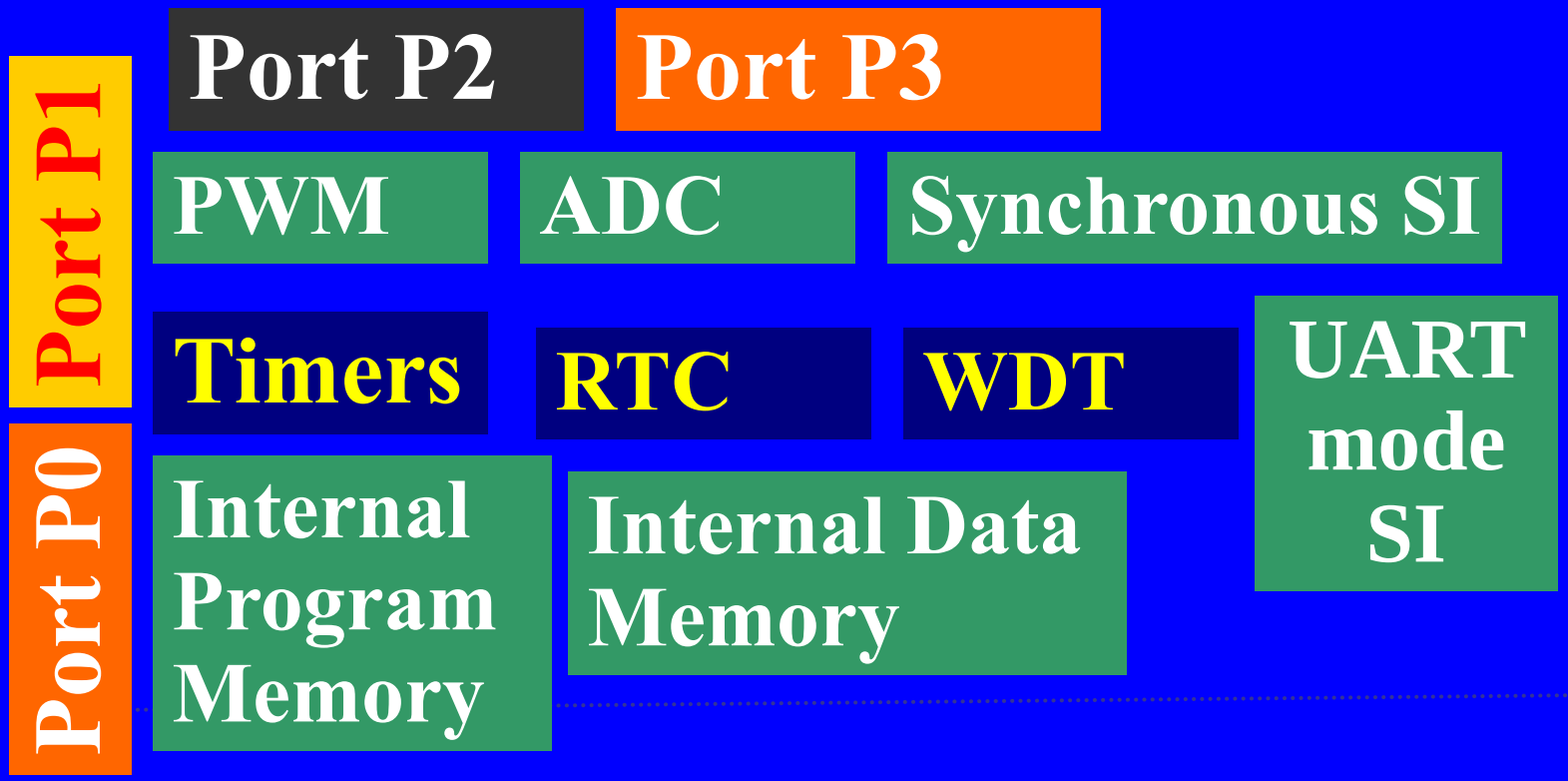
Chapter 2

Overview of Architecture and Microcontroller-Resources

Lesson 4

Timers, Real Time Clock Interrupts and Watchdog Timer

Microcontroller-resources



Timer-Counter Functions

Timer-Counter, overflow, interrupt and ISR

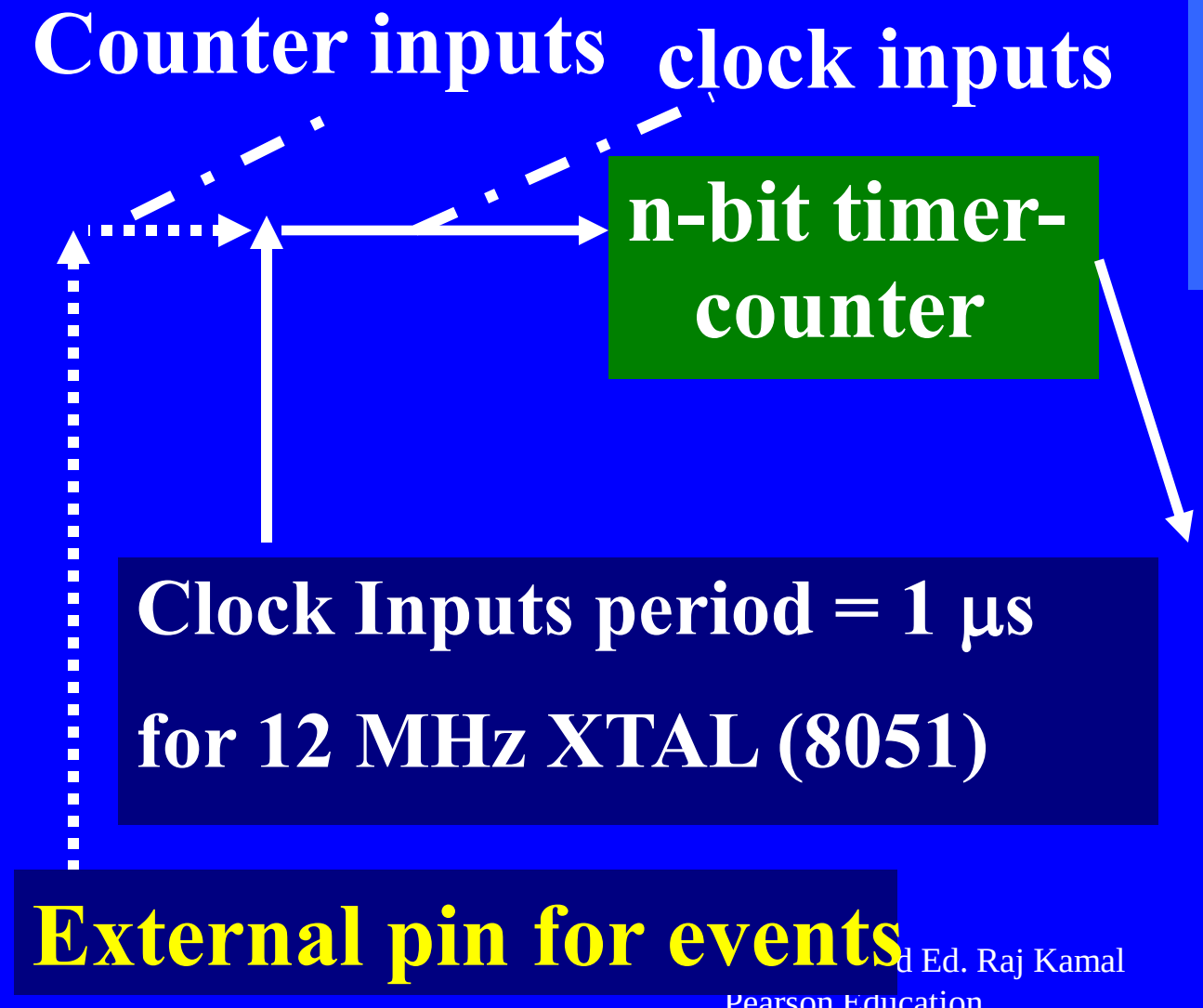
Timer-counter as counter

- A counter gets inputs from external events
- Inputs can be -ve edge transitions at a pin of an MCU
- Counts *x* increment on each count
- Reaches upper limit of all bits = 1s and then on next counts all bits = 0s. The counter is said to overflow

Timer-counter as Timer

- A timer is a counter getting constant interval periodic inputs from a clock source
- Counts *x* increment on each clock-input
- Timer timeouts (overflows) when on all bits becoming = 0s and *x* = 0.

n-bit timer-counter example



Timer overflow interrupt if not masked, an ISR executes

Overflows after 2^n clock inputs if initial count bits all 0s.

