ECE3849
D-Term 2021

Real Time Embedded Systems

Module 4 Part 4

Module 4 Part 4 Overview

- Timer I/O: Edge-time Capture mode
  - Measuring period
  - Measuring pulse width
  - Analog comparator peripheral
- Timer I/O: Compare mode / PWM output
- PWM Module

Timer I/O

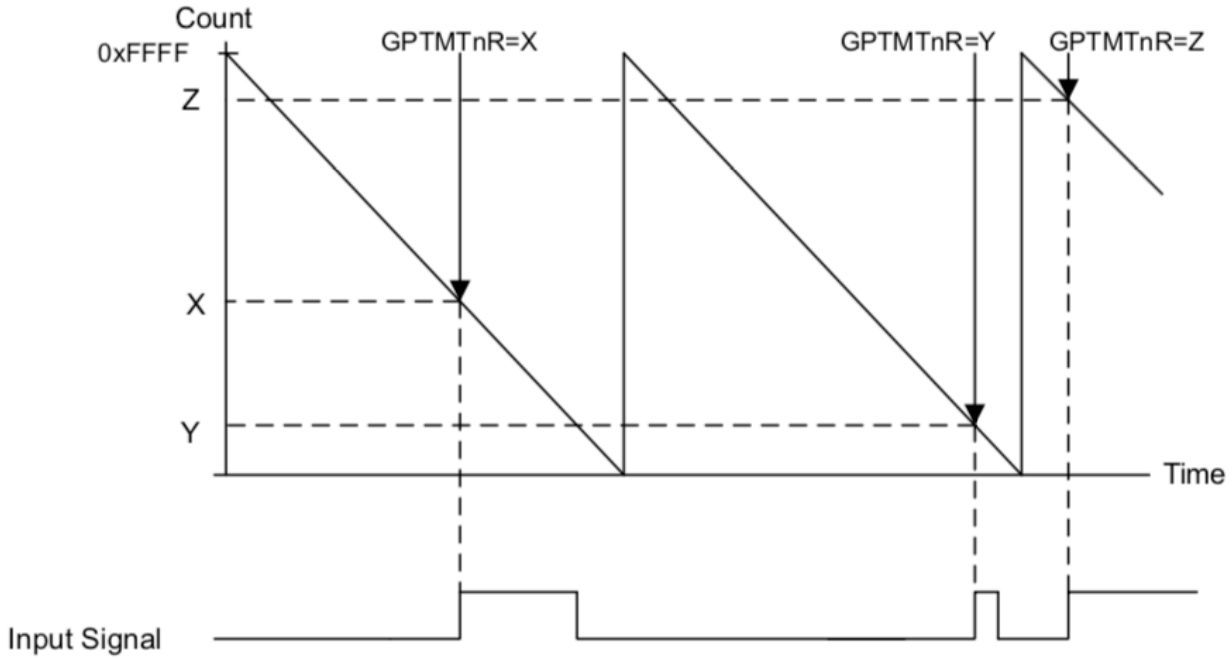
- Timer Modes
  - One-shot mode
    - Interrupts once after a certain amount of time.
    - Example: measuring CPU load.
  - Periodic mode
    - Interrupts on a regular interval to start periodic tasks.
    - Example: sampling button state.
  - Input: Capture (edge timing) mode
    - Starts the timer running and captures the value of a running timer on an edge of a digital input signal.
      - Good for measuring delays, pulse widths, periods.
    - An interrupts occurs when event is captured.
  - Output: Compare mode / PWM mode
    - Counter runs, when the count equals the match count, a GPIO output can be forced high or low.
    - Produces a pulse width modulated (PWM) output, programmed by the match count value.

