DEVICES AND COMMUNICATION BUSES FOR DEVICES NETWORK–

Lesson-13: TIMING AND COUNTING DEVICES

Timer

- Timer is a device, which counts the input at regular interval (δT) using clock pulses at its input.
- The counts increment on each pulse and store in a register, called count register
- Output bits (in a count register or at the output pins) for the present counts.

Evaluation of Time

- The counts multiplied by the interval δT give the time.
- The (present counts –initial counts) × δT interval gives the time interval between two instances when present count bits are read and initial counts were read or set.

Timer

- Has an input pin (or a control bit in control register) for resetting it for all count bits = 0s.
- Has an output pin (or a status bit in status register) for output when all count bits = 0s after reaching the maximum value, which also means after timeout or overflow.

Counter

- A device, which counts the input due to the events at irregular or regular intervals.
- The counts gives the number of input events or pulses since it was last read.
- Has a register to enable read of present counts
- Functions as timer when counting regular interval clock pulses

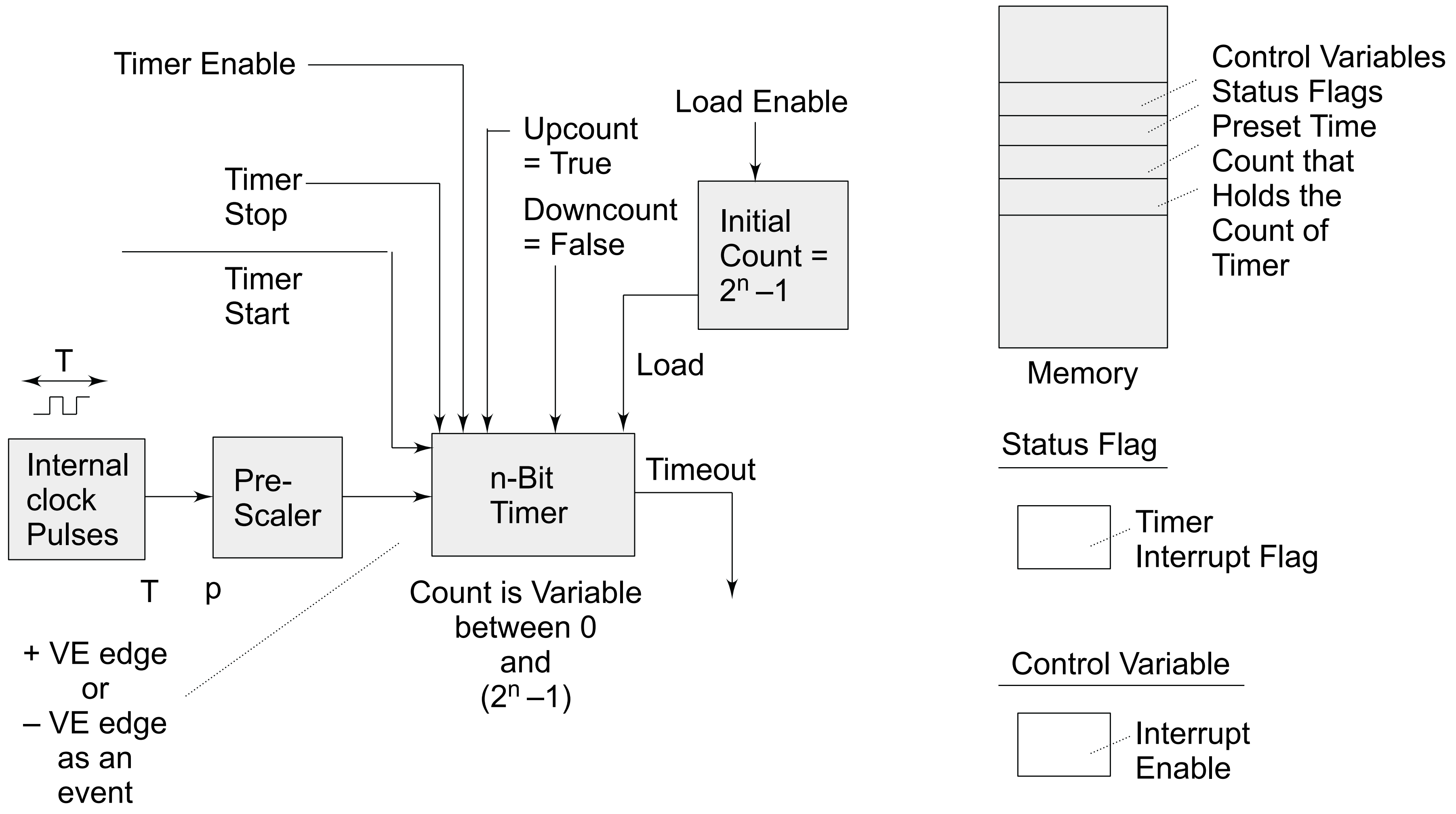
Counter

- Has an input pin (or a control bit in control register) for resetting it for all count bits = 0s.
- Has an output pin (or a status bit in status register) for output when all count bits = 0s after reaching the maximum value, which also means after timeout or overflow.

Timer or Counter Interrupt