ISR (interrupt service routine)

- ISR (interrupt service routine) is a program that runs on an interrupt, for example, on an timer-overflow interrupt
- Return to the interrupted program after finishing running of all the codes.

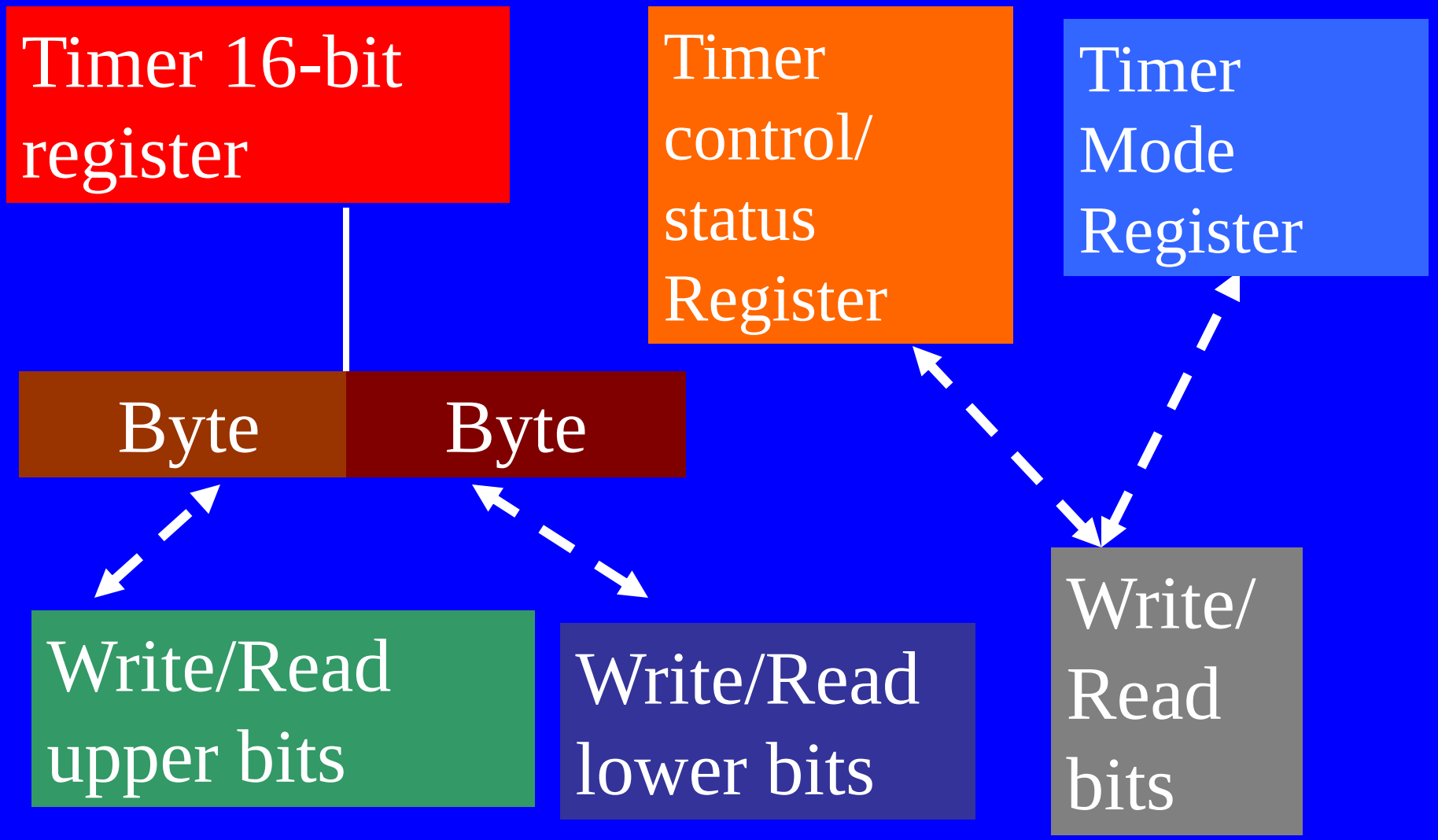
ISR on Timer overflow

- An interrupt can be masked(disabled) to unable the ISR from running. Usually by a bit in interrupt enable register control register or a control register.
