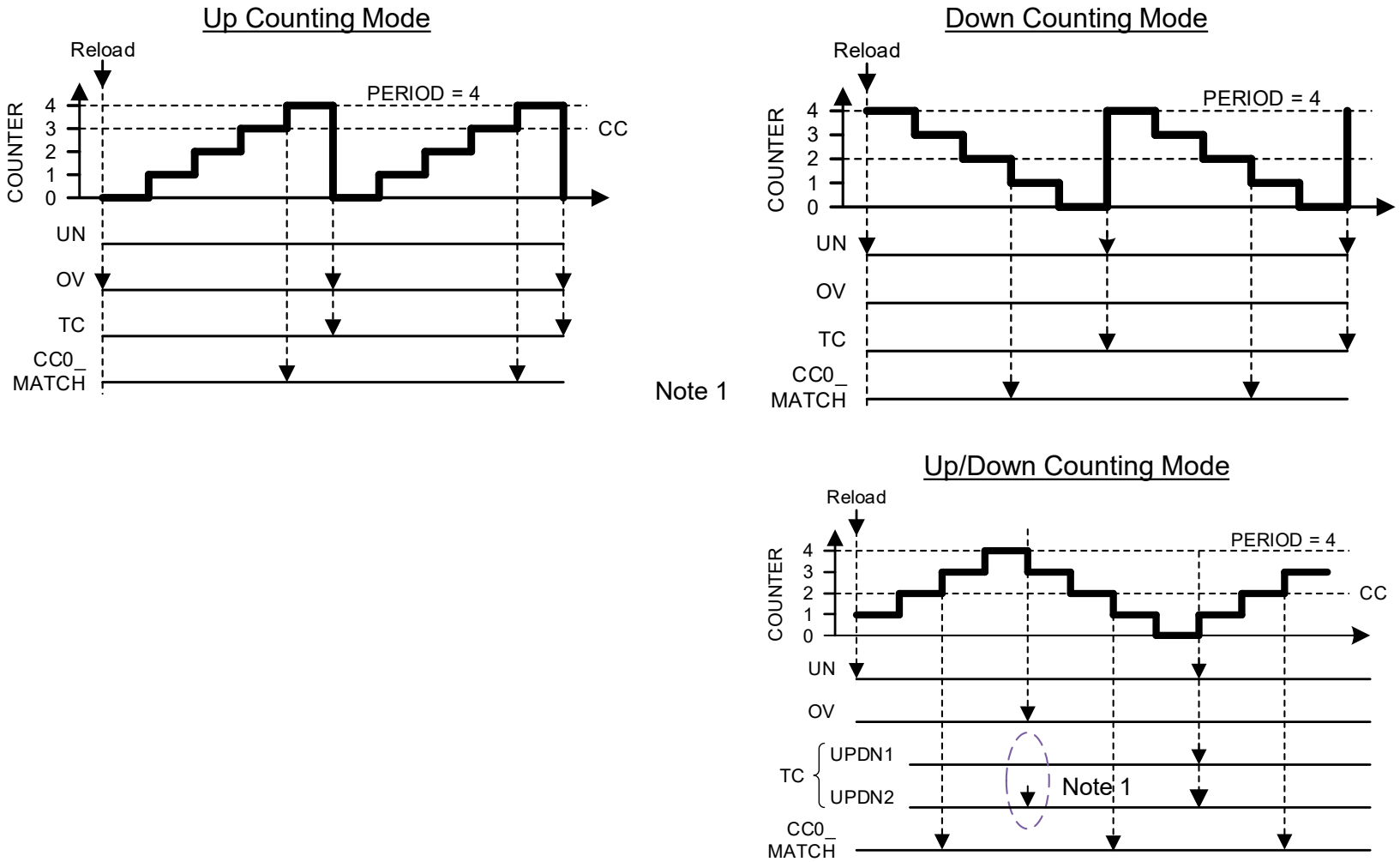
Timer Operation: One-shot Mode



Hint Bar

Review the TCPWM chapter in the TRM for additional details

Timer Operation: Up/Down modes (Continuous mode)



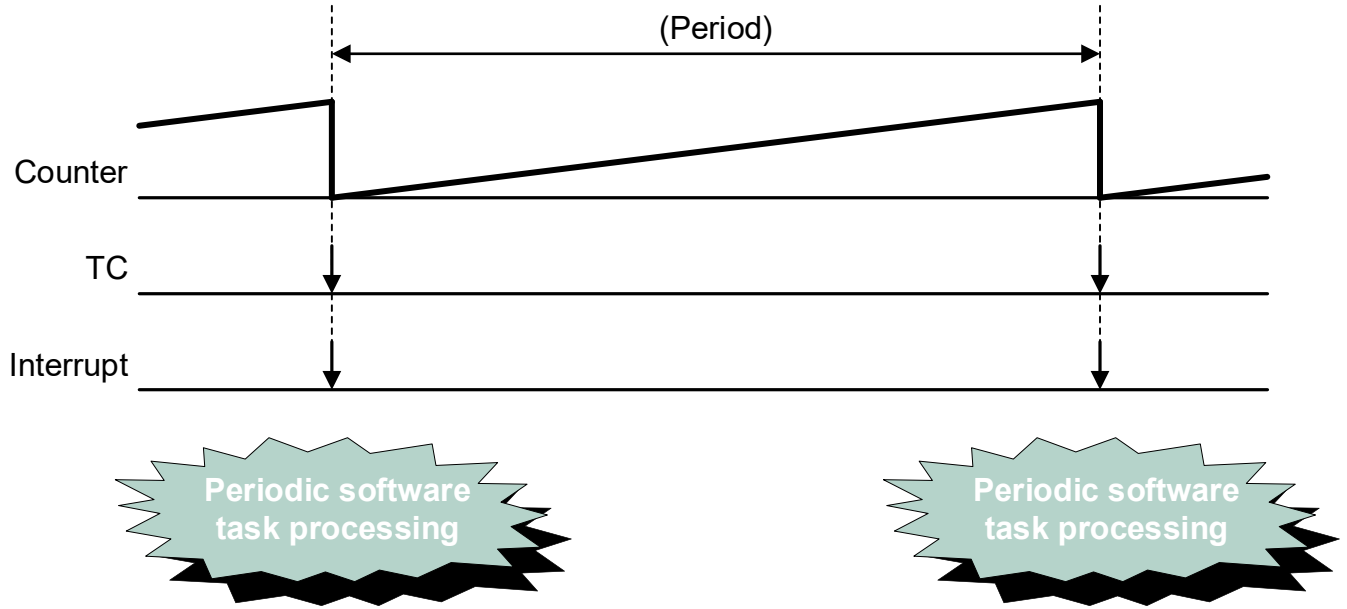
[1] In COUNT_UPDN2, TC is output when the counter overflows and underflows.

Hint Bar

Review the TCPWM chapter in the TRM for additional details

Timer Use Case

› Generate periodic interrupt for software task processing.



Hint Bar

Review the TCPWM chapter in the TRM for additional details

Capture

- Capture functionality uses the value of the counter to measure the width and period of the input pulse
  - Event generation
    - SW selects input triggers (Reload[1], Start[2], Stop[3], Count[4], Capture0/1[5])
  - 16-bit or 32-bit counter
    - Counter operation
      - One-shot mode and Continuous mode
      - Up/Down modes (COUNT_UP/COUNT_DOWN/COUNT_UPDN1/2)
  - Signal output
    - TR_OUT0: OV, UN, TC, CC0_MATCH, CC1_MATCH
    - TR_OUT1: OV, UN, TC, CC0_MATCH, CC1_MATCH
    - Interrupt: TC, CC0_MATCH, CC1_MATCH

Hint Bar

Review the TCPWM chapter in the TRM for additional details

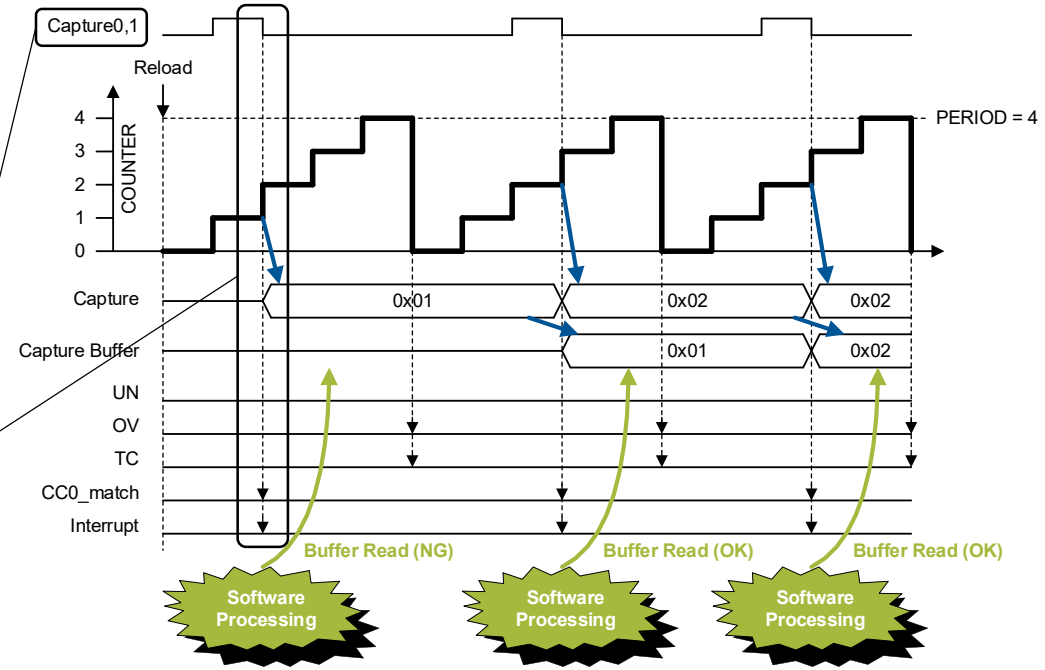
[1] Reload: Initializes and starts the counter.
[2] Start: Starts the counter.
[3] Stop: Stops the counter.
[4] Count: Count event increments/decrements the counter.
[5] Capture0/1: Copies the counter value to CC0/1 and CC0/1 to CC0/1_BUFF.