- When a timer or counter becomes 0x00 or 0x0000 after 0xFF or 0xFFFF (maximum value), it can generate an ‘interrupt’, or an output ‘Time-Out’ or set a status bit ‘TOV’

Timer cum Counting Device



A Hardware Timer is a Counter that gets Clock Period Inputs at Regular Intervals

Free running Counter (Blind running Counter)

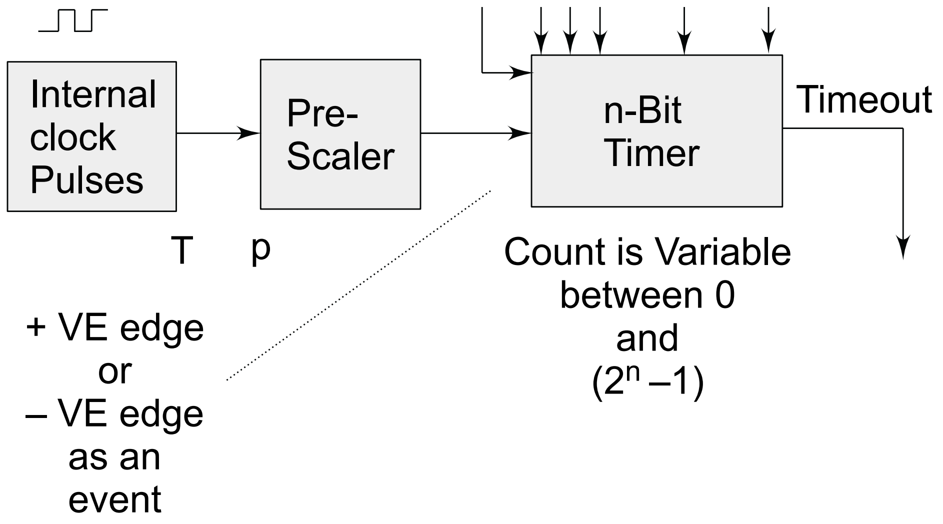
- A counting device may be a free running (blind counting) device giving overflow interrupts at fixed intervals
- A pre-scalar for the clock input pulses to fix the intervals

Free Running Counter

It is useful

- for action or initiating chain of actions,
- processor interrupts at the preset instances
- noting the instances of occurrences of the events
- processor interrupts for requesting the processor to use the capturing of counts at the input instance
- comparing of counts on the events for future actions

Free running Timer cum Blind Counting Device


Internal clock Pulses
Pre-Scaler
n-Bit Timer
Timeout
T
p
+ VE edge or – VE edge as an event
Count is Variable between 0 and (2^n –1)

A Hardware Timer is a Counter that gets Inputs at Regular Intervals

Free running (blind counting) device Many Applications Based on

- comparing the count (instance) with the one preloaded in a compare register [an additional register for defining an instance for an action]
- capturing counts (instance) in an additional register on an input event. [An addition input pin for sensing an event and saving the counts at the instance of event and taking action.]