- Timer-counter interrupt can also be masked. If not masked,a timer will overflow and interrupt after $(2^n - x)$ clock-inputs.

8-bit Timer-counter

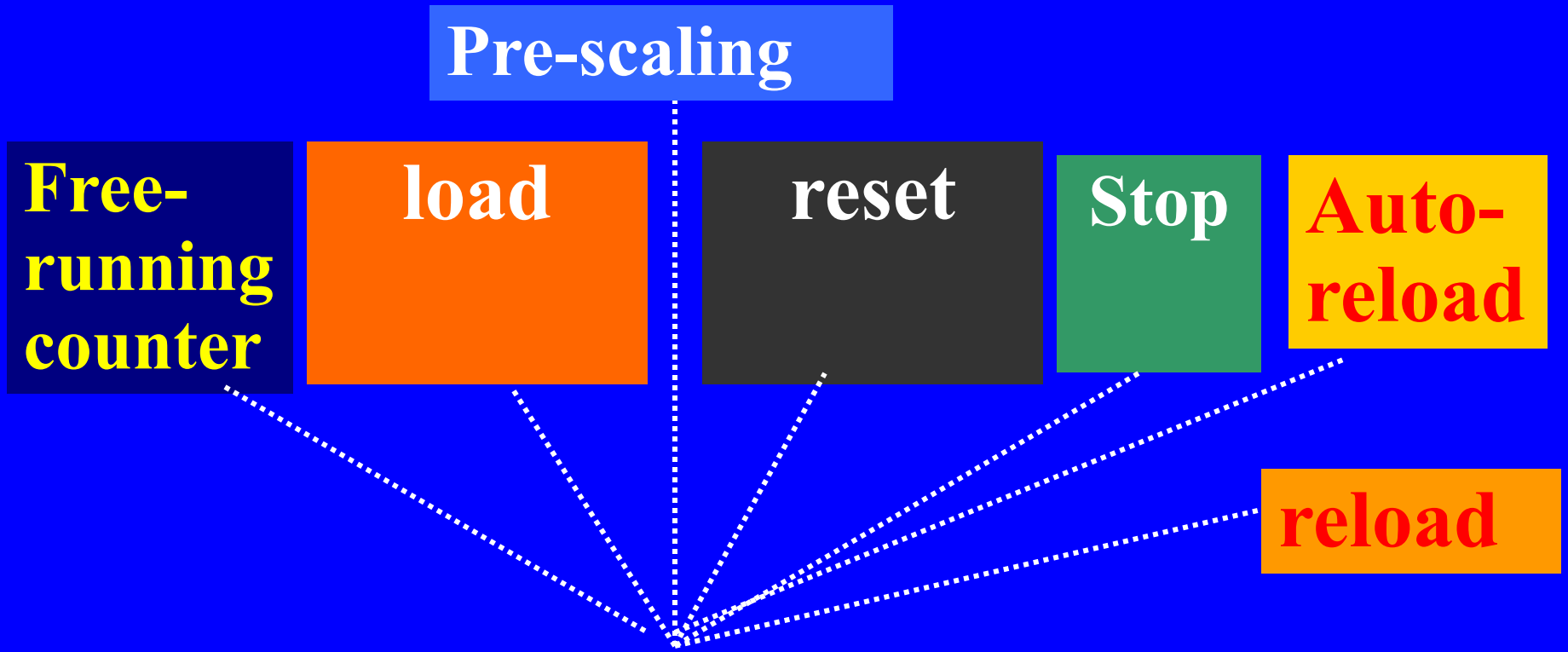
- Let initial value in an 8-bit counter = 0.The 8-bit counter overflows (interrupts) after 2^8= 256 counts or 256 edge transitions.
- Let initial value loaded in an 8-bit timer = x_0 The 8-bit timer overflows after ($2^8 - x$) clock input or (256 - x) edge transitions. If clock inputs are after every 1 μs, then timeout (interrupt) period = (256 -x_0) μs

