Timer Modules

ENCE361 Embedded Systems 1

Course Coordinator: Ciaran Moore (ciaran.moore@Canterbury.ac.nz)

Lecturer: Le Yang (le.yang@canterbury.ac.nz)

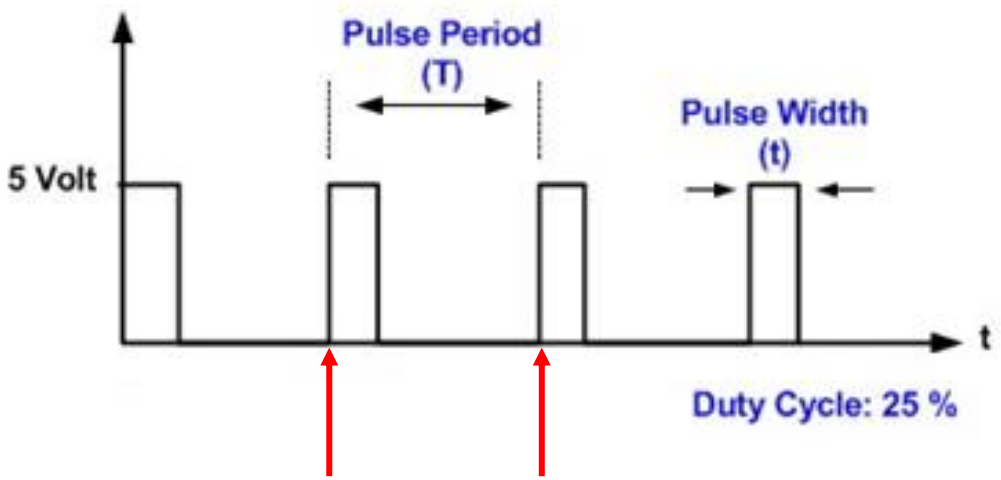
Department of Electrical and Computer Engineering

Where we're going today

- **Overview of timer modules**
- General-purpose timer modules (GPTMs) on Tiva C-series TM4C123x MCU
- Watchdog timer module
- Homework

Basic Timer Functions (1)

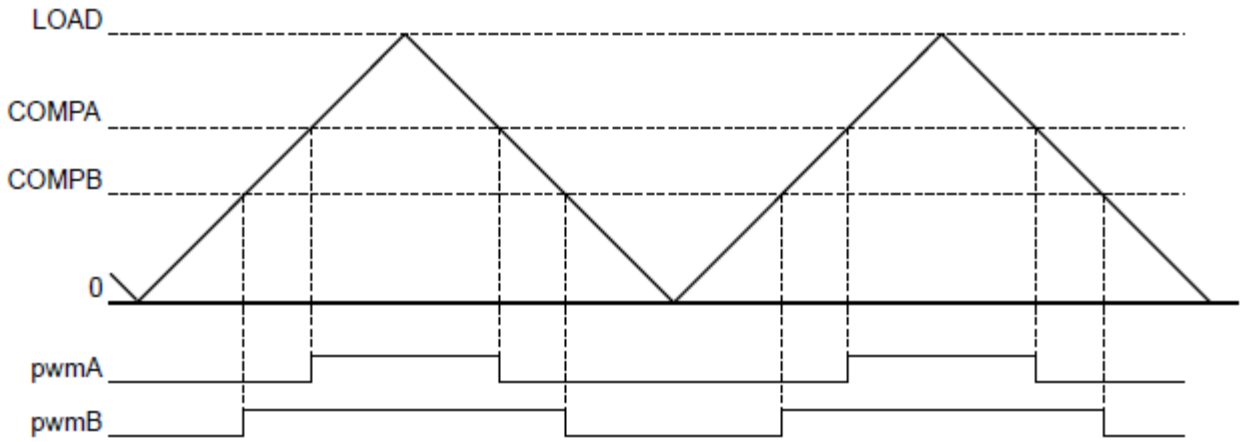
- All timer operations are based on free-running counters
- Period Measurement
  - E.g., to measure the period of square ware, need to capture and time-stamp edges
  - Typically referred to as input capture



Edge detection and time-stamping

Basic Timer Functions (2)

- Waveform generation
  - The value of the free-running counter is compared to a user-programmed threshold and actions are taken if they match
  - Typically referred to as output compare
  - Example: generate 2 pulse width modulation (PWM) signals using a count-up/down timer and 2 thresholds, COMPA and COMPB (see lecture 17)



