UNIT II

EMBEDDED NETWORKING

2.5 TIMING AND COUNTING DEVICES

Timer

- Timer is a device , which counts the input at regular interval (δ T) 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

Time counts multiplied by the interval δ T give the time.

• The (present counts −initial counts) ×δ T 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 re setting it for all count bits =0s.
- _Has an output in (or a status bit 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

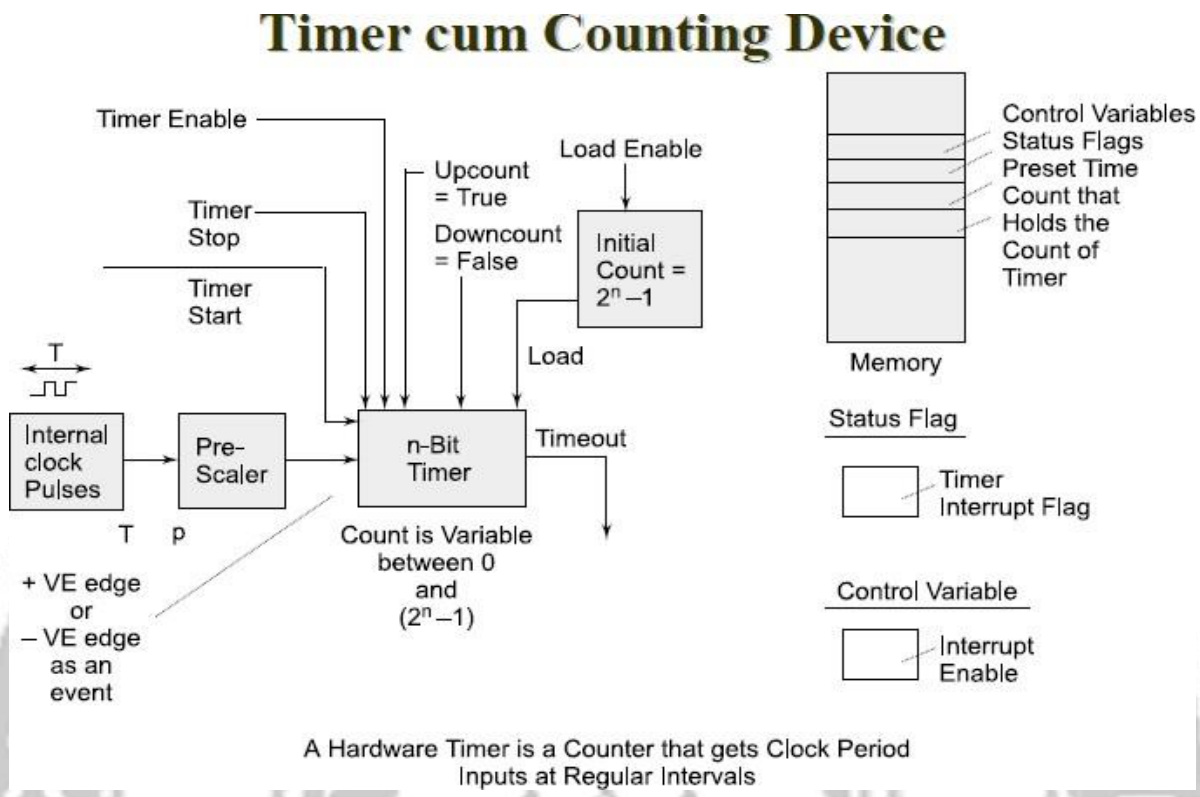
• Functions as timer when counting regular interval clock pulses

_ Has an input pin (or a control bit in control register) for resetting it for all count bits =0s.

_Has an output in (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 0x00or 0x0000after 0xFF or 0 x FFFF(maximumvalue), itcangeneratean_interrupt‘,oranoutput_Time-Out‘orsetastatusbit_TOV‘



Free running Counter (Blind Running Counter)

• A counting device may be a free running(blind counting) device giving over flow interrupts at fixed intervals

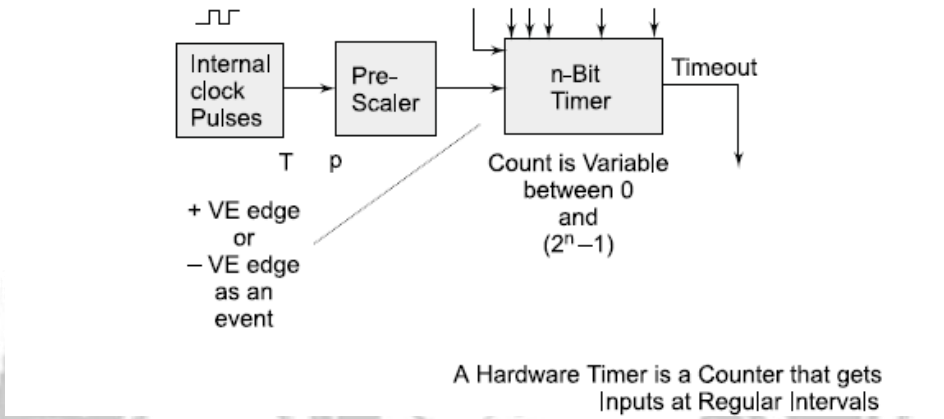
• A pre-scalar for the clock input pulses to fix the intervals Free Running Counter

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



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 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(0to1 or1 to0 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

• Pre scalar can be programmed as p =1, 2, 4, 8, 16, 32, by programming a pre scale register.