Timer-Counter Registers



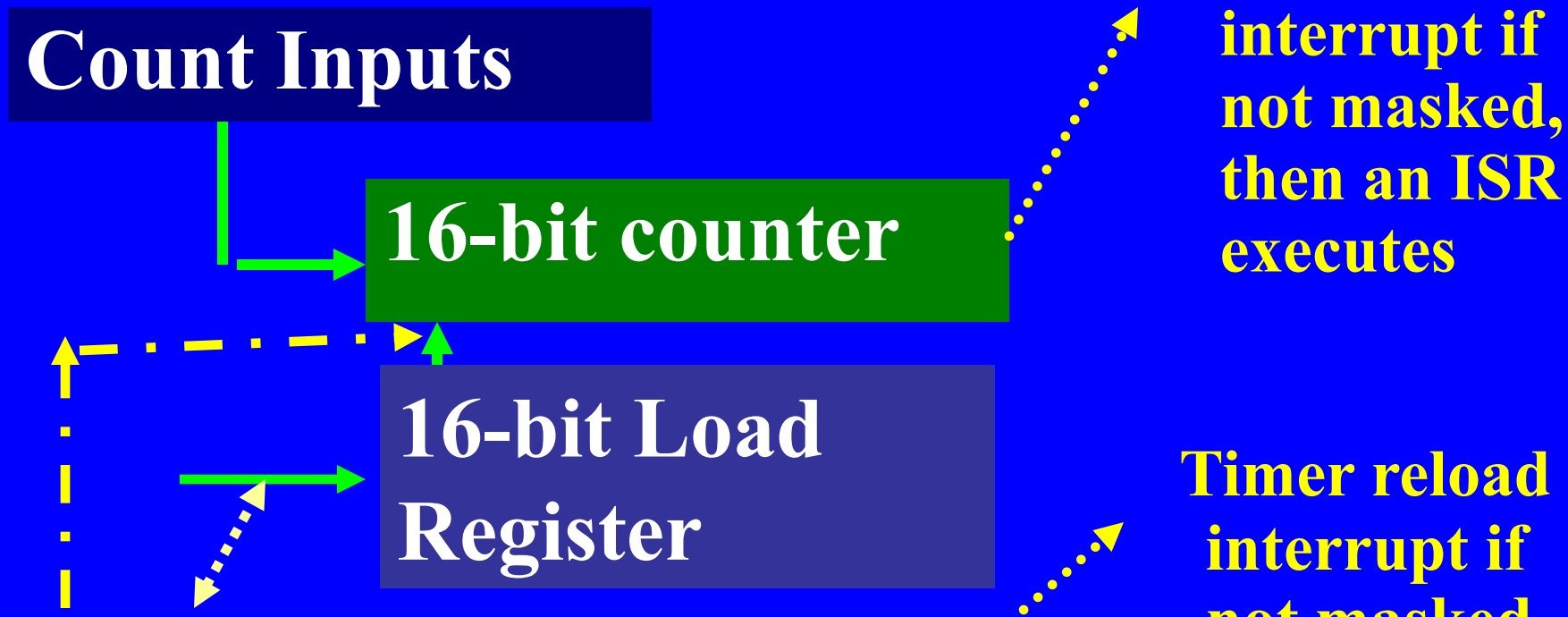
8051/8052 Timing/Counting Devices -

- **T1 and T0 and T2**



Timer-Function options

A Reload of Counter on an input



Free Running Counter (FRC)