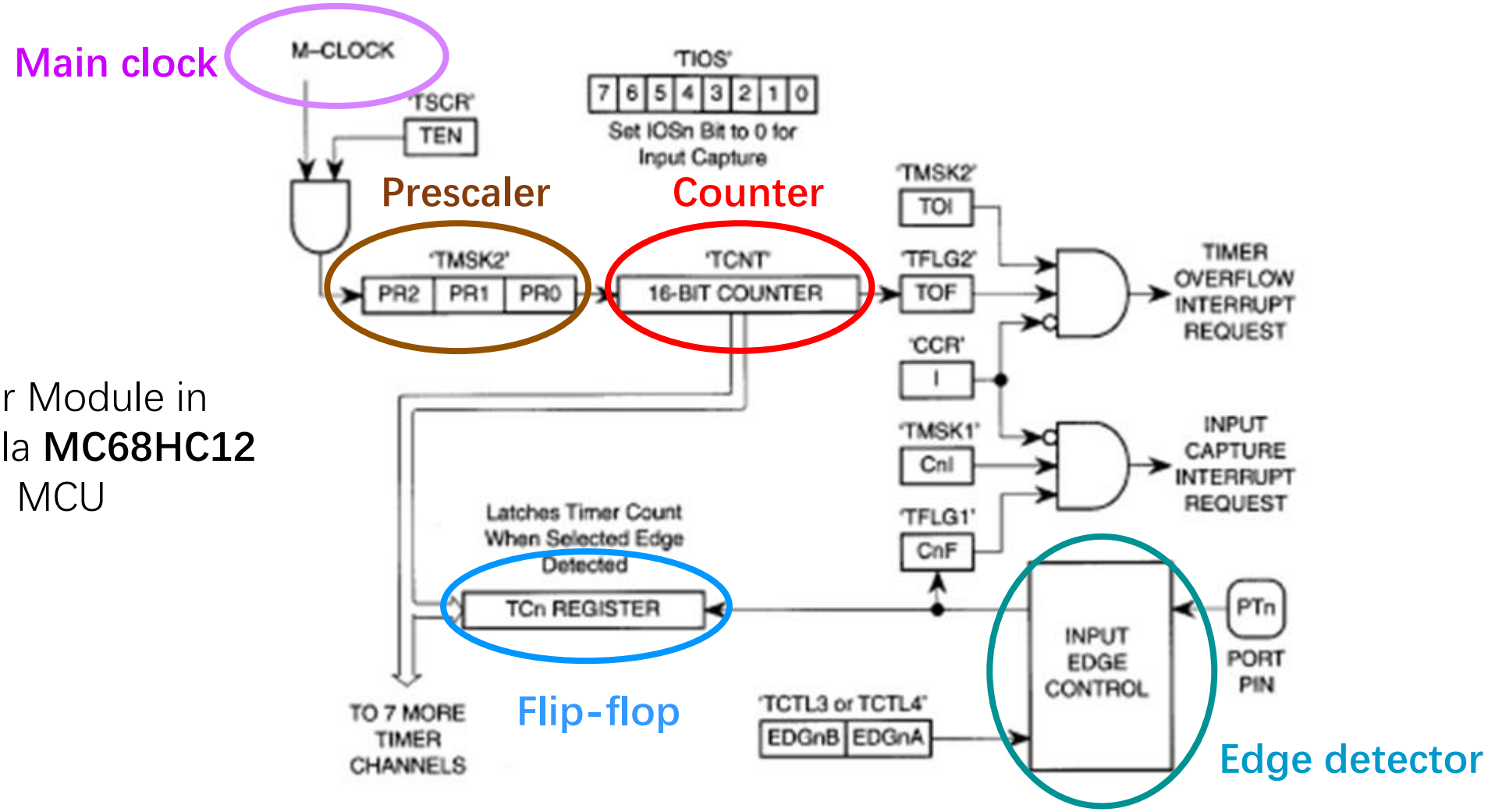
From page 1236, TM4C123GH6PM Data Sheet.pdf

Desired Characteristics for Timer Module

- Support a free-running programmable counter as wide (bit-wise) as possible to achieve good time resolution
- Provide a prescaler to extend the timer period
  - Prescaler is an additional counting circuit to reduce a high frequency signal to a lower frequency one via integer division (see Homework Problems 1 & 3)
- Provide multiple I/O pins
- Support both interrupt and polling operations