Capture Operation

- Capture using Capture0/1 signal
  - Measure between falling edges of Capture0 signal

Input pulse
- This signal is input from an external pin

When a capture event occurs
- COUNTER is copied into CC0/1
- CC0/1 is copied to CC0/1_BUFF register



- Advantage
  - Provides the following measurement functions:
    - “H” pulse width/“L” pulse width
    - Rising cycle/falling cycle
    - Edge-to-edge measurement
- Use Case
  - Measure the input pulse period (vehicle speed pulse)

Hint Bar

Review the TCPWM chapter in the TRM for additional details

Quadrature Decoder

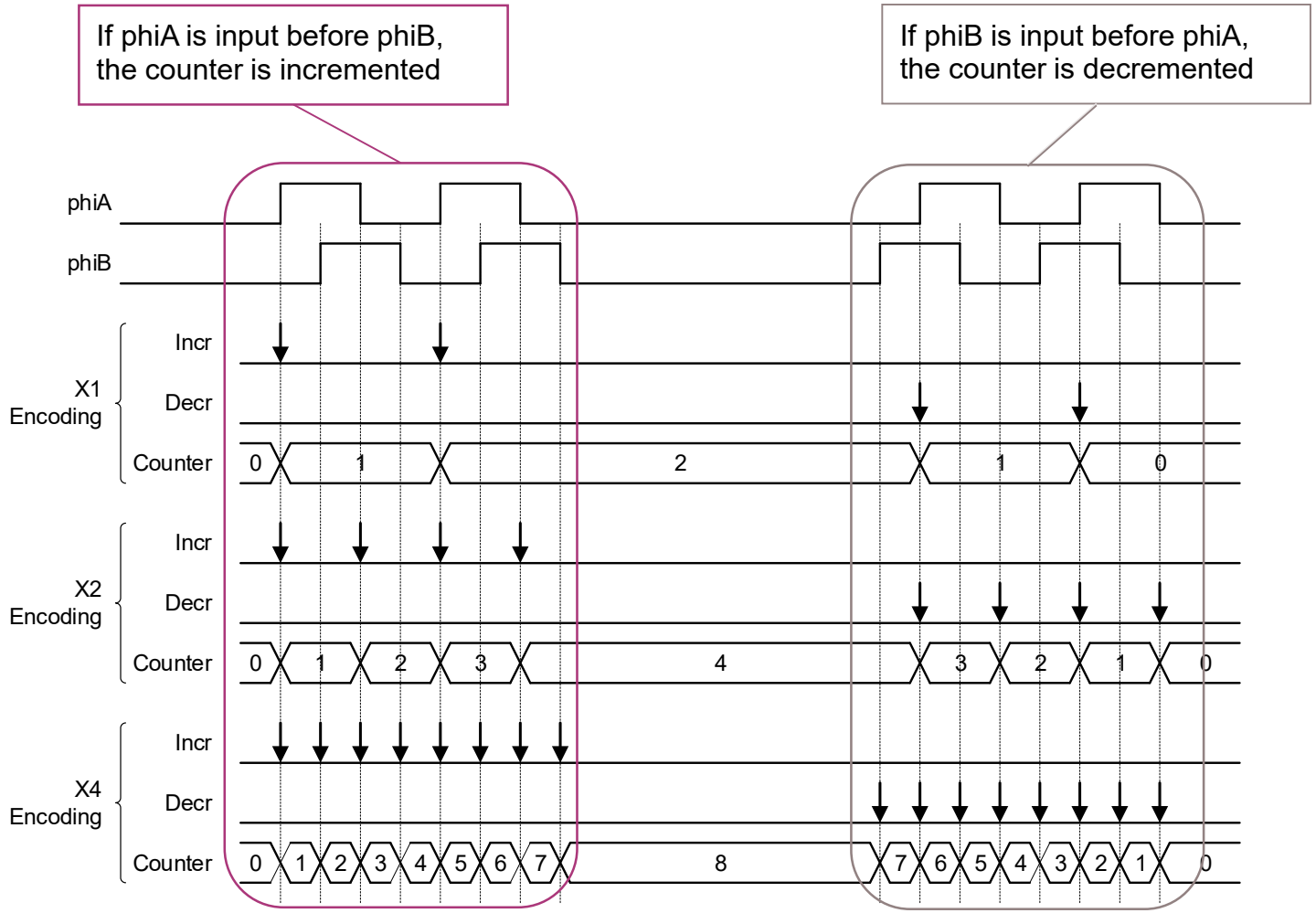
- The Quadrature Decoder has four Range modes and four Encoding modes
  - Range modes
    - QUAD_RANGE0
    - QUAD_RANGE0_CMP
    - QUAD_RANGE1_CMP
    - QUAD_RANGE1_CAPT
  - Encoding modes
    - X1 encoding
    - X2 encoding
    - X4 encoding
    - Up/down rotary count mode
- Each Range mode has:
  - Four encoding modes
  - Different input trigger for each range mode

Hint Bar

Review the TCPWM chapter in the TRM for additional details

Encoding Modes

- X1, X2, and X4 encoding modes

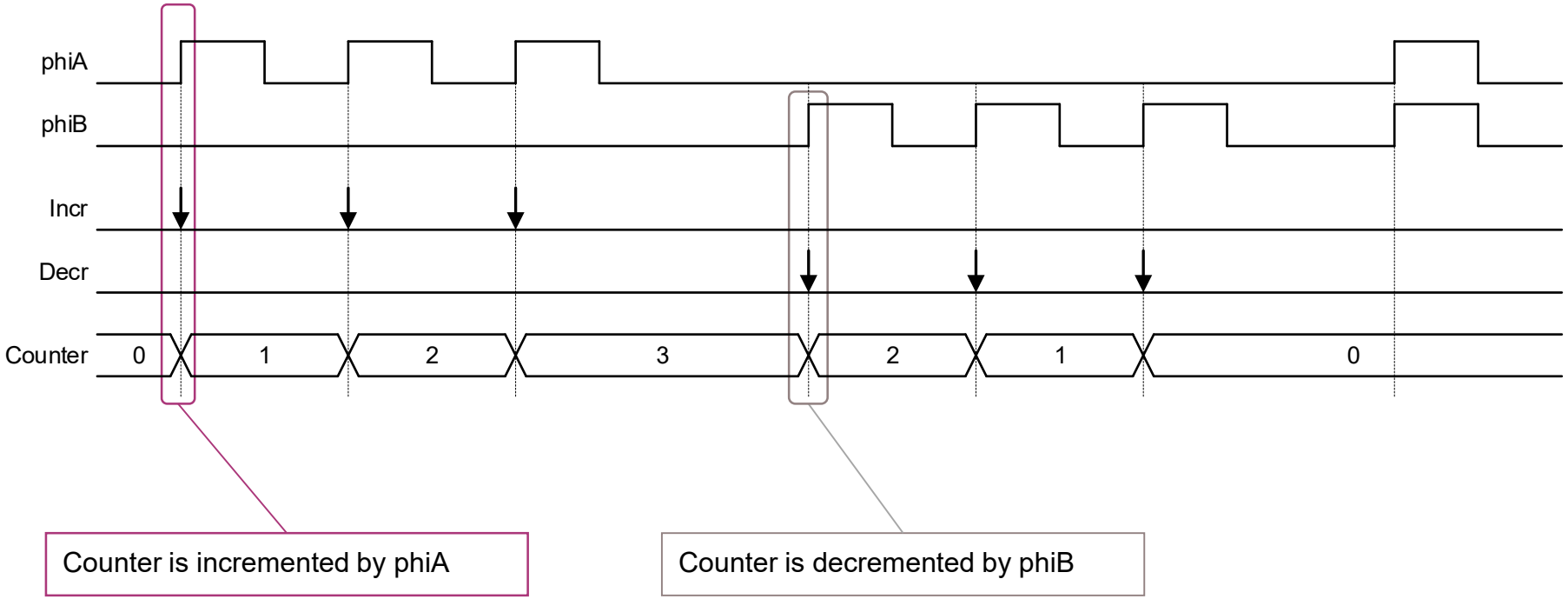


Hint Bar

Review the TCPWM chapter in the TRM for additional details

Encoding Modes

- Up/Down rotary count mode



Hint Bar

Review the TCPWM chapter in the TRM for additional details

Range Modes

Range Modes Supported	Counter Range	Counter Initial Value	Copy the Counter Value[1]	Compare Event[1]	Capture Event[1]
QUAD_RANGE0	0x0000 to 0xFFFF/0xFFFFFFFF	0x8000	✔	–	–
QUAD_RANGE0_CMP	0x0000 to 0xFFFF/0xFFFFFFFF	0x8000	✔	✔	–
QUAD_RANGE1_CMP	0x0000 to PERIOD	0x0000	–	✔	–
QUAD_RANGE1_CAPT[2]	0x0000 to PERIOD	0x0000	–	✔	✔

Hint Bar

Review the TCPWM chapter in the TRM for additional details

- Advantage
  - Output of the quadrature encoder can be decoded without external circuits

[1] For details about each item, see the respective time chart on the following pages - 29, 31, 33, 35.
[2] QUAD_RANGE1_CAPT mode provides the same functionality as the QUAD_RANGE1_CMP; the only difference is that capture functions are available.