•Pre scalar divides the input pulses as per the programmed value of *p*.

• Count interval = p × Δ t interval

• Δ T=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 over flows after p×2n×δTinterval

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 pre scaling bits (or rate set bits) in appropriate control registers.
- Initiating an event after a preset delay time. Delay is as per *count value* loaded.
- 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 pre setting an alarm].
- 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*.
- 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.
- Wait for a message from a queue or mail box or semaphore for a pre set time when using RTOS. There is a pre defined waiting period is done before RTOS lets a task run.

Watch dog timer.

It resets the system after a defined time.

- _ Baud or Bit Rate Control for serial communication on a line or network. Timer time out interrupts define the time of each baud
- _Input pulse counting when using a timer which is ticked by giving on 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.
- _ Scheduling of various tasks .A chain of software - timers interrupt and RTOS uses these interrupts to schedule the tasks.
- _Time slicing of various tasks. A multitasking or multi-programmed operating system resents the illusion that multiple tasks or programs are running simultaneously by switching between programs very rapidly for example, afterevery16.6 m s.

- _ 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 pre defined slots of time]

Time division multiplexing (TDM)

- _ Timer device used for multiplexing the input from a number of channels.
- _ Each channel in put 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.

Software Timer

_ A software, which executes and increases or decreases account- variable (count value) on an interrupt from on a system timer output or from on a real time clock interrupt.

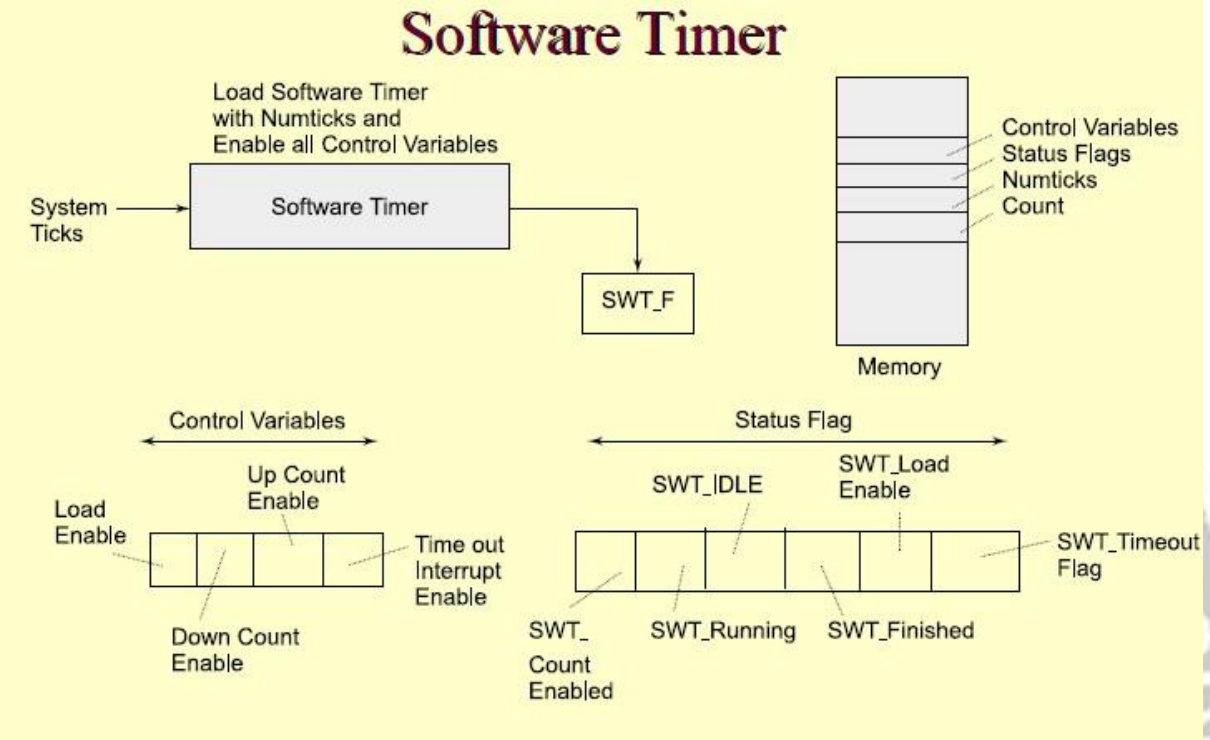
_ The software timer also generate interrupt on over flow of count-value or on finishing value of the count variable.

System clock

- In a system an hardware- timing device is programmed to tick at constant intervals.
- At each tick there is an interrupt
- A chain of interrupts thus occur at periodic intervals.
- The interval is as per a preset *count value*
- The interrupts are called system clock interrupts, when used to control the schedules and timings of the system

Software timer (SWT)

- SWT is a timer based on the system clock interrupts
- The interrupt functions as a clock input to an SWT.
- This input is common to all the SWTs that are in the list of activated SWTs.
- Any number of SWTs can be made active in a list.
- Each SWT will set a status flag on its time out (*count-value*reaching0).



• Actions are analogous to that of a hardware timer. While there is physical limit (1, 2 or 3 or4) for the number of hardware timers in a system, SWTs can be limited by the number of interrupt vectors provided by the user.

• Certain processors (microcontrollers) also defines the interrupt vectoraddressesof2 or 4 SWTs