Input Capture in "Simpler" Timer Module

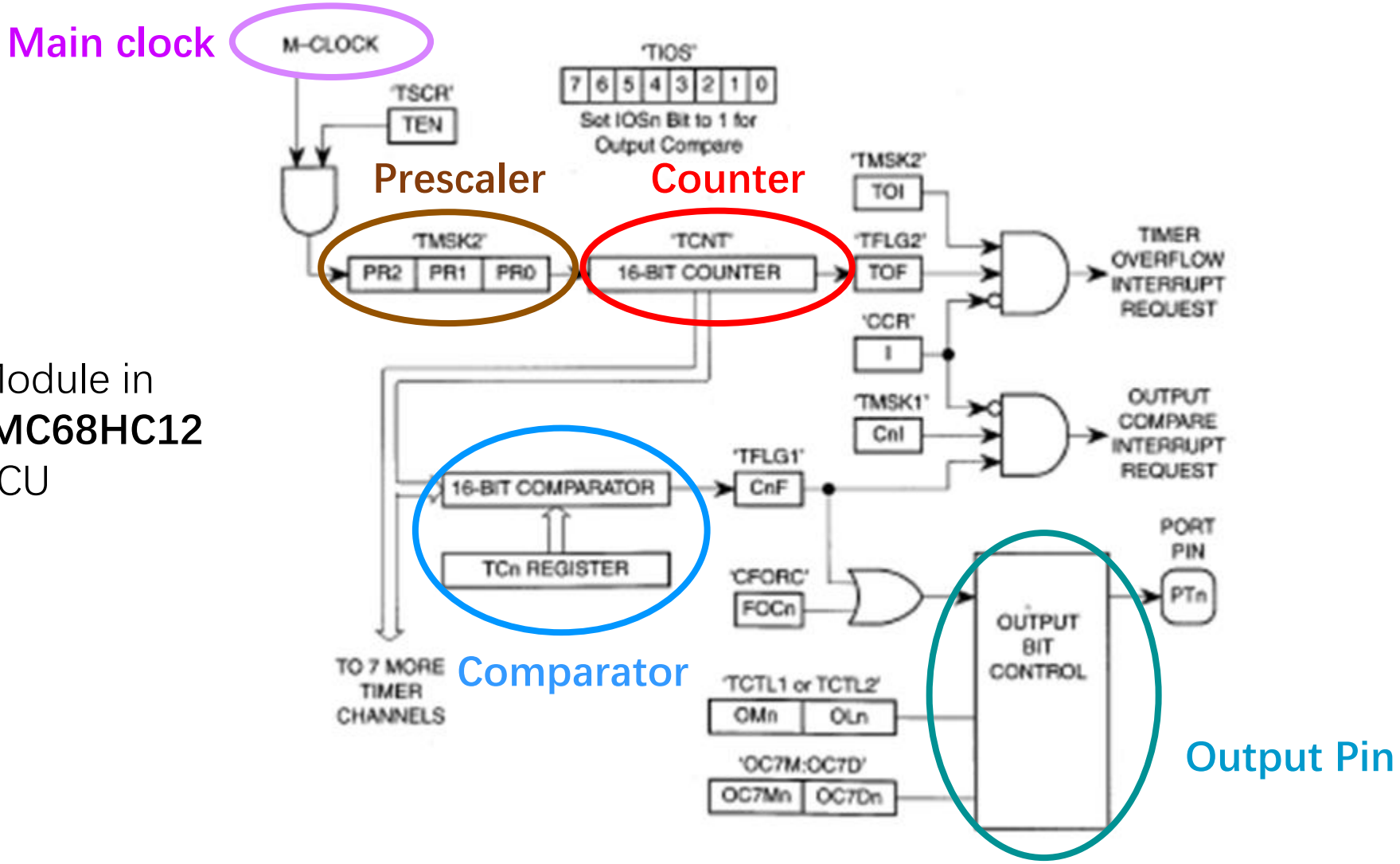
Timer Module in Motorola **MC68HC12** MCU



From F. M. Cady, Software and Hardware Engineering: Motorola M68HC12, Oxford University Press, 2000

Output Comparison in "Simpler" Timer Module

Timer Module in Motorola **MC68HC12** MCU



From F. M. Cady, Software and Hardware Engineering: Motorola M68HC12, Oxford University Press, 2000

