TIMER DESIGN

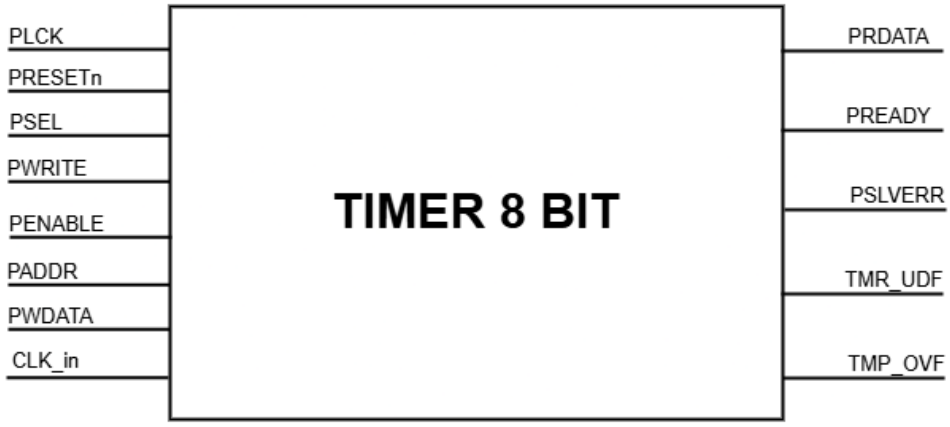
Full name: Nguyen Manh Loc

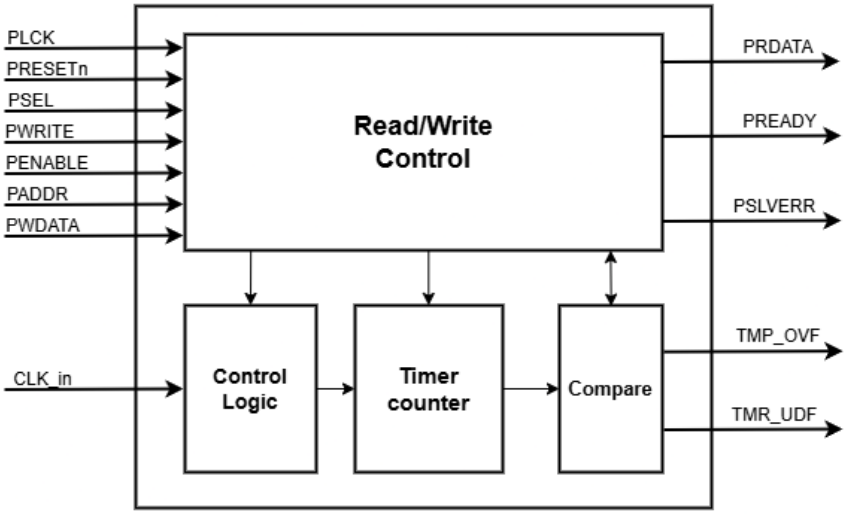
Gmail: manhloc272@gmail.com

1. **Overview**
    a. **Feature**

This LSI has an on-chip 8-bit timer module with two channels operating on the basis of an 8-bit counter. The 8-bit timer module can be used to count external events and be used as a multifunction timer in a variety of applications, such as generation of counter reset, interrupt requests, and pulse output with an arbitrary duty cycle using a compare-match signal with two registers.

b. Block diagram





c. Input/Outputs pin (LSI pin – CHIP PIN/PORT)