Configuring Capture Mode

- Configure a timer in edge-time capture mode
  - The timer is 24-bits in edge-time capture mode.
    - 8 MSBs, Bits[23:16] stored in the Prescaler register.
    - 16 LSBs, Bits[15:0] stored in the load register.
- Specify the timer period.
  - Timer period = timer load value * clock period.
  - Use the maximum period allowed by hardware to maximize capture window time.
    - Load value = 0xFFFF.
    - Prescaler value = 0xFF.
- Specify which edge triggers a capture event.
  - Options are rising edge, falling edge, or both.
- Enable capture interrupt.
- Configure the Capture Compare PWM (CCP) input pin.
  - In Capture mode this pin is the digital input being monitored.
  - It is configured in peripheral mode instead of GPIO mode.

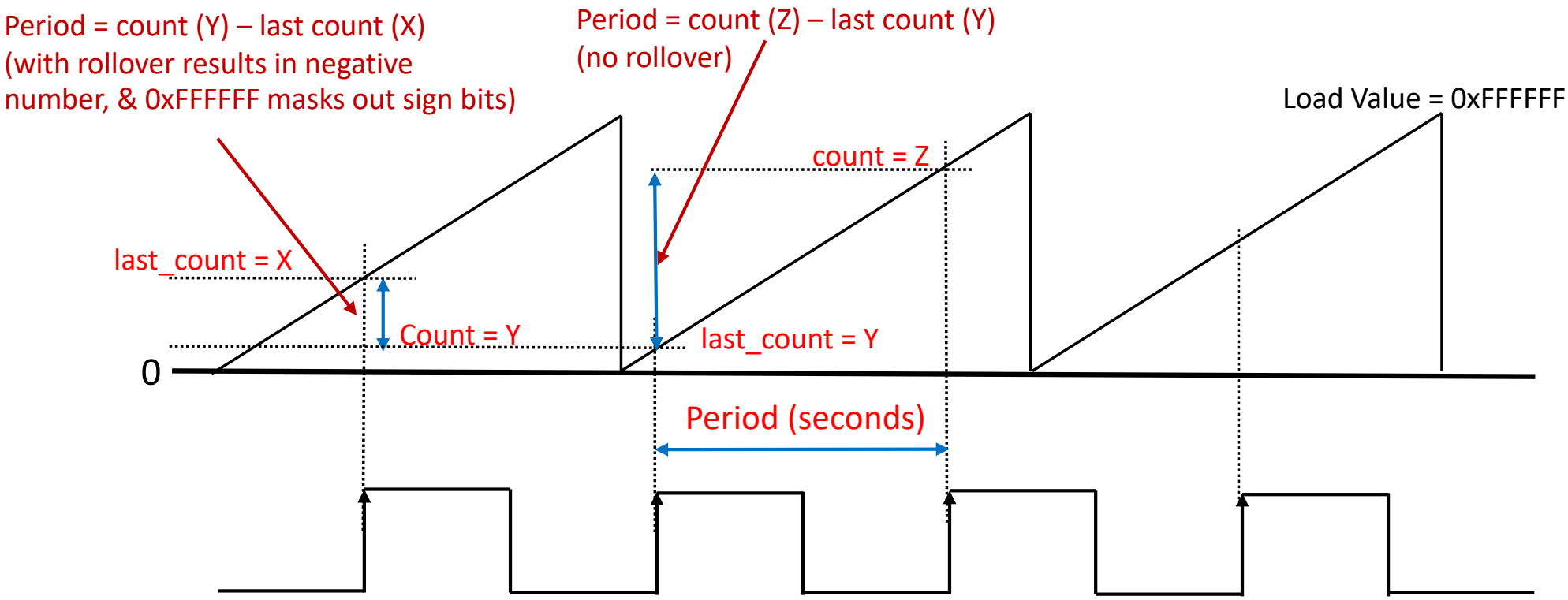
Capture Mode Operation

- In capture mode the timer is continually counting.
  - It can count up or down depending on configuration.
- When the specified event occurs,
  - The event timer value is stored in the GPTMTnPS and GPTMTnR registers.
  - An Hwi / ISR is called.
  - The counter continues counting and rolling over automatically but the event timer value is held.
  - On each event, the GPTMTnPS and GPTMTnR registers are overwritten.