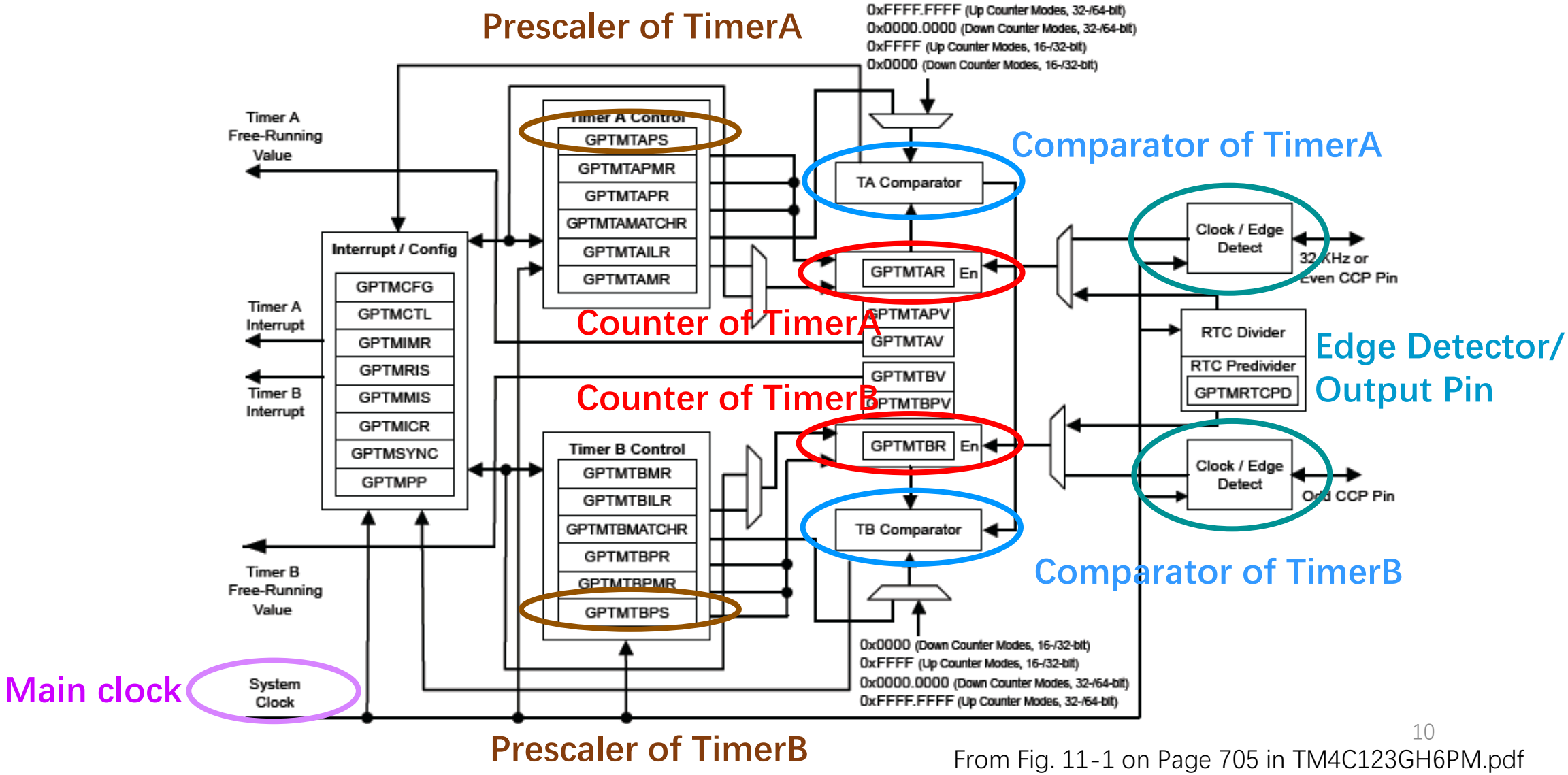
Where we're going today

- Overview of timer modules
- **General-purpose timer modules (GPTMs) on Tiva C-series TM4C123x MCU**
- Watchdog timer module
- Homework