Free running (Blind Counts) input OC-enable pin (or a control bit in control register)

- For enabling an output when all count bits at free running count = preloaded counts in the compare register.
- At that instance a status bit or output pin also sets in and an interrupt ‘OCINT’ of processor can occur for event of comparison equality.
- Generates alarm or processor interrupts at the preset times or after preset interval from another event

Free running (Blind Counts) input capture -enable pin (or a control bit in control register) for Instance of Event Capture

- A register for capturing the counts on an instance of an input (0 to 1 or 1 to 0 or toggling) transition
- A status bit can also sets in and processor interrupt can occur for the capture event

Free running (Blind Counts) Pre-scaling

- Prescalar can be programmed as p = 1, 2, 4, 8, 16, 32, .. by programming a prescaler register.
- Prescalar divides the input pulses as per the programmed value of *p*.
- Count interval = $p \times \delta T$ interval
- δT = clock pulses period, clock frequency = δT^{-1}

Free running (Blind Counts) Overflow

- It has an output pin (or a status bit in status register) for output when all count bits = 0s after reaching the maximum value, which also means after timeout or overflow
- Free running n-bit counter overflows after $p \times 2^n \times \delta T$ interval

Uses of a timer device

- Real Time Clock Ticks (System Heart Beats). [Real time clock is a clock, which, once the system starts, does not stop and can't be reset and its *count value* can't be reloaded. *Real time endlessly flows and never returns back*!] Real Time Clock is set for ticks using prescaling bits (or rate set bits) in appropriate control registers.

Uses of a timer device

- Initiating an event after a preset delay time. Delay is as per *count value* loaded.

Uses of a timer device

- Initiating an event (or a pair of events or a chain of events) after a comparison(s) with between the pre-set time(s) with counted value(s). [It is similar to a preset alarm(s).].
- A preset time is loaded in a Compare Register. [It is similar to presetting an alarm].

Uses of a timer device

- Capturing the *count value* at the timer on an event. The information of *time* (instance of the event) is thus stored at the *capture register*.

Uses of a timer device

- Finding the time interval between two events. *Counts* are captured at each event in capture register(s) and read. The intervals are thus found out.

Uses of a timer device

- Wait for a message from a queue or mailbox or semaphore for a preset time when using RTOS. There is a A predefined waiting period is done before RTOS lets a task run.

Uses of a timer device

- Watchdog timer. It resets the system after a defined time.

Uses of a timer device

- Baud or Bit Rate Control for serial communication on a line or network. Timer timeout interrupts define the time of each baud

Uses of a timer device

- Input pulse counting when using a timer, which is ticked by giving non-periodic inputs instead of the clock inputs. The timer acts as a counter if, in place of clock inputs, the inputs are given to the timer for each instance to be counted.

Uses of a timer device

- Scheduling of various tasks. A chain of software-timers interrupt and RTOS uses these interrupts to schedule the tasks.

Uses of a timer device

- Time slicing of various tasks. A multitasking or multi-programmed operating system presents the illusion that multiple tasks or programs are running simultaneously by switching between programs very rapidly, for example, after every 16.6 ms.
- Process known as a *context switch*. [RTOS switches after preset time-delay from one running task to the next. task. Each task can therefore run in predefined slots of time]

Uses of a timer device

- Time division multiplexing (TDM)
- Timer device used for multiplexing the input from a number of channels.
- Each channel input allotted a distinct and fixed-time slot to get a TDM output. [For example, multiple telephone calls are the inputs and TDM device generates the TDM output for launching it into the optical fiber.

Timer States

Reset State (initial count = 0)
Initial Load State (initial count loaded)
Present State (counting or idle or before start or after overflow or overrun)
Overflow State (count received to make count = 0 after reaching the maximum count)
Overrun State (several counts received after reaching the overflow state)
Running (Active) or Stop (Blocked) state

Timer States

Finished (Done) state (stopped after a preset time interval or timeout)
Reset enabled/disabled State (enabled resetting of count = 0 by an input)
Load enabled/disabled State (reset count = initial count after the timeout)
Auto Re-Load enabled/disabled State (enabled count = initial count after the timeout)
Service Routine Execution enable/disable State (enabled after timeout or overflow)

Summary

We learnt

- Timer
- Counter
- Free running counter or blind counting counter
- Out-compare register and actions on compare
- Input-capture register and actions on capture

We learnt

- Timer uses
- Timer states

End of Lesson 13 of Chapter 3