Timer: Measuring Period

- To measure period, set the event trigger to rising edge.
  - The period will be the current event count (count) minus the previous event count (last_count) times the system clock period.
    - Period (counts) = count – last_count.
    - Period (seconds) @ 120 MHz system clock = (count-last_count)* (1/120MHz).
- When, the counter rolls over the period (counts) will be a negative values.
  - To handle counter rollover for a count of 2^N-1, simply & with 2^N-1.
  - Example for the 24-bit value, period (counts) = (count – last_count) & 0xFFFFFF.



Calculating the Period of an Input Signal

```
uint32_t period = 0, last_count = 0;
void TimerCaptureISR(void) {
    <clear timer capture interrupt flag>;
    uint32_t count = <read captured timer count>;
    period = (count - last_count) & 0xffffff;
    last_count = count;
}
```

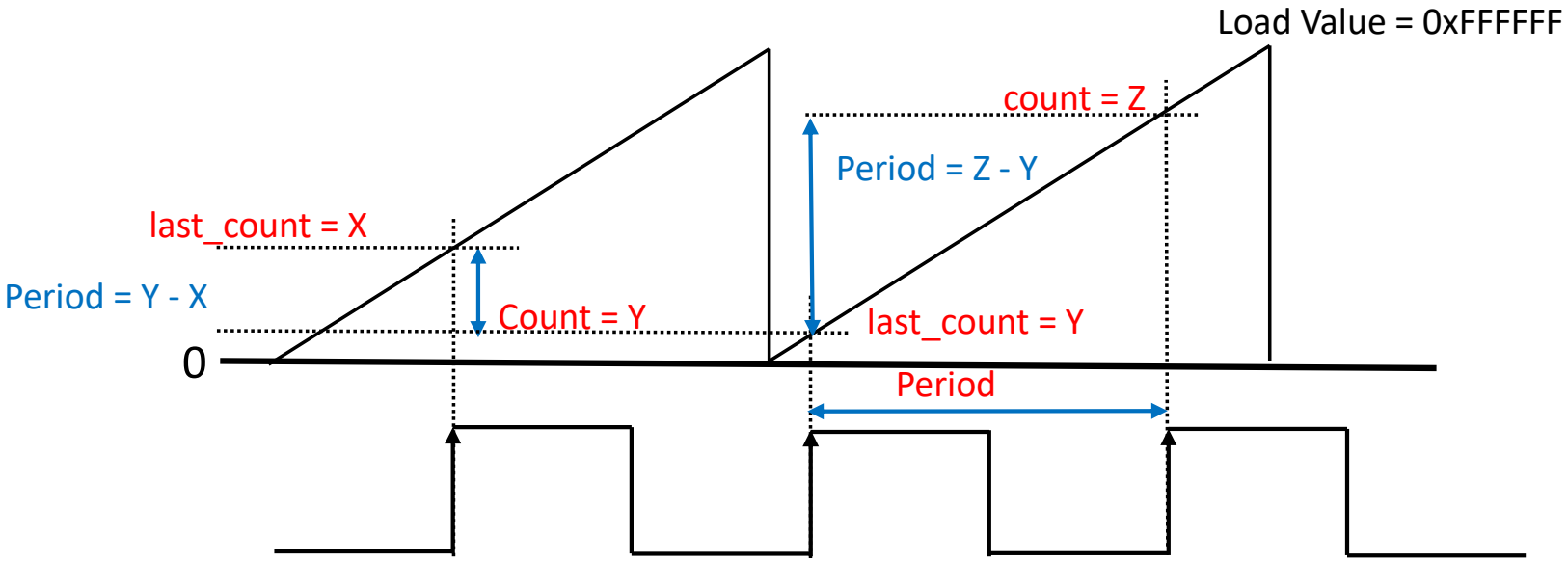
Initialize the period and last_count.

Count = last rising edge event count.
TimerValueGet() returns the 24-bit value.

Period = count – last count.
& 0xFFFFFF corrects for the negative value when roll over occurs.
3-bit example of rollover :
0001 (count)
-0011 (last_count)

1110 & 0111 = 110;

Store count in last_count for next measurement.