QUAD_RANGE0

- Event generation
  - SW selects the input triggers (index[1], phiB[2], Stop[3], phiA[4])
- 16-bit or 32-bit counter
  - Counter operation
    - Four encoding modes can be used (X1, X2, X4, and up/down rotary count mode)
- Signal output
  - TR_OUT0: TC, CC0_MATCH
  - TR_OUT1: TC, CC0_MATCH
  - Interrupt: TC, CC0_MATCH

Hint Bar

Review the TCPWM chapter in the TRM for additional details

[1] index: Initializes and starts the counter.
[2] phiB: Quadrature phiB input.
[3] Stop: Stops the quadrature functionality.
[4] phiA: Quadrature phiA input.

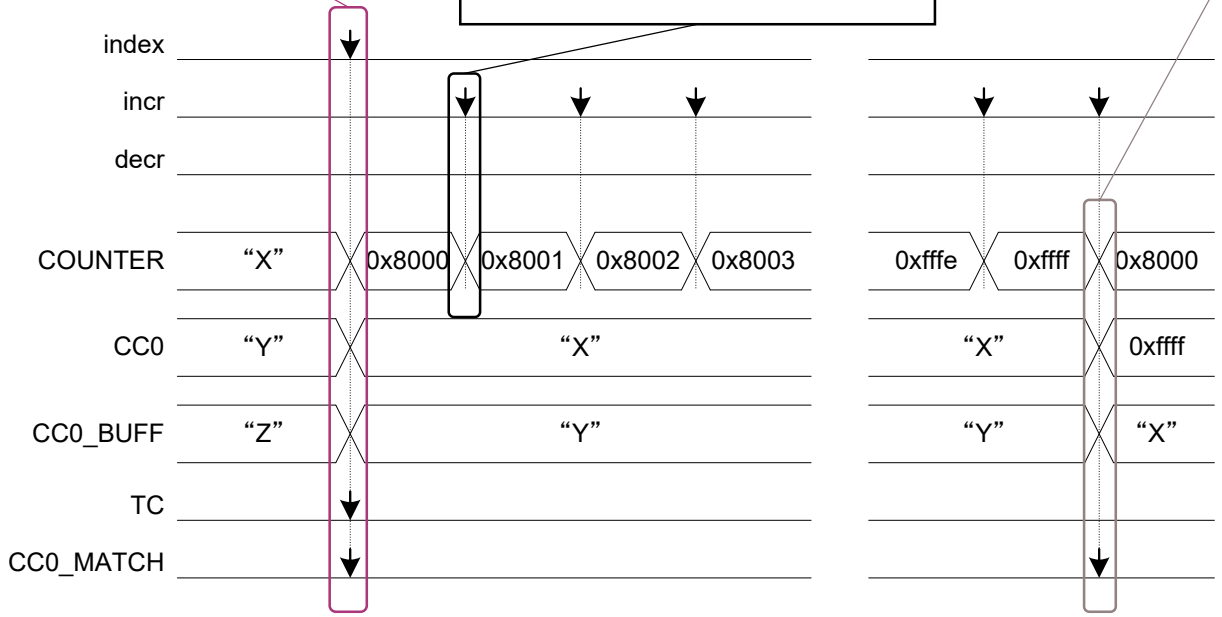
QUAD_RANGE0 Operation

› Operation example (X1 encoding mode)

On an index event:
- CC0 is copied to CC0_BUFF
- COUNTER is copied to CC0
- COUNTER is set to 0x8000
- TC and CC0_MATCH events are generated

When COUNTER is 0xFFFF:
- CC0 is copied to CC0_BUFF
- COUNTER (0xFFFF) is copied to CC0
- COUNTER is set to 0x8000
- CC0_MATCH event is generated

On an incr event:
- COUNTER is incremented



Hint Bar

Review the TCPWM chapter in the TRM for additional details

QUAD_RANGE0_CMP

- Event generation
  - SW selects the input triggers (index[1], phiB[2], Stop[3], phiA[4], second index[5])
- 16-bit or 32-bit counter
  - Counter operation
    - Four encoding modes can be used (X1, X2, X4, and up/down rotary count mode)
- Signal output
  - TR_OUT0: TC, CC0_MATCH, CC1_MATCH
  - TR_OUT1: TC, CC0_MATCH, CC1_MATCH
  - Interrupt: TC, CC0_MATCH, CC1_MATCH

Hint Bar

Review the TCPWM chapter in the TRM for additional details

[1] index: Initializes and starts the counter.
[2] phiB: Quadrature phiB input.
[3] Stop: Stops the quadrature functionality.
[4] phiA: Quadrature phiA input.
[5] Second index: This event acts as a second quadrature index input. It has the same function as the index event.

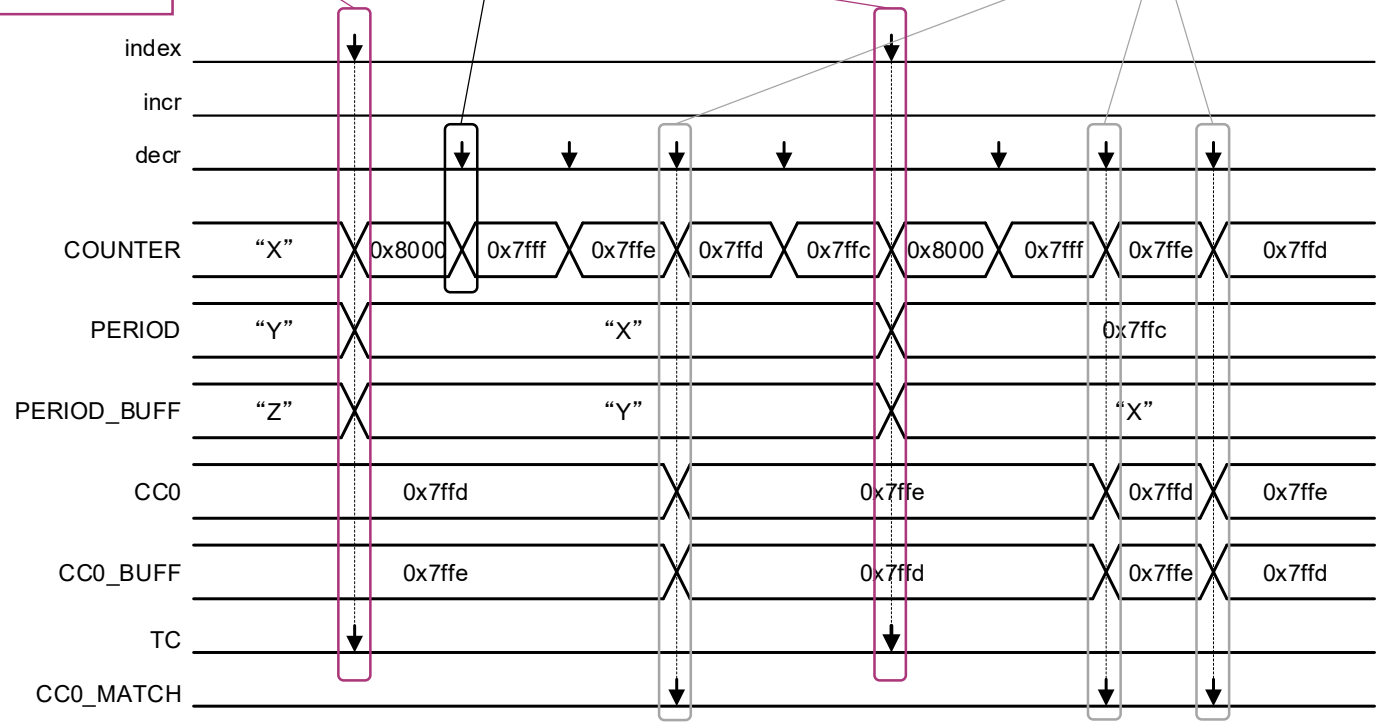
QUAD_RANGE0_CMP Operation

› Operation example (X1 encoding mode)

On an index event:
- PERIOD is copied to PERIOD_BUFF
- COUNTER is copied to PERIOD
- COUNTER is set to 0x8000
- TC event is generated

On a decrement event:
- COUNTER is decremented

On a compare event:
- CC0/1_MATCH event is generated when COUNTER changes to a state in which COUNTER equals CC0/1
- CC0/1 and CC0/1_BUFF are exchanged on a CC0/1_MATCH event



Hint Bar

Review the TCPWM chapter in the TRM for additional details

QUAD_RANGE1_CMP

- Event generation
  - SW selects the input triggers (index[1], phiB[2], Stop[3], phiA[4])
- 16-bit or 32-bit counter
  - Counter operation
  - Four encoding modes can be used (X1, X2, X4, and up/down rotary count mode)
- Signal output
  - TR_OUT0: OV, UN, TC, CC0_MATCH, CC1_MATCH
  - TR_OUT1: OV, UN, TC, CC0_MATCH, CC1_MATCH
  - Interrupt: TC, CC0_MATCH, CC1_MATCH

Hint Bar

Review the TCPWM chapter in the TRM for additional details

[1] index: Initializes and starts the counter.
[2] phiB: Quadrature phiB input.
[3] Stop: Stops the quadrature functionality.
[4] phiA: This event acts as a quadrature phiA input.