PORT name	Attribute	BIT WIDTH	Description
PCLK	Input	1	Clock. The rising edge of PCLK times all transfers on the APB.
PRESETn	Input	1	Reset. The APB reset signal is active LOW. This signal is usually connected directly to the system bus reset signal.
PSEL	Input	1	Select. The APB bridge unit generates this signal for each peripheral bus slave. It indicates that the slave device has been selected and needs to transmit data. There is one PSELx signal for each slave.
PWRITE	Input	1	Direction. This signal indicates an APB write access when HIGH and an APB read access when LOW.
PENABLE	Input	1	Enable. This signal indicates the second and subsequent cycles of an APB transfer.
PADDR	Input	8	Address. This is the APB address bus. It can be up to 32 bits wide and is driven by the peripheral bus bridge unit.
PWDATA	Input	8	Write data. This bus is driven by the peripheral bus bridge unit during write cycles when PWRITE is HIGH.
CLK_inX	Input	4	Clock. The rising edge of CLK_inX.
PRDATA	Output	8	Read Data. The selected slave drives this bus during read cycles when PWRITE is LOW.
PREADY	Output	1	Ready. The slave uses this signal to extend an APB transfer.
PSLVERR	Output	1	This signal indicates a transfer failure.
TMR_UDF	Output	1	Provided timer counter status when counter count from FF to 00.
TMR_OVF	Output	1	Provided timer counter status when counter count from 00 to FF.

d. Register specification

TDR

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
tdr[7]	tdr[6]	tdr[5]	tdr[4]	tdr[3]	tdr[2]	tdr[1]	tdr[0]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit name	Attribute	Description
tdr[7:0]	R/W	8 bits register for setting start value counter for TCNT 00: value 0 in decimal 01: value 1 in decimal … FF: value 255 in decimal

TCR

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
tcr[7]	tcr[6]	tcr[5]	tcr[4]	tcr[3]	tcr[2]	tcr[1]	tcr[0]
Load	Reserved	Up/Dw	En	Reserved	Reserved	cks1	cks0

Bit name	Attribute	Description
tcr[7]	R/W	Manual load data from TDR to TCNT when it active High. 1: load data to TCNT 0: Normal operation.
tcr[6]	Reserved	Reserved
tcr[5]	R/W	Control counter up or counter down 0: counter up 1: counter down
tcr[4]	R/W	0: disable 1: enable
tcr[3]	Reserved	Reserved
tcr[2]	Reserved	Reserved
tcr[1:0]	R/W	Select internal clocks for circuit 00: T*2 01: T*4 10: T*8 11: T*16

TSR

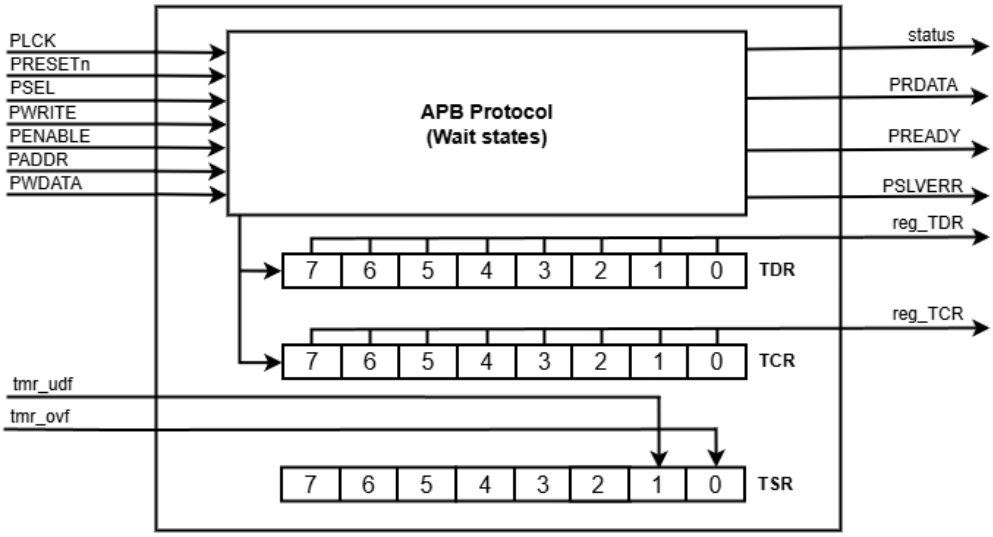
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
tsr[7]	tsr[6]	tsr[5]	tsr[4]	tsr[3]	tsr[2]	tsr[1]	tsr[0]
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Dw	Up

Bit name	Attribute	Description
tsr[7:2]	R	Reserved
tsr[1]	R/W*	Timer counter underflow when counter 8’h00 down to 8’hff:

		This bit is only set by hardware, clear by software
tsr[0]	R/W*	Timer counter overflow when counter 8'hFF to 8'h00: This bit is only set by hardware, clear by software

2. Read/write register control

a. Input/Outputs pin



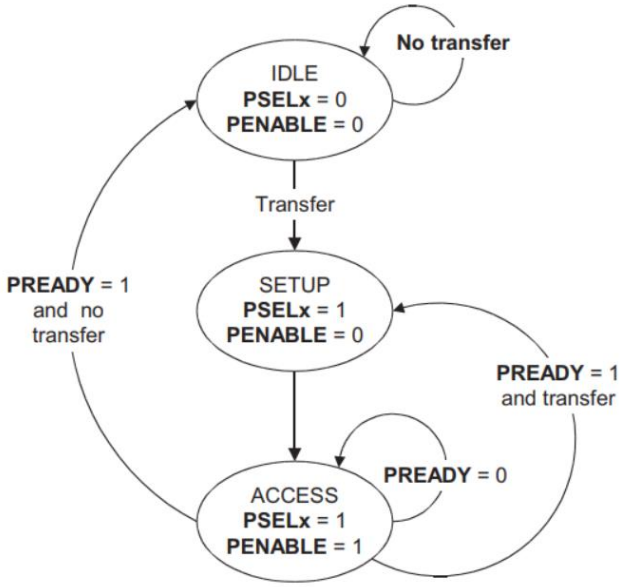
PORT name	Attribute	BIT WIDTH	Description
status	Output	2	Controls abnormal changes of counter overflow or underflow.

b. Functional/Protocol

Function: As APB slave, allows CPU to read/write:

- TCR (Control): Configure, control timer.
- TDR (Data): Load count/compare value, read current value.
- TSR (Status): Read status, clear interrupt flag.

c. State machine control for APB Protocol



The sate machine operates though the following states:

IDLE: This is the default state of the APB.

SETUP: When a transfer is required the bus moves into the SETUP state, where the appropriate select signal, PSELx is asserted. The bus only remains in the SETUP state for one clock cycle and always moves to the ACCESS state on the next rising edge of the clock.

ACCESS: The enable signal, PENABLE, is asserted in the ACCESS state. The address, write, select, and write data signals must remain stable during the transition from the SETUP to ACCESS state.

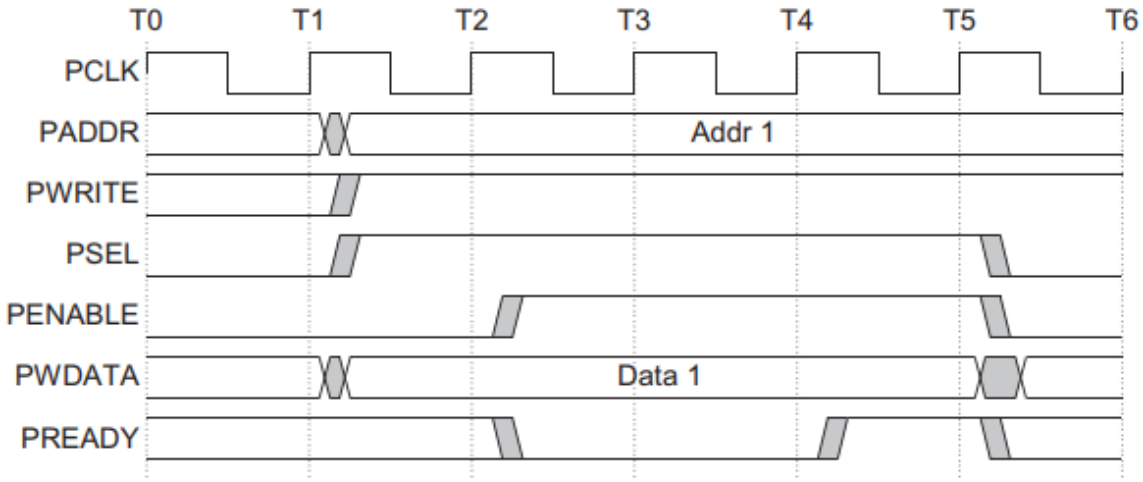
Exit from ACCESS state is controlled by the PREADY signal from the slave:

- If PREADY is held LOW by the slave then the peripheral bus remain in the ACCESS state.