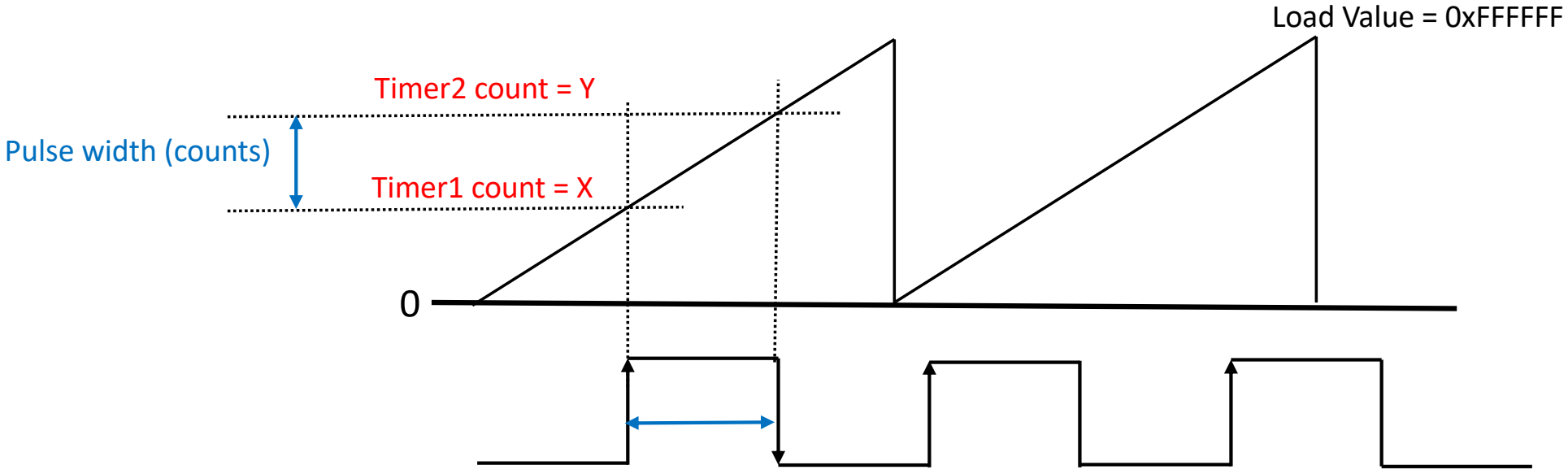


Measurement Limitations

- Measured period resolution.
  - 1 CPU clock period @ 120 MHz = 8.333 nsec.
- Maximum measurable period is the period of the timer.
  - Maximum measurable period:
    - Maximum count value * CPU clock period.
  - TM4C1294 specific values.
    - Maximum 24-bit value = 16, 777, 215.
    - Maximum period = 16,777,215*8.333 nsec = ~0.14 second.
- Minimum measurable period = maximum ISR response time.
  - The ISR must complete before the next edge event. (relative deadline).
  - For zero latency interrupts we saw latencies of ~ 0.35 usec + the execution time to calculate the period.
    - To be conservative don't plan on measuring signals faster than ~ 1MHz without careful profiling.
  - For RTOS controlled interrupts we saw latencies of ~4 usec + execution time.

Measuring Pulse Width

- If you want to know the time between input signal transitions, simply program the the trigger event to both rising and falling edges.
- If you want to know the positive pulse width this requires two synchronized timers.
  - Synchronized timers start counting at the same time and have the same count.
  - Timer1 captures on the rising edge.
  - Timer2 captures on the falling edge.
  - Pulse width in counts = timer2 count – timer1 count.
- How small a pulse width can we measure with this method?
  - ?



Pulse width (seconds) = (timer2 count – timer1 count) * system clock period.