QUAD_RANGE1_CMP Operation

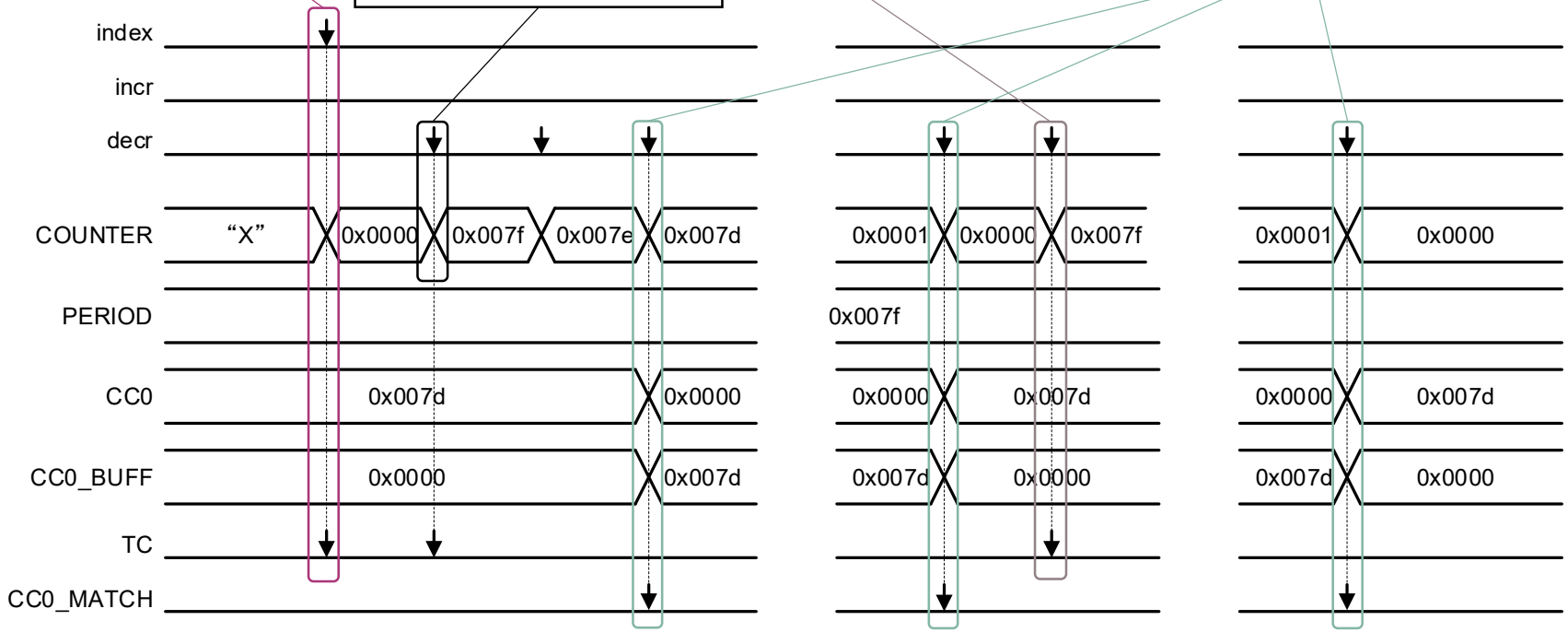
› Operation example (X1 encoding mode)

On an index event:
- COUNTER is set to 0x0000
- TC event is generated

When COUNTER is 0x0000 and decrementing:
- COUNTER is set to PERIOD
- TC event is generated

On a compare event
- CC0/1_MATCH event is generated when COUNTER changes to a state in which COUNTER equals CC0/1
- CC0/1 and CC0/1_BUFF are exchanged on a CC0/1_MATCH event

On a decrement event:
- COUNTER is decremented



Hint Bar

Review the TCPWM chapter in the TRM for additional details

QUAD_RANGE1_CAPT

- Event generation
  - SW selects the input triggers (index[1], phiB[2], Stop[3], phiA[4], Capture0[5]/1[6])
- 16-bit or 32-bit counter
  - Counter operation
    - Four encoding modes can be used (X1, X2, X4, and up/down rotary count mode)
- Signal output
  - TR_OUT0: OV, UN, TC, CC0_MATCH, CC1_MATCH
  - TR_OUT1: OV, UN, TC, CC0_MATCH, CC1_MATCH
  - Interrupt: TC, CC0_MATCH, CC1_MATCH

Hint Bar

Review the TCPWM chapter in the TRM for additional details

[1] index: Initializes and starts the counter.
[2] phiB: Quadrature phiB input.
[3] Stop: Stops the quadrature functionality.
[4] phiA: This event acts as a quadrature phiA input.
[5] Capture0: Capture event to copy COUNTER to CC0, CC0 to CC0_BUFF.
[6] Capture1: Second capture event to copy COUNTER to CC1, CC1 to CC1_BUFF.

QUAD_RANGE1_CAPT Operation

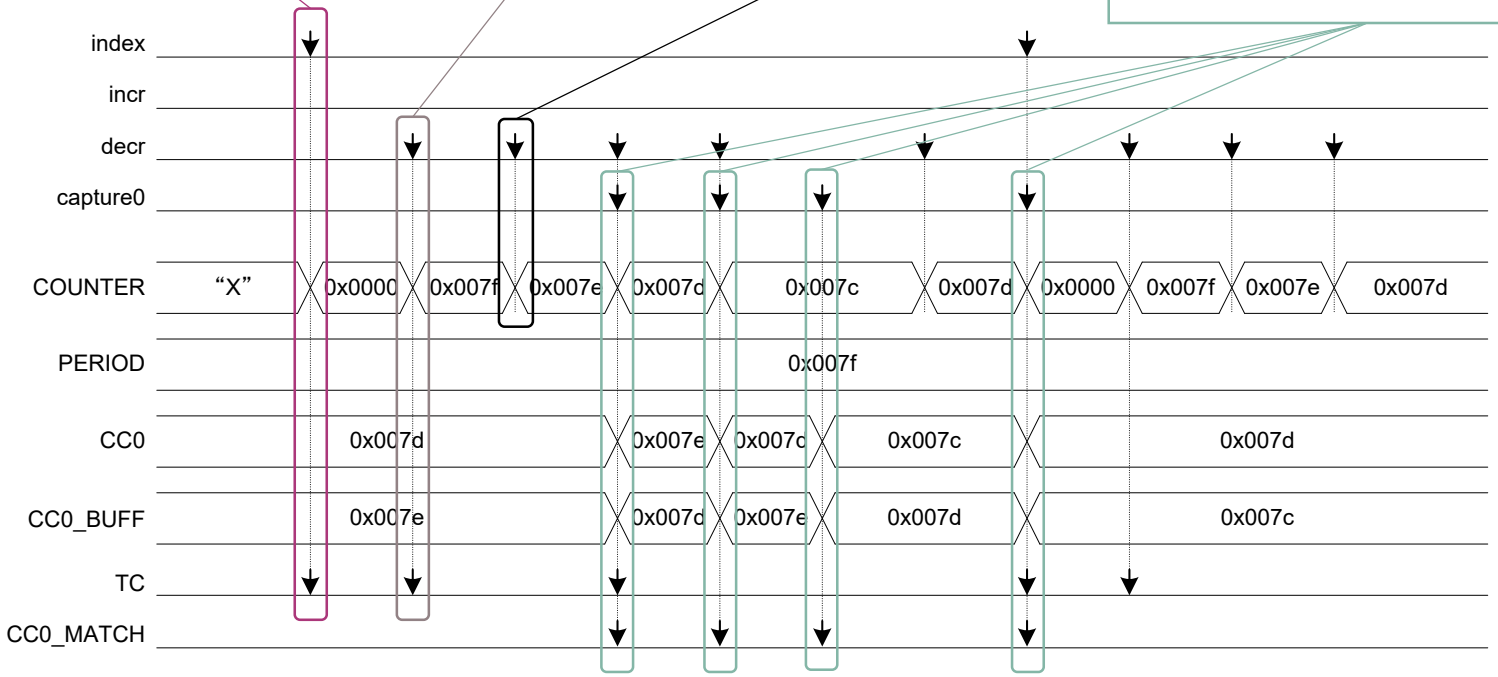
› Operation example (X1 encoding mode).

On an index event:
- COUNTER is set to 0x0000
- TC event is generated

When COUNTER is 0x0000 and decrementing:
- COUNTER is set to PERIOD
- TC event is generated

On a decrement event:
- COUNTER is decremented

On a capture event:
- Capture COUNTER to CC0/1
- CC0/1_MATCH event is generated when COUNTER changes to a state in which COUNTER equals CC0/1
- CC0/1 and CC0/1_BUFF are exchanged by a CC0/1_MATCH event



Hint Bar

Review the TCPWM chapter in the TRM for additional details

PWM

- The comparison output is a PWM signal whose period depends on the period register value and whose duty cycle depends on the compare and period register values
- Event generation
  - SW selects the input triggers (Reload[1], Start[2], Stop/Kill[3], Count[4], Capture0[5], Stop1/Kill1[6])
- 16-bit or 32-bit counter
  - Auto reload
    - CC0/1 and CC0/1_BUFF are exchanged on a CC0/1_MATCH event
    - PERIOD and PERIOD_BUFF are exchanged on a switch event and TC event
  - Counter operation
    - One-shot mode and Continuous mode
    - Up/Down modes (COUNT_UP/COUNT_DOWN/COUNT_UPDN1/2)
- Signal output
  - TR_OUT0: OV, UN, TC, CC0_MATCH, CC1_MATCH, LINE_OUT[7]
  - TR_OUT1: OV, UN, TC, CC0_MATCH, CC1_MATCH, LINE_OUT[7]
  - Interrupt: TC, CC0_MATCH, CC1_MATCH