- If PREADY is driven HIGH by the slave then the ACCESS state is exited and the bus returns to the IDLE state if no more transfers are required. Alternatively, the bus moves directly to the SETUP state if another transfer follows.

d. Timing chart

Write transaction wait state



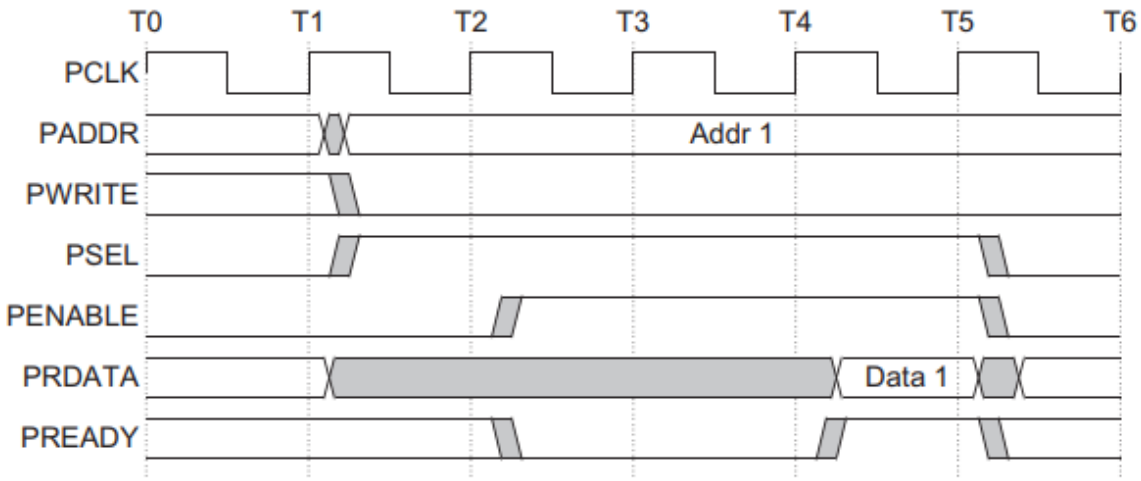
The write transfer starts with the address, write data, write signal and select signal all changing after the rising edge of the clock. The first clock cycle of the transfer is called the Setup phase. After the following clock edge the enable signal is asserted, PENABLE, and this indicates that the Access phase is taking place. The address, data and control signals all remain valid throughout the Access phase. PREADY signal from the slave can extend the transfer. PREADY signal from the slave can extend the transfer. During an Access phase, when PENABLE is HIGH, the transfer can be extended by driving PREADY LOW. The following signals remain unchanged for the additional cycles:

- address, PADDR
- write signal, PWRITE
- select signal, PSEL
- enable signal, PENABLE
- write data, PWDATA.

PREADY can take any value when PENABLE is LOW. This ensures that peripherals that have a fixed two cycle access can tie PREADY HIGH. The transfer completes at the end of this cycle.

The enable signal, PENABLE, is deasserted at the end of the transfer. The select signal, PSELx, also goes LOW unless the transfer is to be followed immediately by another transfer to the same peripheral.

Read transaction wait state

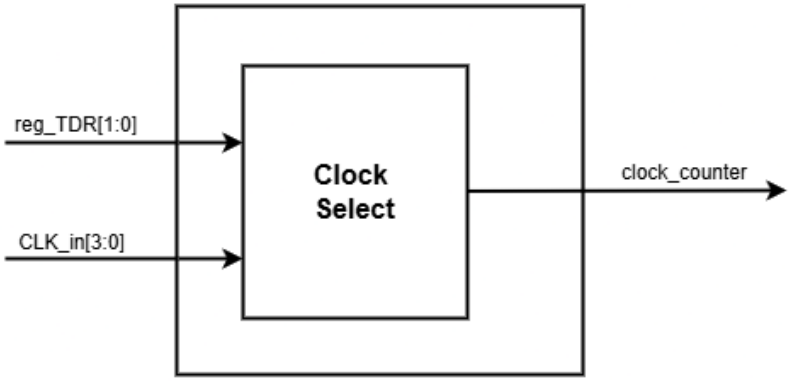


The timing of the address, write, select, and enable signals are as described in Write transfers. The slave must provide the data before the end of the read transfer. The PREADY signal can extend the transfer. The transfer is extended if PREADY is driven LOW during an Access phase. The protocol ensures that the following remain unchanged for the additional cycles:

- address, PADDR
- write signal, PWRITE
- select signal, PSEL
- enable signal, PENABLE.

Figure shows that two cycles are added using the PREADY signal. However, you can add any number of additional cycles, from zero upwards.

3. **Control logic**
    a. **Input/Outputs pin**

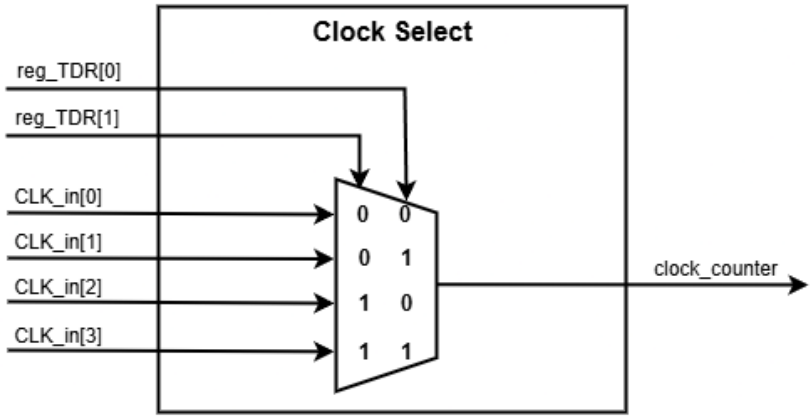


PORT name	Attribute	BIT WIDTH	Description
clock_couter	Output	1	Select internal clocks for circuit 00: T*2 01: T*4 10: T*8 11: T*16

b. Functional/Protocol

This control logic functions as a multiplexer (MUX). It receives various clock sources as input and, based on control signals (typically from the timer's configuration register), it selects one of these clock sources to output as the clock_counter signal. This clock_counter signal then serves as the main clock pulse to drive the counting operation of the 8-bit counter within the timer.