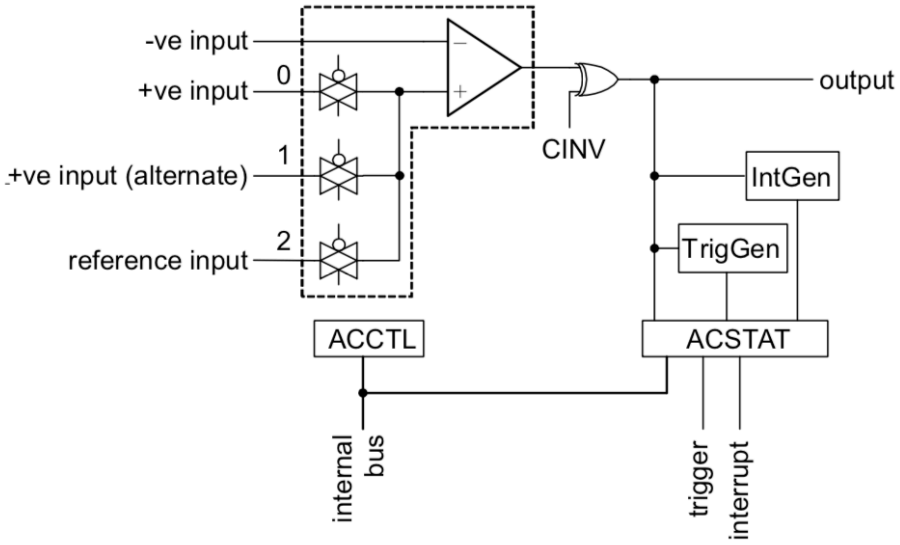
Pulse Width Measurement Limits with two timers.

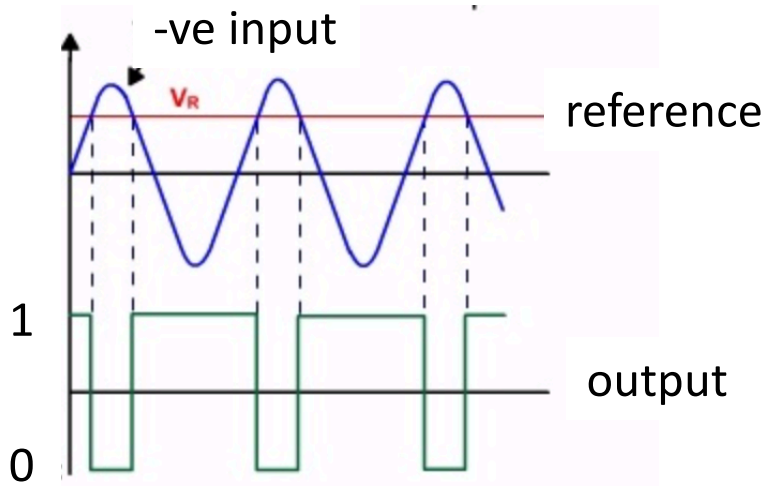
- Our minimum measurable pulse width is not limited by the ISR execution time as it was with period.
  - We have two separate timers that can be triggered independently.
- However, the event trigger has minimum conditions for detecting an event.
  - For rising edge detection, the input must be high for two clocks following the edge.
  - For falling edge detection, the input must be low for two clocks following the edge.
- The minimum measurable pulse width is two system clocks.
- The maximum measurable pulse width is still limited by the timer period, ~ 0.14 seconds.

Measuring Analog Signal Period

- A comparator with a set reference can be used to convert an analog signal into a digital signal.
  - If analog voltage >= Reference voltage: Comparator output = 0.
  - If analog voltage < Reference voltage: Comparator output = 1.
- TM4C1294 has an analog comparator peripheral.
  - It can be used to trigger an interrupt or start an ADC sampling sequence.
    - To measure the period, we would trigger an interrupt to read a timer count value and perform the same period measurement as before.
  - It can compare
    - Two external analog inputs to each other, -ve and +ve in picture
    - One external analog input to an internal programmable reference voltage.
  - It has programmable polarity, output can be 1 or 0 for analog voltage >= reference.