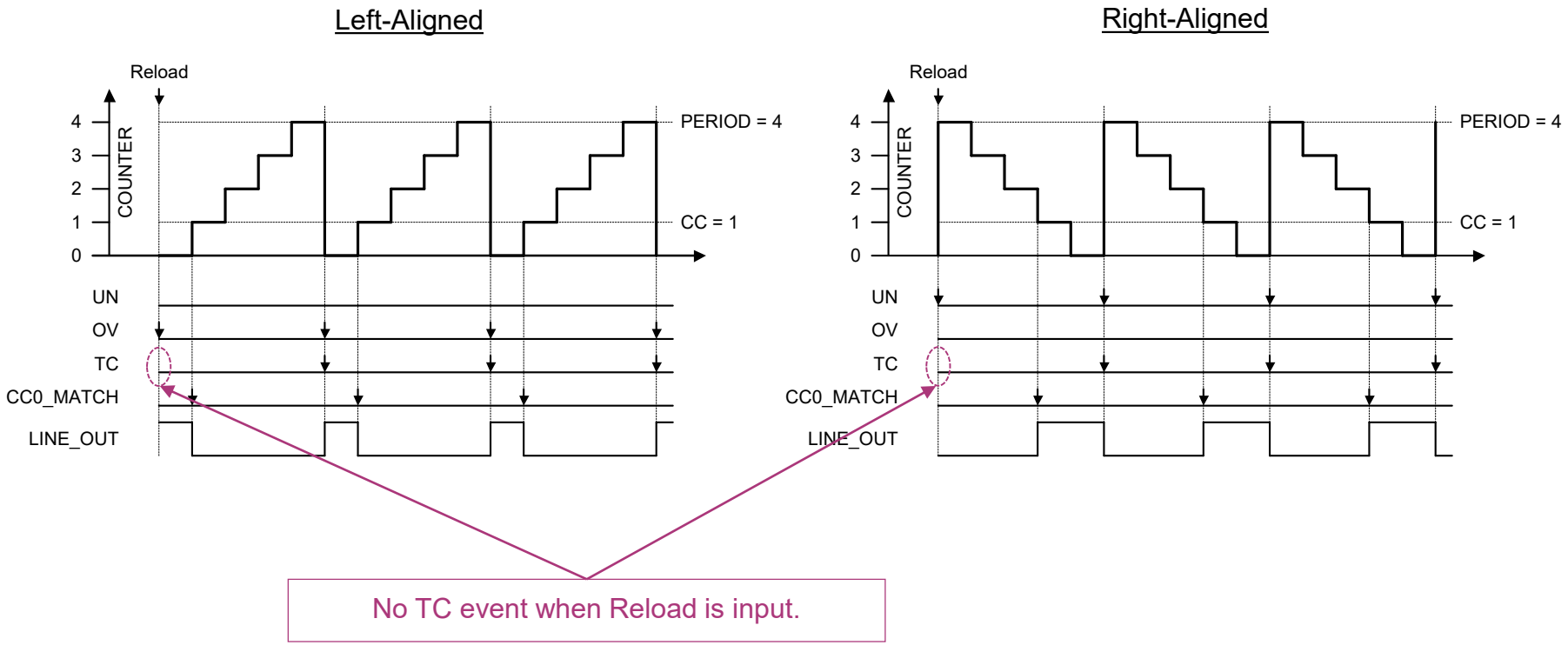
Hint Bar

Review the TCPWM chapter in the TRM for additional details

[1] Reload: Sets the counter value and starts the counter.
[2] Start: Starts the counter.
[3] Stop/kill: Stops the counter. Different stop/kill modes exist.
[4] Count: Count event incr/decr the counter.
[5] Capture0: This event acts as a switch event.
[6] Stop1/kill1: This event acts as a second stop/kill event.
[7] LINE_OUT can be output with inverted polarity.

PWM Operation

› Operation example (Continuous mode, COUNT_UP/DOWN)



Hint Bar

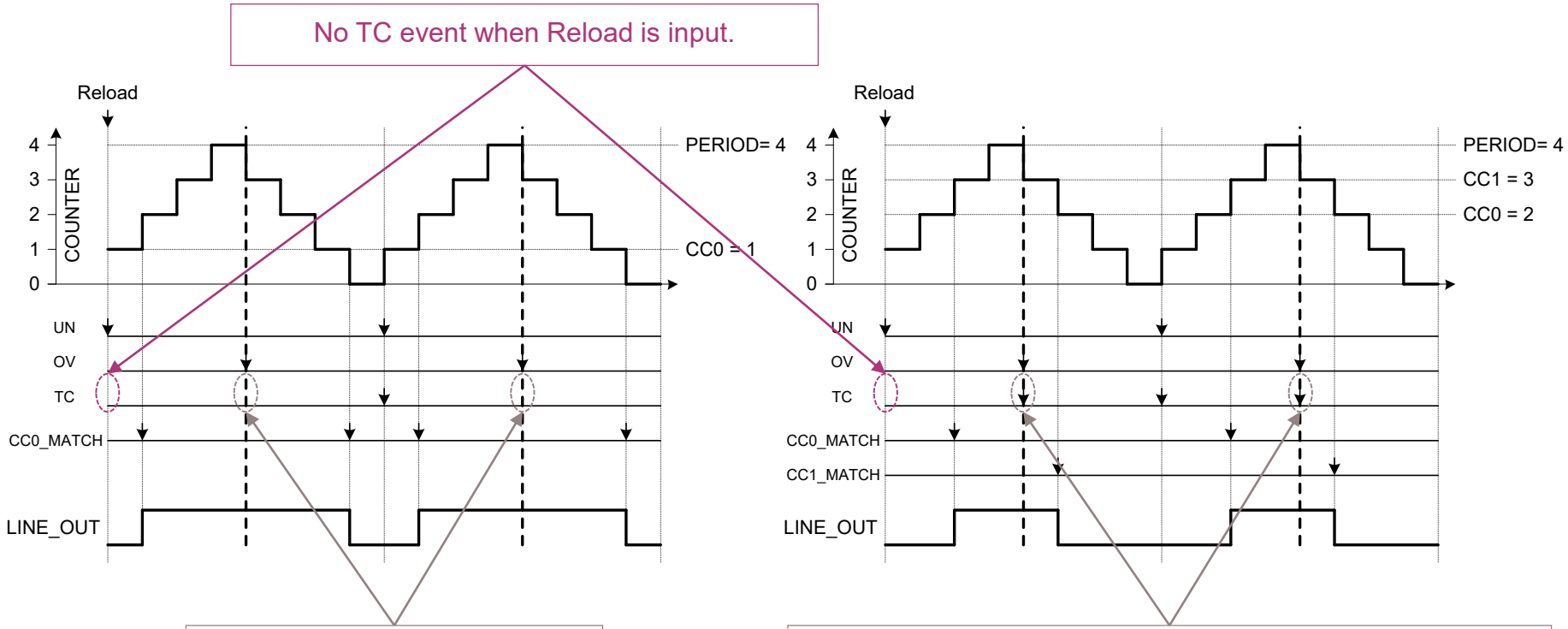
Review the TCPWM chapter in the TRM for additional details

PWM Operation

› Operation example (Continuous mode)

Center Aligned, Symmetric (COUNT_UPDN1)

Center Aligned, Asymmetric (COUNT_UPDN2[1])



No TC event occurs in UPDN1.

TC event occurs in UPDN2.

(This is useful when you want to generate an interrupt for both UN and OV.)

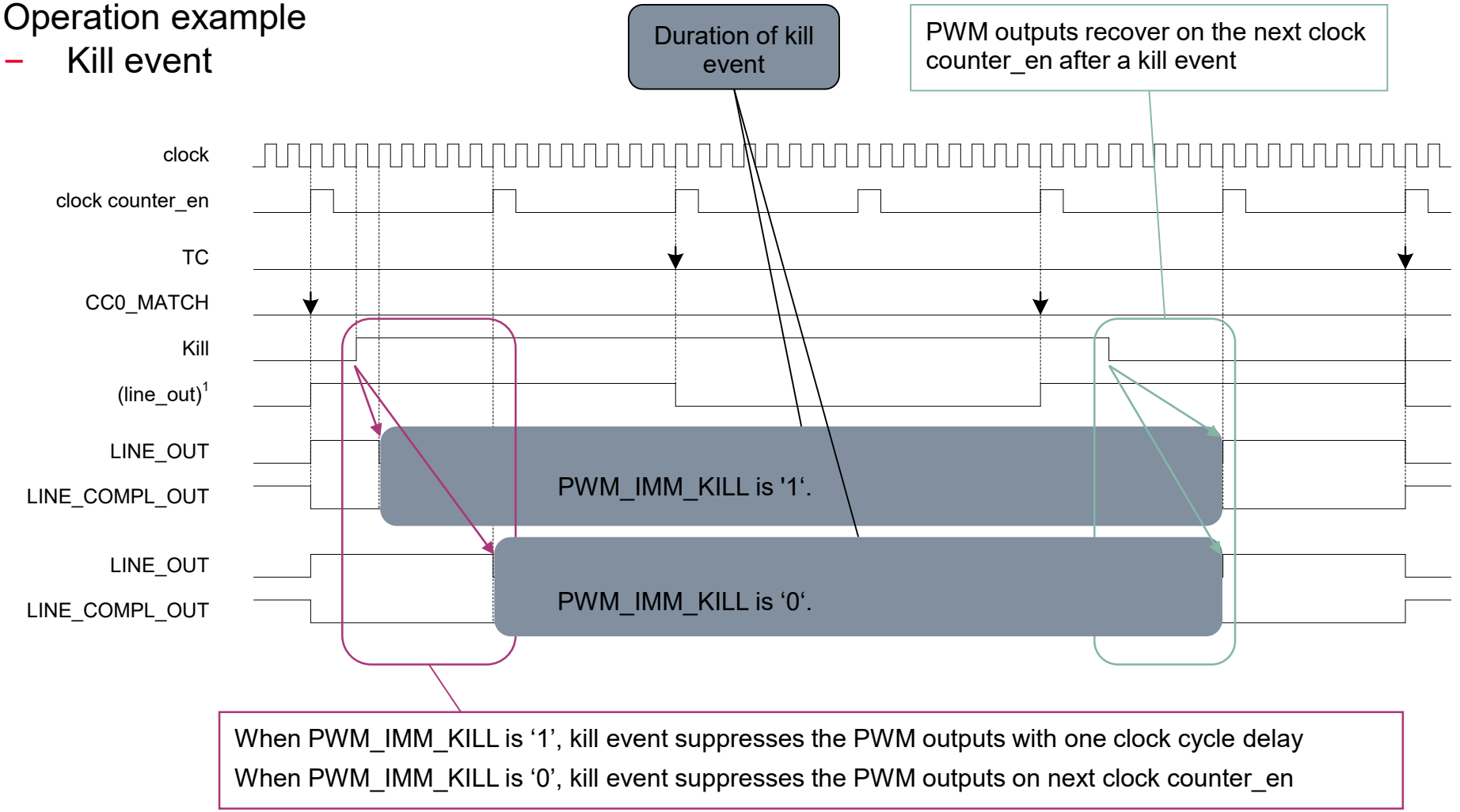
[1] Asymmetric PWM signals can be generated using the COUNT_UPDN2 mode.