GPTM in Tiva C-Series TM4C123x MCU

- 12 general-purpose timer modules (GPTMs)
  - 6 are 16/32-bit modules
    - Each has 2 16-bit (half-width) timers, TimerA and TimerB
    - Two 16-bit timers can be cascaded into one 32-bit (full-width) timer, TimerA
  - 6 are 32/64-bit modules
    - Each has two 32-bit (half-width) timers, TimerA and TimerB
    - Two 32-bit timers can be cascaded into one 64-bit (full-width) timer, TimerA
  - Each module has one down-counter
  - Once the down-counter reaches 0, an interrupt can be generated to trigger other timers or an event such as ADC conversion

Tiva C-Series TM4C123x GPTM Block Diagram



From Fig. 11-1 on Page 705 in TM4C123GH6PM.pdf

Timer Initialization for Input Capture

```
void initialize_timer(void)
{
    // Timer initialisation
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOB);    // Check TM4C123GH6PM Data Sheet.pdf
    SysCtlPeripheralEnable(SYSCTL_PERIPH_TIMER2);  // for specific pin and the associated timer

   // Configure timer A for input edge time capture
   TimerDisable(TIMER2_BASE, TIMER_BOTH);                  // Disable timer for setup

   GPIOPinTypeTimer(GPIO_PORTB_BASE, GPIO_PIN_0);    // Use PB0 for input
   GPIOConfig(GPIO_PB0_T2CCP0);

   TimerConfigure(TIMER2_BASE, TIMER_CFG_A_PERIODIC);                  // Periodic timer
   TimerControlEvent(TIME2_BASE, TIMER_A, TIMER_EVENT_POS_EDGE);  // Detect positive edge

   TimerEnable(TIMER2_BASE, TIMER_A);                  // Enable timer
}
```

Example Code for Event Measurement

```
//
// Fill two buffers with ADC reads and return time it takes
//
uint32_t fill_buffers(uint16_t *buf0Ptr, uint16_t *buf1Ptr)
{
        // Record the initial time
        uint32_t first_time = TimerValueGet(TIMER0_BASE, TIMER_A);

        uint32_t i;
        for (i=0; i < WAVE_BUFFER_SIZE; i++)
        {
             getADC(buf0Ptr + i , buf1Ptr + i);
        }

        // Return the time taken by filling two buffers (Recall: down-counter based timer)
        return (first_time - TimerValueGet(TIMER0_BASE, TIMER_A));
}
```

Where we're going today

- Overview of timer modules
- General-purpose timer modules on Tiva C-series TM4C123x MCU
- **Watchdog timer module**
- Homework

Watchdog Timer Module (1)

- Slow hardware or hardware/software error can lead to system hang
  - System ceases temporarily or permanently to respond to inputs
- Watchdog timer module detects system hang (dog hits the floor) and reset system

Watchdog Timer Module (2)

- Watchdog timer module on Tiva C-series TM4C123x MCU has
  - 32-bit down counter
  - Interrupt generate logic
  - Programmable load register
  - Locking register
- Once 32-bit counter reaches 0, the 1st timeout interrupt is generated and reload counter
- If counter reaches 0 again before the 1st timeout interrupt is cleared, reset system

Watchdog Timer Module (3)

- Reload the load register (kick the dog) before the 32-bit counter reaches 0
  - Counter continues down-counting from the reloaded value

API Functions for Watchdog Timers

- Writing to locking register prevents configured watchdog timer from being altered inadvertently
  - Unlock → reload counter → lock again

- void WatchdogUnlock(uint32_t ulBase)

- void WatchdogReloadSet(uint32_t ulBase, unsigned long ulLoadVal)
  - Configure the value to load into watchdog timer when it reaches 0 for the 1st time
  - Immediately load ulLoadVal into watchdog timer ulBase if it is still running when the function is called (kick the dog)

- void WatchdogLock(uint32_t ulBase)

Homework

1. A 16-bit timer is used to measure intervals up to 200 ms. If the microprocessor clock rate is 20 MHz and the timer clock prescaler is an 8-bit unsigned integer, what is the best time resolution possible for the interval measurements and what is the value of the prescaler to achieve that?

2. With reference to Slide 6, describe what happens when an edge is detected on pin “PTn”, indicating the end of an interval.

3. The code on Slide 12 will return an incorrect interval sometimes. Explain what may cause the problem. Assuming a 16-bit counter, modify the code to overcome the problem.

4. Describe a situation where an embedded system requires the use of a watchdog timer.