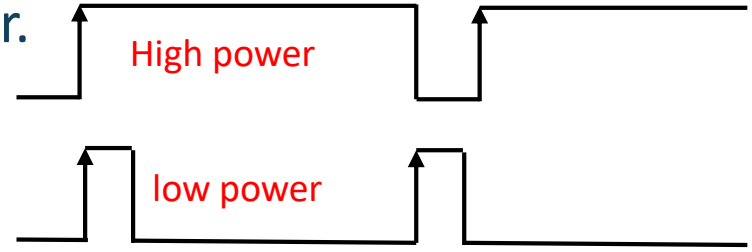
Figure 22-2. Structure of Comparator Unit





Pulse Width Modulation

- Pulse Width Modulation
  - Is a method for controlling the average power into a load by varying the pulse width.
  - Higher pulse widths deliver higher power.
  - Lower pulse widths lower power.



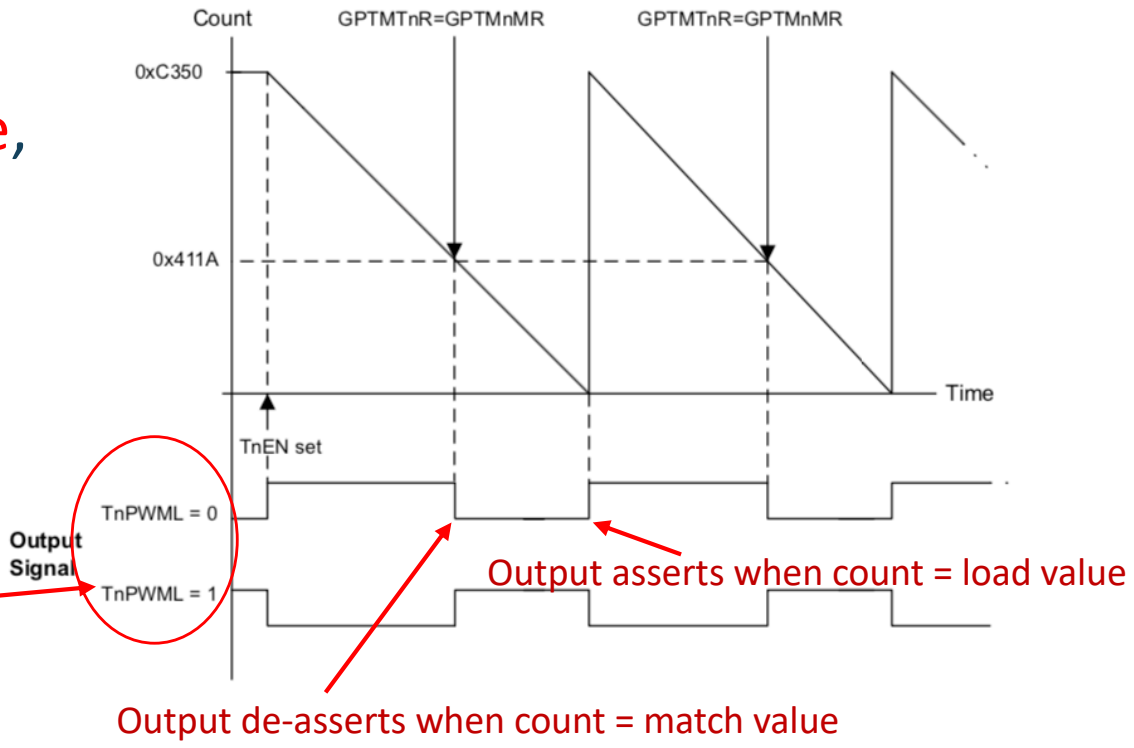
- Applications
  - LED control for brightness.
  - Motor / Servo control.
  - Switching power supplies.
  - Audio applications (Lab 3)
- PWM support in TM4C1294
  - Timers can be configured in compare mode to output a PWM signal.
  - PWM generator peripheral with 4 PWM blocks and 8 outputs.