c. Design circuit

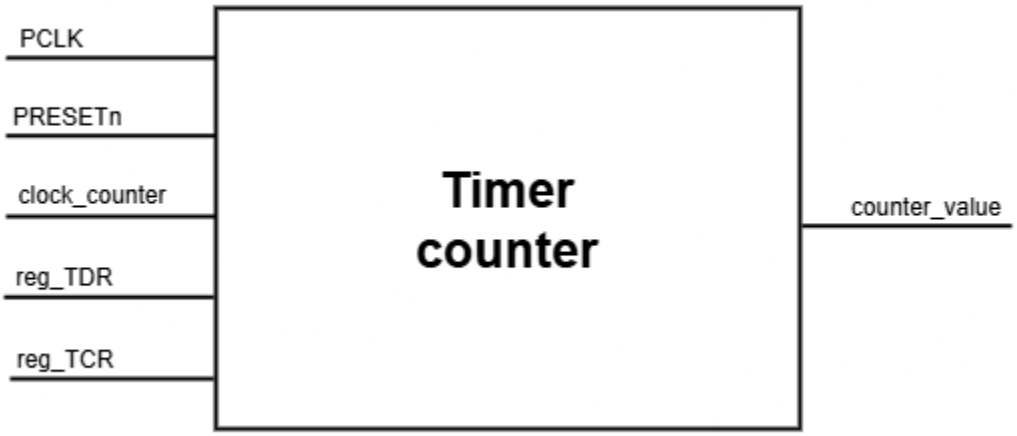


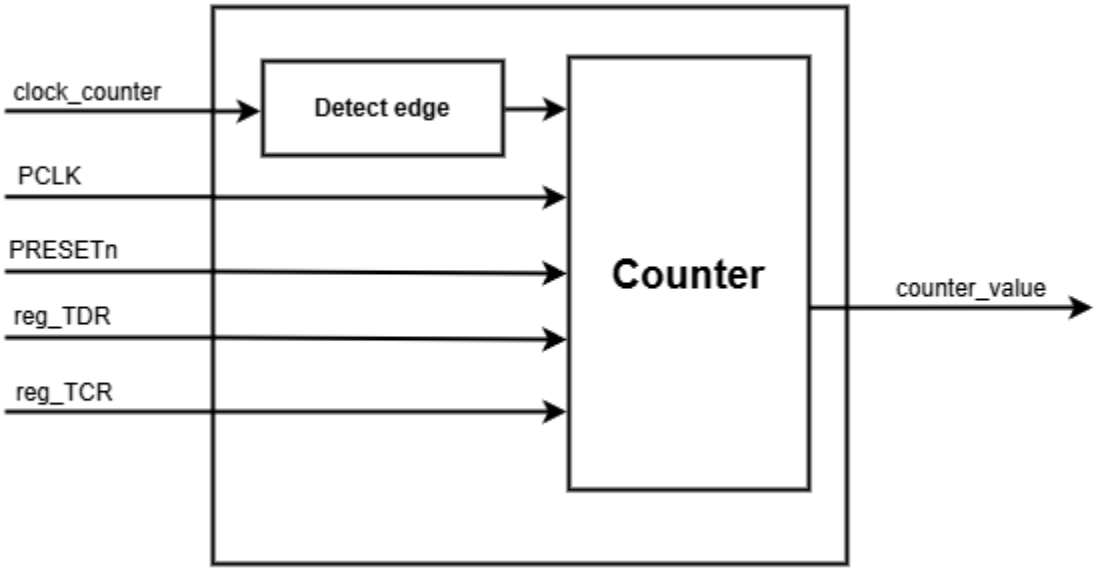
d. Timing chart

No need due to all circuit using inside control logics are combination logic: These logics will be update value immediately without clock signal.

4. Timer counter (TCNT)

a. Input/Outputs pin





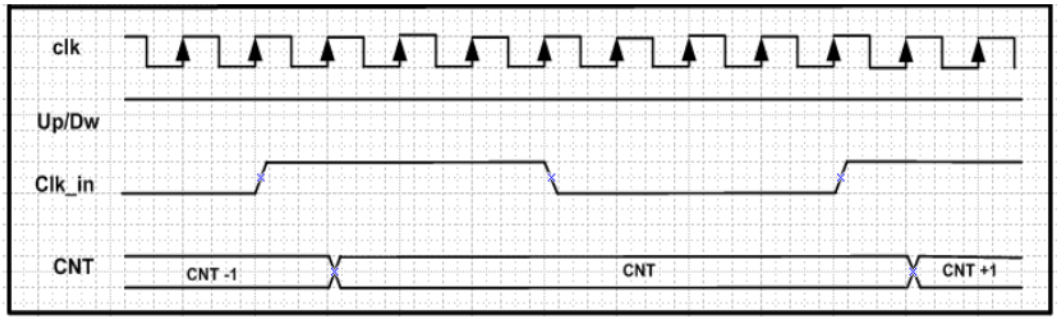
PORT name	Attribute	BIT WIDTH	Description
Counter_value	Output	8	The current count result

b. Functional/Protocol

The timer counter module operates based on the clock_counter signal, which is passed through an edge detection block to generate count pulses. When the PRESETn signal is low, the counter resets to 0. The preset value from reg_TDR is loaded into the counter when a load condition is triggered (as defined in reg_TCR). The counter performs up or down counting depending on the control bit in reg_TCR. The current count result is output through counter_value.