Hint Bar

Review the TCPWM chapter in the TRM for additional details

PWM Operation

- Operation example
  - Kill event



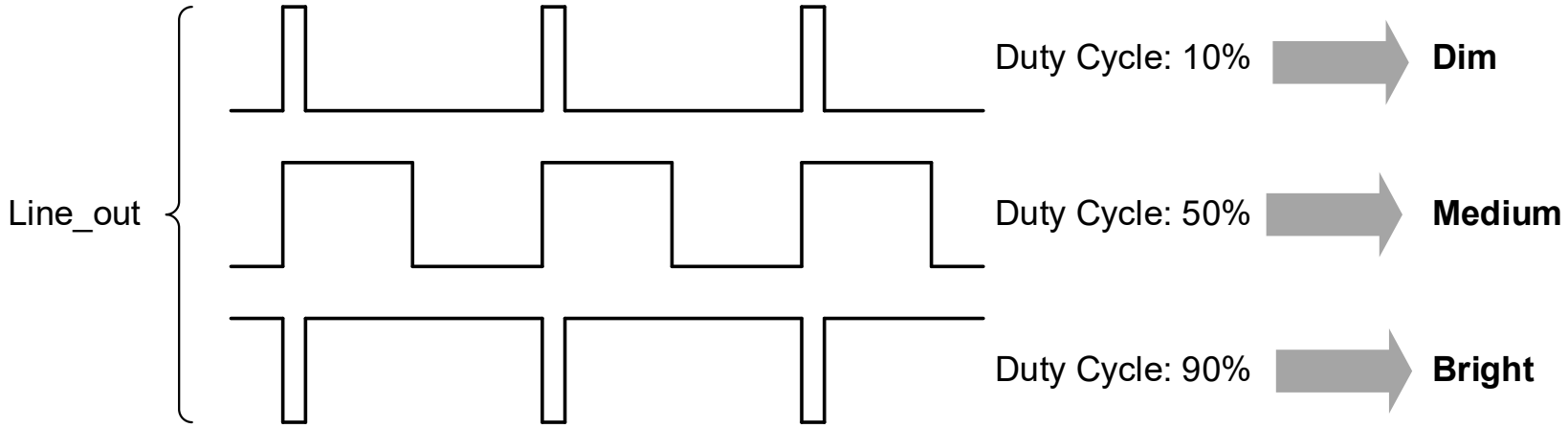
Hint Bar

Review the TCPWM chapter in the TRM for additional details

[1] line_out is the signal when the kill event does not occur.

PWM Use Case

- Control LED brightness
  - Control brightness of the LED with duty cycle of the PWM signal



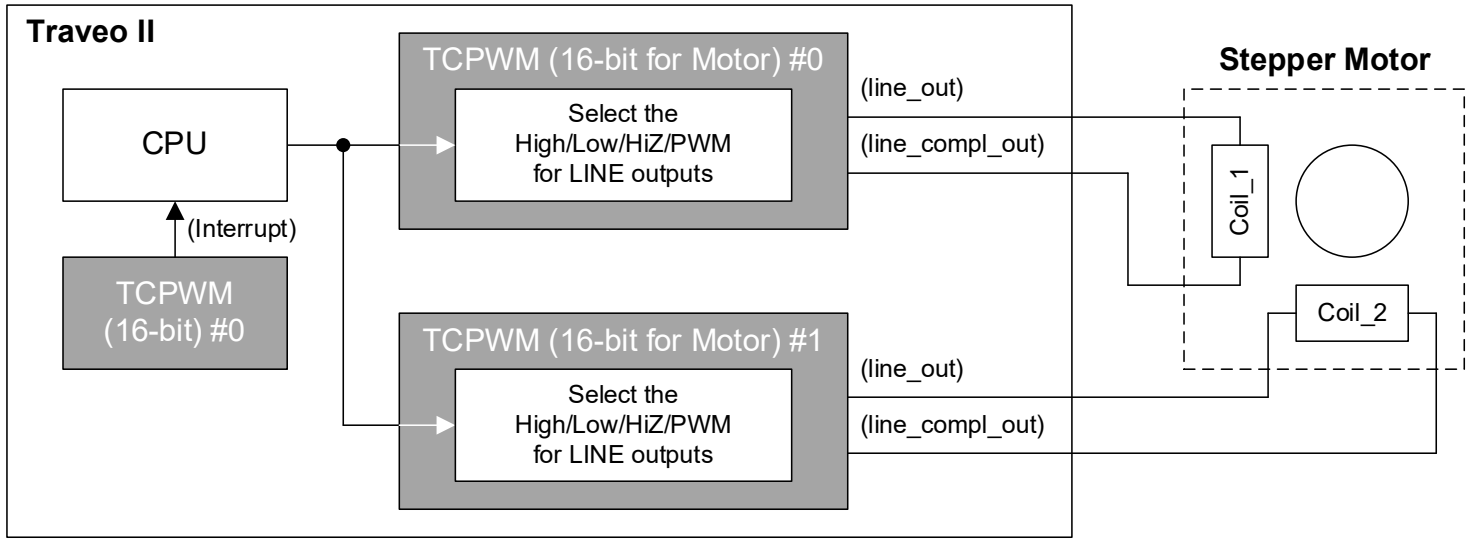
- The line output signal can be output with 0% and 100% duty, except in the following cases:
  - COUNT_UP mode
    To get 100% duty cycle, CC0 must be set to a value higher than the period. If period is 0xFFFF (for 16-bit counter) or 0xFFFFFFFF (for 32-bit counter), 100% duty cycle cannot be achieved.
  - COUNT_DOWN mode
    To get 0% duty cycle, CC0 must be set to a value higher than the period. If period is 0xFFFF (for 16-bit counter) or 0xFFFFFFFF (for 32-bit counter), 0% duty cycle cannot be achieved.

Hint Bar

Review the TCPWM chapter in the TRM for additional details

PWM Use Case

- Stepper Motor Control (SMC)
  - TCPWM supports “PWM Output Selection” for stepper motors, including micro stepper control



  - TCPWM (16 bit) #0: Generates the interrupt signal for motor control period
  - TCPWM (16 bit for Motor) #0/1: Generates the SMC signals
- Advantage
  - TCPWM counter outputs can be connected to GPIO_SMC[1] to support a drive strength of 30 mA for driving a stepper motor directly

Hint Bar

Review the TCPWM chapter in the TRM for additional details

[1] It is only supported by the Cluster device family.

PWM_DT

- PWM_DT functionality is the same as PWM functionality, except that dead time can be inserted and clock pre-scaling is not provided
- Event generation
  - SW selects the input triggers (Reload[1], Start[2], Stop/kill[3], Count[4], Capture0[5], Stop1/kill1[6])
- 16-bit or 32-bit counter
  - Auto reload
    - CC0/1 and CC0/1_BUFF are exchanged on a CC0/1_MATCH event
    - PERIOD and PERIOD_BUFF are exchanged on a switch event and TC event
  - Counter operation
    - One-shot mode and Continuous mode
    - Up/Down modes (COUNT_UP/COUNT_DOWN/COUNT_UPDN1/2)
  - Dead time insertion
    - Range is 0 to 65535 counter clock cycles (16-bit, only for Counter Group 2)
    - Range is 0 to 255 (8-bit)
- Signal output
  - TR_OUT0: OV, UN, TC, CC0_MATCH, CC1_MATCH, LINE_OUT[7]
  - TR_OUT1: OV, UN, TC, CC0_MATCH, CC1_MATCH, LINE_OUT[7]
  - Interrupt: TC, CC0_MATCH, CC1_MATCH