Timer: PWM Mode

- PWM mode configuration
  - The Timer is limited counting down with a a maximum of 24- bits.
  - Set the load value to determine the period / frequency of the output signal. (GPTMTnILR / GPTMTnPR registers).
  - Set the timer match value to determine the pulse width (GPTMTnMATCHR / GPTMTnPMR registers).
  - Configure the CCP pin as an output in peripheral mode.

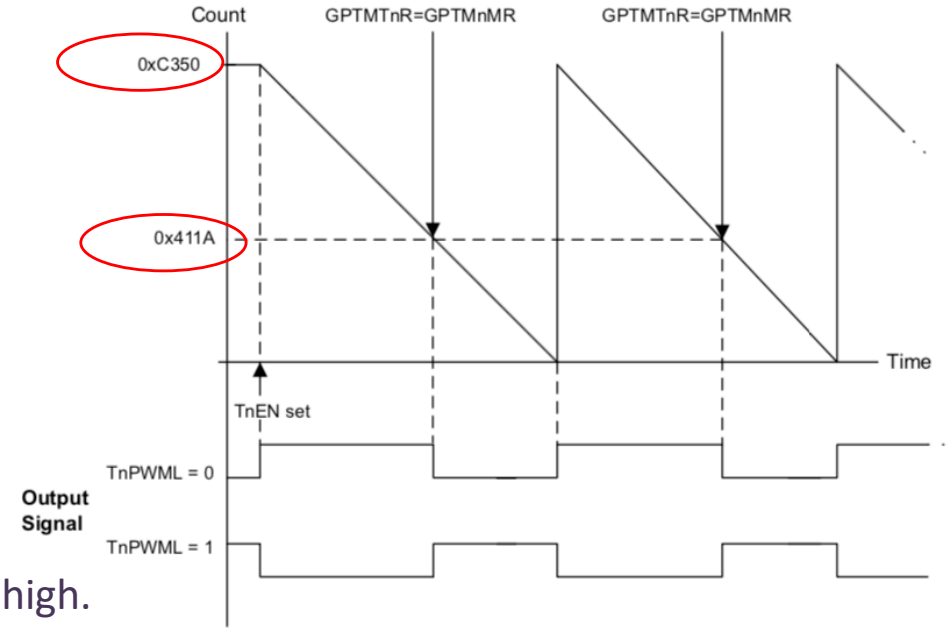
- PWM Operation
  - If the count = load value,
    - The output asserts.
  - If the count value = the match value,
    - The output de-asserts.
  - The output polarity is configurable.

Figure 13-4. 16-Bit PWM Mode Example



Timer: PWM Mode Example

Figure 13-4. 16-Bit PWM Mode Example



- Example
  - System Clock Frequency:
    - 50 MHz / 20 nsec
  - Output Signal Period:
    - 1 msec
  - Output Signal Duty cycle:
    - 66% with TnPWML = 0
    - Duty cycle is the % of time the output is high.
- Load Value calculation
  - Output signal period / system clock period
    - 1 msec / 20 nsec = 50000 = 0xC350
- Duty cycle calculation (TnPWML = 0)
  - Load value *(1- duty cycle)
    - 50000(1 - 0.66) = 16666 = 0x411A
  - The load is scaled by (1 – duty cycle) because the timer is counting down.

PWM Module

- The PWM Module offers a lot more flexibility than the PWM Timer mode.
- PWM Module Overview.
  - The module 4 PWM generators.
    - They can be operated independently or synchronized.
    - Each has two outputs.
    - The polarity of the outputs is configurable.
  - The module can divide down the system clock for longer count periods.
  - Each generator has…
    - A 16-bit counter that can count up and / or down.
    - The zero and load count condition can trigger events.
    - Two comparators A and B can also trigger events.
    - When an event occurs it can be configured to,
      - Do nothing / ignore event.
      - Toggle the output.
      - Force the output high.
      - Force the output low.
- In contrast, the Timer in PWM mode.
  - Only had one comparator, i.e. one match value.
  - Could only force high on load value and force low on the match value when TnPWML = 0.