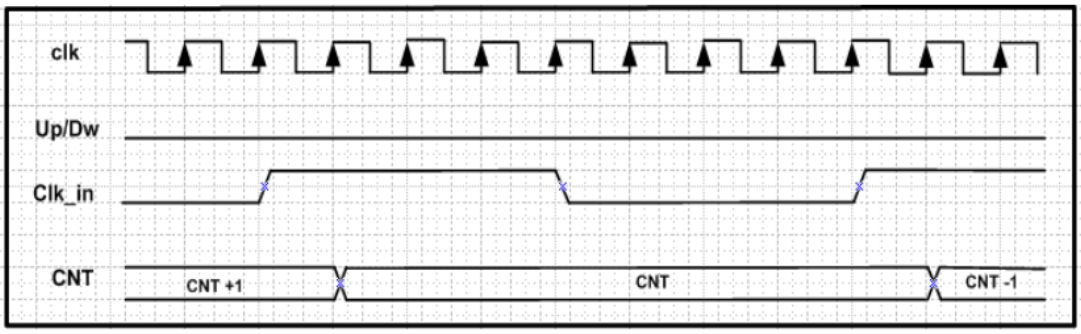
c. Timing chart

Counter up:



The diagram illustrates the operation of an up counter, in which the Clk_in signal acts as a trigger but is synchronized with the system clock PCLK (clk). As a result, each change in the CNT value occurs one clock cycle after a rising edge is detected on Clk_in.

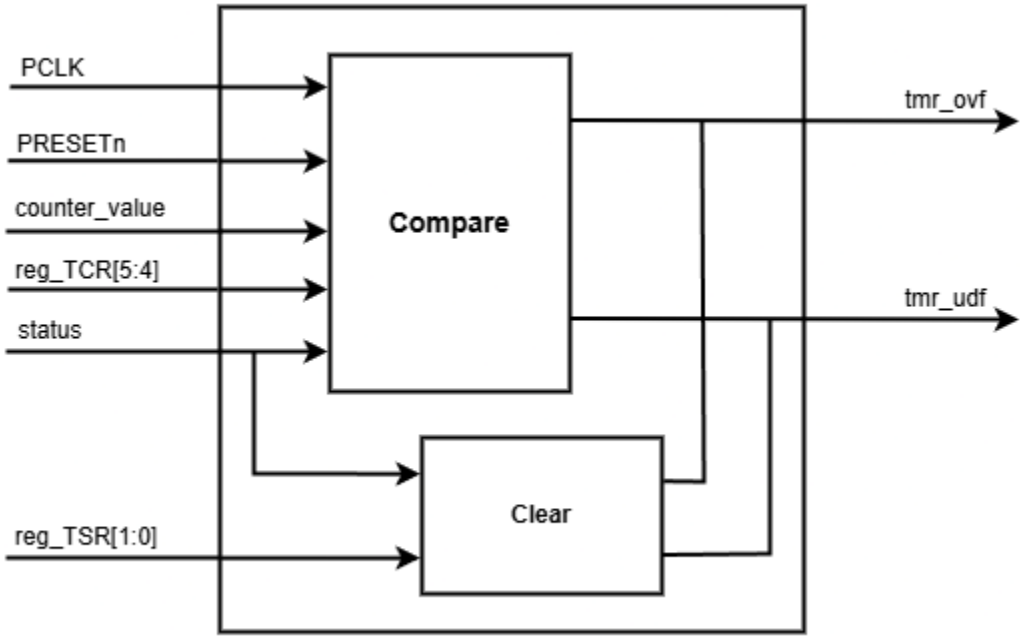
Counter down:



The diagram illustrates the operation of an down counter, in which the Clk_in signal acts as a trigger but is synchronized with the system clock PCLK (clk). As a result, each change in the CNT value occurs one clock cycle after a rising edge is detected on Clk_in.