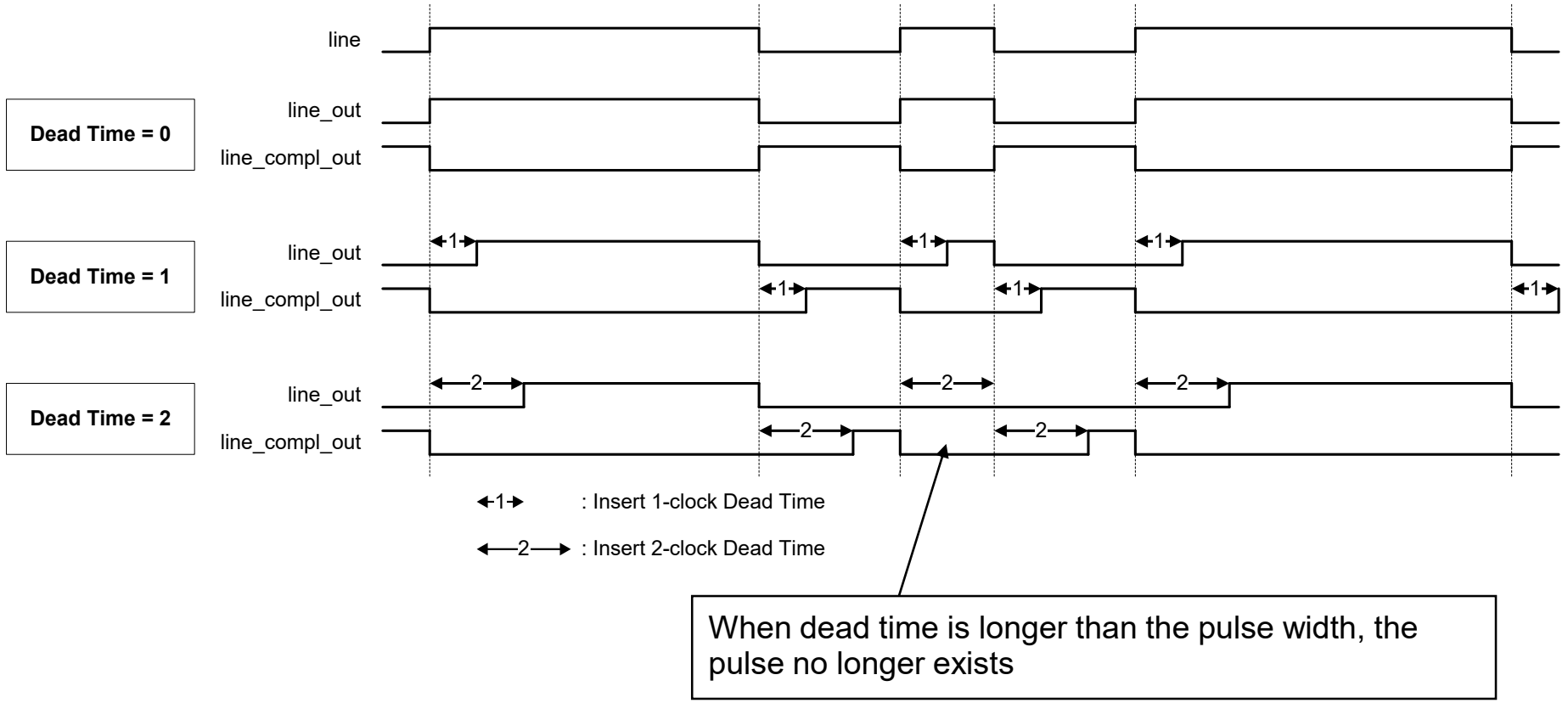
Hint Bar

Review the TCPWM chapter in the TRM for additional details

[1] Reload: Sets the counter value and starts the counter.
[2] Start: Starts the counter.
[3] Stop/kill: Stops the counter. Different stop/kill modes exist.
[4] Count: Count event incr/decr the counter.
[5] Capture0: This event acts as a switch event.
[6] Stop1/kill1: This event acts as a second stop/kill event.
[7] LINE_OUT can be output with inverted polarity.

Dead Time Insertion

- Example (Dead time insertion: 0-clock, 1-clock, 2-clock)
  - To prevent a short circuit due to switching element delay, insert the dead time

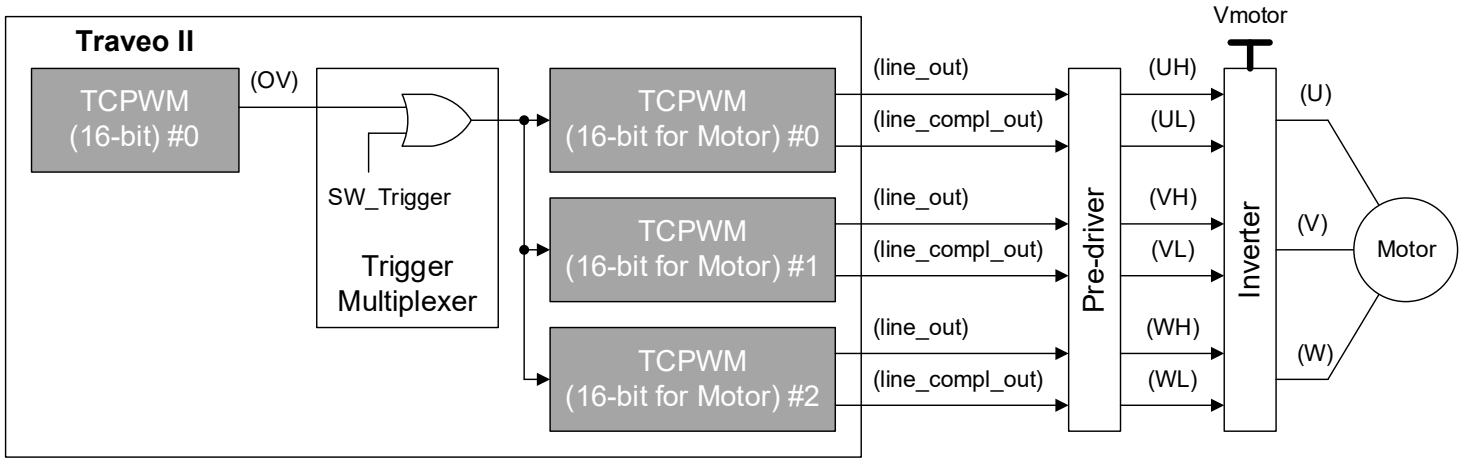


Hint Bar

Review the TCPWM chapter in the TRM for additional details

PWM_DT Use Case

- Three-phase motor control
  - By using three TCPWMs, a three-phase motor can be controlled using a PWM signal with dead time



  - TCPWM (16 bit) #0: Generates the motor control period
  - TCPWM (16 bit for motor) #0/1/2: Generates the motor control signals
- Three TCPWMs can be activated simultaneously using overflow of TCPWM or SW command of trigger multiplexer

Hint Bar

Review the TCPWM chapter in the TRM for additional details

[1] Reload: Sets the counter value and starts the counter.
[2] Start: Starts the counter.
[3] Stop/kill: Stops the counter. Different stop/kill modes exist.
[4] Count: Count event incr/decr the counter.
[5] Capture0: This event acts as a switch event.
[6] Stop1/kill1: This event acts as a second stop/kill event.
[7] LINE_OUT can be output with inverted polarity.

PWM_PR

- The PWM PR functionality changes the counter value using the linear feedback shift register (LFSR)
- Event generation
  - SW selects the input triggers (Reload[1], Start[2], Stop/kill[3], Capture0[4]/1[5])
- 16-bit or 32-bit counter
  - Auto reload
    - CC0/1 and CC0/1_BUFF are exchanged on a CC0/1_MATCH event
    - PERIOD and PERIOD_BUFF are exchanged on a switch event and TC event
    - LINE_SEL and LINE_SEL_BUFF are exchanged on a switch event and TC event
  - Counter operation
    - One-shot mode and Continuous mode
  - Generate a pseudo-random number sequence
  - The generated signal has different frequency/noise characteristics than a regular PWM signal
  - Programmable LFSR length
- Signal output
  - TR_OUT0: OV, UN, TC, CC0_MATCH, CC1_MATCH, LINE_OUT[6]
  - TR_OUT1: OV, UN, TC, CC0_MATCH, CC1_MATCH, LINE_OUT[6]
  - Interrupt: TC, CC0_MATCH, CC1_MATCH

Hint Bar

Review the TCPWM chapter in the TRM for additional details

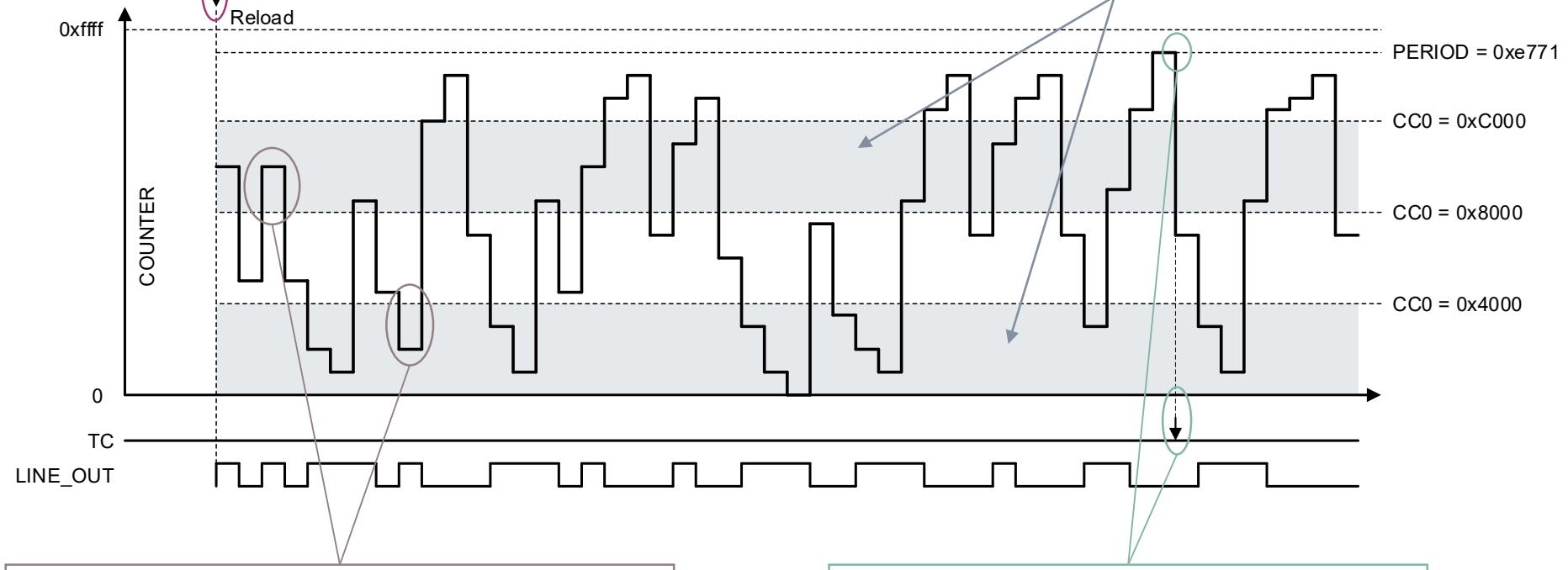
[1] Reload: Same behavior as start event.
[2] Start: Starts the counter.
[3] Stop/kill: Stops the counter. Different stop/kill modes exist
[4] Capture0: Switch event.
[5] Capture1: Second stop/kill event.
[6] LINE_OUT can be output with inverted polarity.