5. Overflow/Underflow comparison

a. Input/Outputs pin



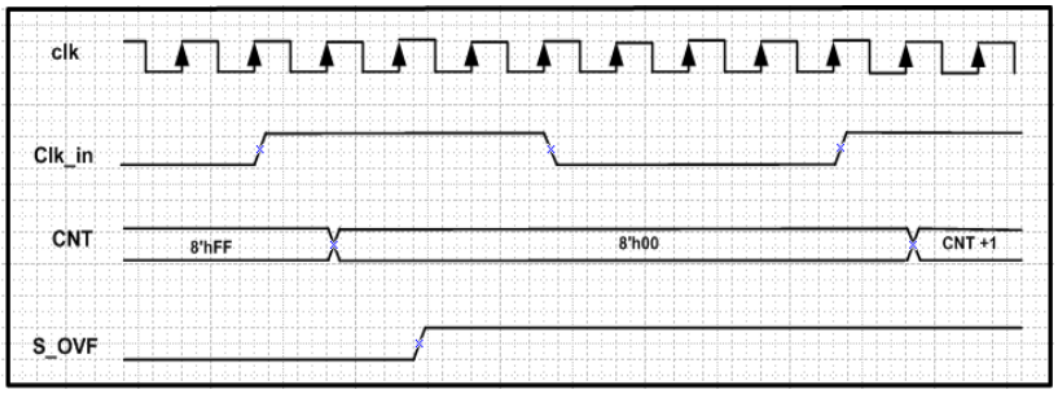
b. Functional/Protocol

The Compare module in the diagram performs the function of monitoring the counter status to detect overflow conditions. Specifically, this module uses the input signals including: the clock signal PCLK, the asynchronous reset signal PRESETn, the current counter value counter_value, the control bits reg_TCR[5:4], and status. Based on these signals, the Compare module checks for overflow or underflow conditions. When an overflow is detected, the module sets the tmr_ovf or tmr_udf signal to a high logic level (1) to indicate the occurrence of the event.

In addition, the module supports a mechanism to clear the overflow flags via the Clear block. Specifically, the user can write to the bits reg_TSR[1:0] to clear the corresponding overflow flags from the input. The Clear block receives this signal and clears the status flags in the Compare module if requested.

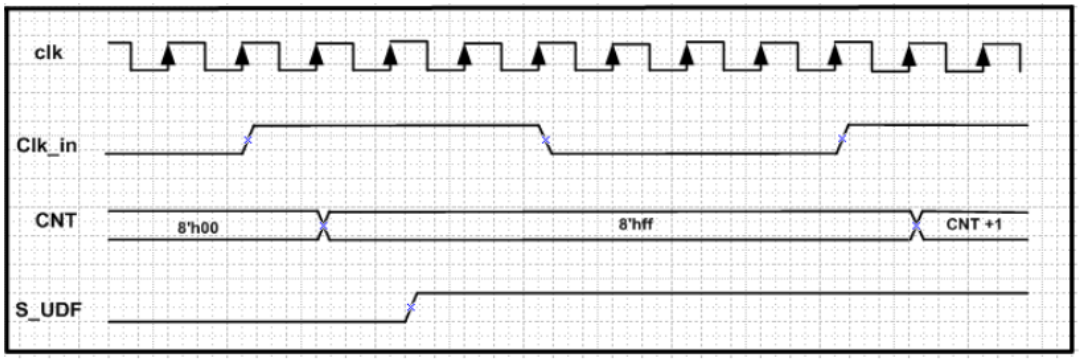
c. Timing chart

Overflow:



The timing diagram illustrates the operation of a counter (CNT) and an overflow signal (S_OVF). Initially, the CNT holds the value 8'hFF. When the Clk_in signal is high, the CNT continues to count. When the CNT reaches its maximum value of 8'hFF and wraps around to 8'h00 on a rising edge of the clock (clk), an overflow event occurs. It's crucial to note that the S_OVF overflow signal is not activated immediately. Instead, S_OVF asserts high one clock cycle (PCLK/clk) after the CNT actually overflows. Subsequently, the CNT continues counting from 8'h00, and S_OVF remains high for a certain period.

Underflow:



The timing diagram illustrates the operation of a counter (CNT) and an underflow signal (S_UDF). Initially, the CNT holds the value 8'h00. When Clk_in is high, the CNT counts down. When the CNT reaches 8'h00 and "wraps around" to 8'hFF on a rising edge of the clock (clk), an underflow event occurs. The S_UDF underflow signal is not activated

immediately; instead, it is delayed by one clock cycle after the CNT's underflow event actually takes place. Subsequently, the CNT continues counting from 8'hFF, and S_UDF remains high.