PWM_PR Operation

› Operation example (Continuous mode, generate a pseudo-random number)

Starts the counter. The counter is not initialized by hardware. The current counter value is used.

- The gray areas represent the counter region in which the LINE_OUT value is '1', for a CC0 value of 0x4000.
- Only the lower 15 bits of COUNTER are used for comparison, while the COUNTER can run up to 16-bit values. Therefore, there are two gray areas.



When COUNTER is between 0x0000 and 0x4000 or 0x8000 and 0xC000, the LINE_OUT value is set to '1'.

When COUNTER equals PERIOD, a TC event is generated.

Hint Bar

Review the TCPWM chapter in the TRM for additional details

Shift Register

- Shift Register (SR) functionality shifts the counter value to the right
- Event generation
  - SW selects the input triggers (Reload[1], Start[2], Stop[3], Shift[4], Serial-in[5])
- 16-bit or 32-bit counter
  - Auto reload
    - CC0/1 and CC0/1_BUFF are exchanged on a CC0/1_MATCH event
  - Counter shift
    - The counter value is shifted to the right
- Signal output
  - TR_OUT0: CC0_MATCH, CC1_MATCH, LINE_OUT[6]
  - TR_OUT1: CC0_MATCH, CC1_MATCH, LINE_OUT[6]
  - Interrupt: CC0_MATCH, CC1_MATCH

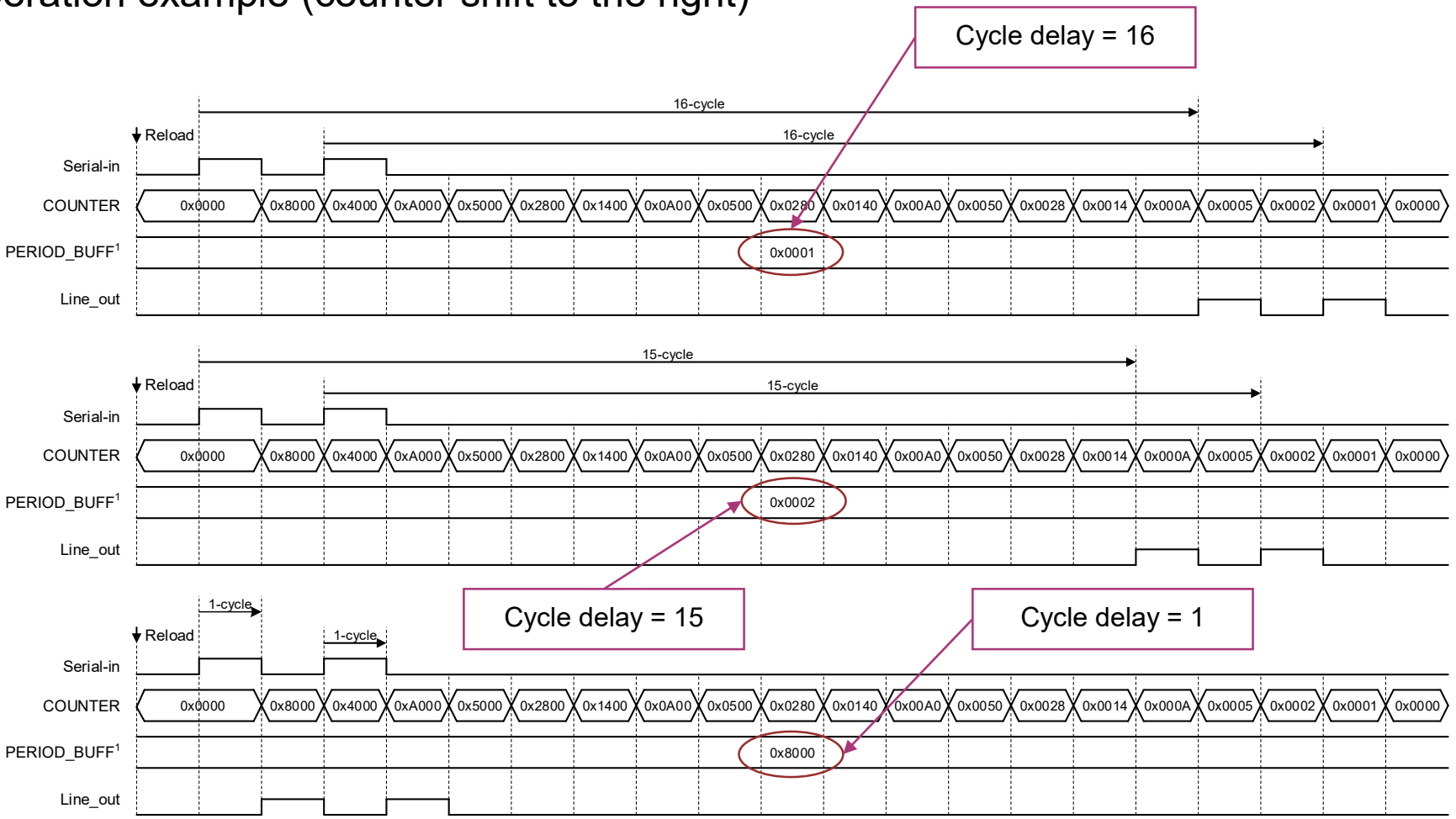
Hint Bar

Review the TCPWM chapter in the TRM for additional details

[1] Reload: Sets the counter value to “0” and starts the counter shift operation.
[2] Start: Starts the counter shift operation.
[3] Stop: Stops the counter.
[4] Shift: Shifts the counter in right direction.
[5] Serial-in: Serial input to the MSB of the counter.
[6] LINE_OUT can be output with inverted polarity.

SR Operation

› Operation example (counter shift to the right)



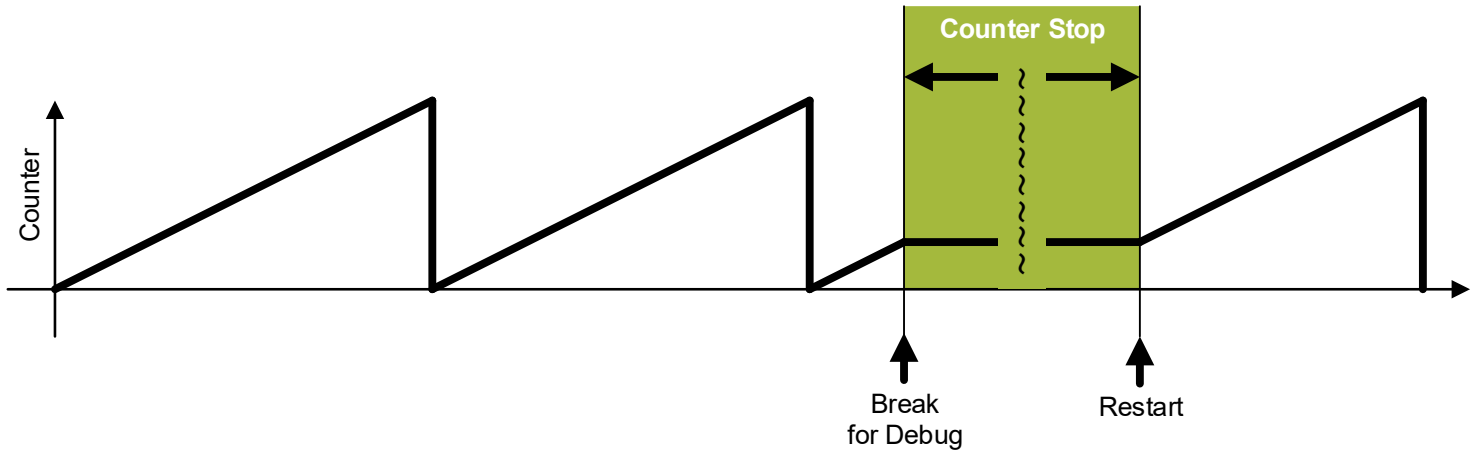
Hint Bar

Review the TCPWM chapter in the TRM for additional details

[1] The set value of the PERIOD_BUFF refers to the delay cycle number of the shift register. The period buffer needs to be set so that only one of the delay taps is valid.

Debug Mode

- TCPWM counters support Debug mode
  - Can be configured per counter if counter operation continues or pauses in debug state
- Use Case
  - Counter operation is paused



- Advantage
  - Even if you break for debugging, you can debug with the same counter operation as the actual operation

Hint Bar

Review the TCPWM chapter in the TRM for additional details

Part of your life. Part of tomorrow.

Revision History

Revision	ECN	Submission Date	Description of Change
**	6151675	04/25/2018	Initial release
*A	6397021	10/24/2018	Added page 2. Updated pages 3, 4, 5, and 8.
*B	6715800	10/28/2019	Added pages 5, 8, and 41. Updated pages 2, 3, 4, 9, 10, 13, 14, 16, 17, 38, 40, 43, and 44.
*C	6822257	02/10/2019	Updated figures on pages 37 and 38. Updated Hint Bar on pages 6 to 49.
*D	7082696	02/01/2021	Updated pages 2, 3, 4, 5. Convert content to IFX format