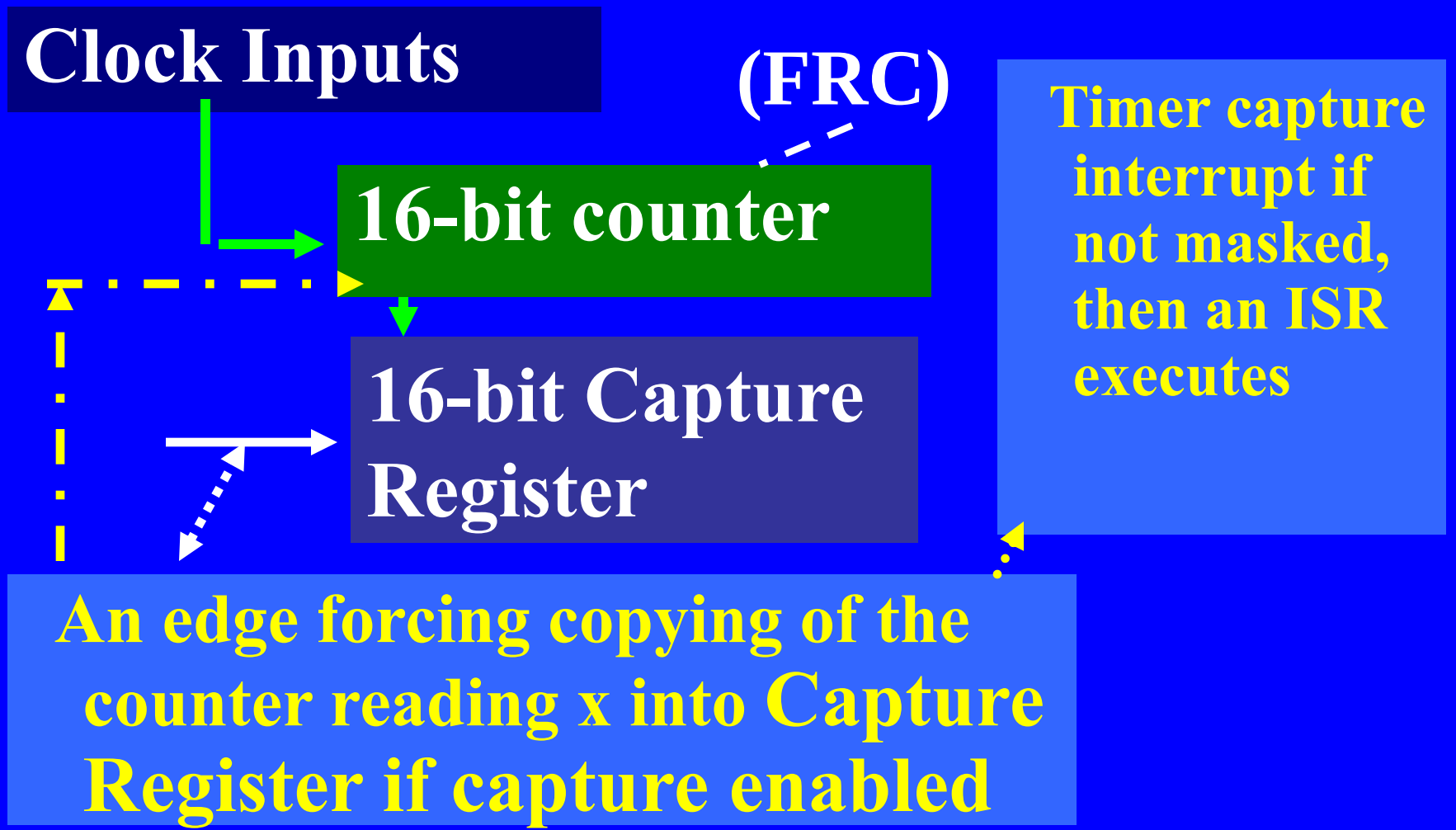
- No reset feasible, No stop feasible
- No initial value load feasible and thus no write to timer-counter register of FRC
- Auto reloads 0s on each successive overflows
- Used for real time control, as its reference is real, unchanged

Pre-scaling

- The clock inputs can be divided by certain factor so that the timer timeouts periods are slowed by that factor. The action is pre-scaling of the counts and factor is called pre-scaling factor.

Timer Input Capture feature

An Input Capture of Timer or FRC



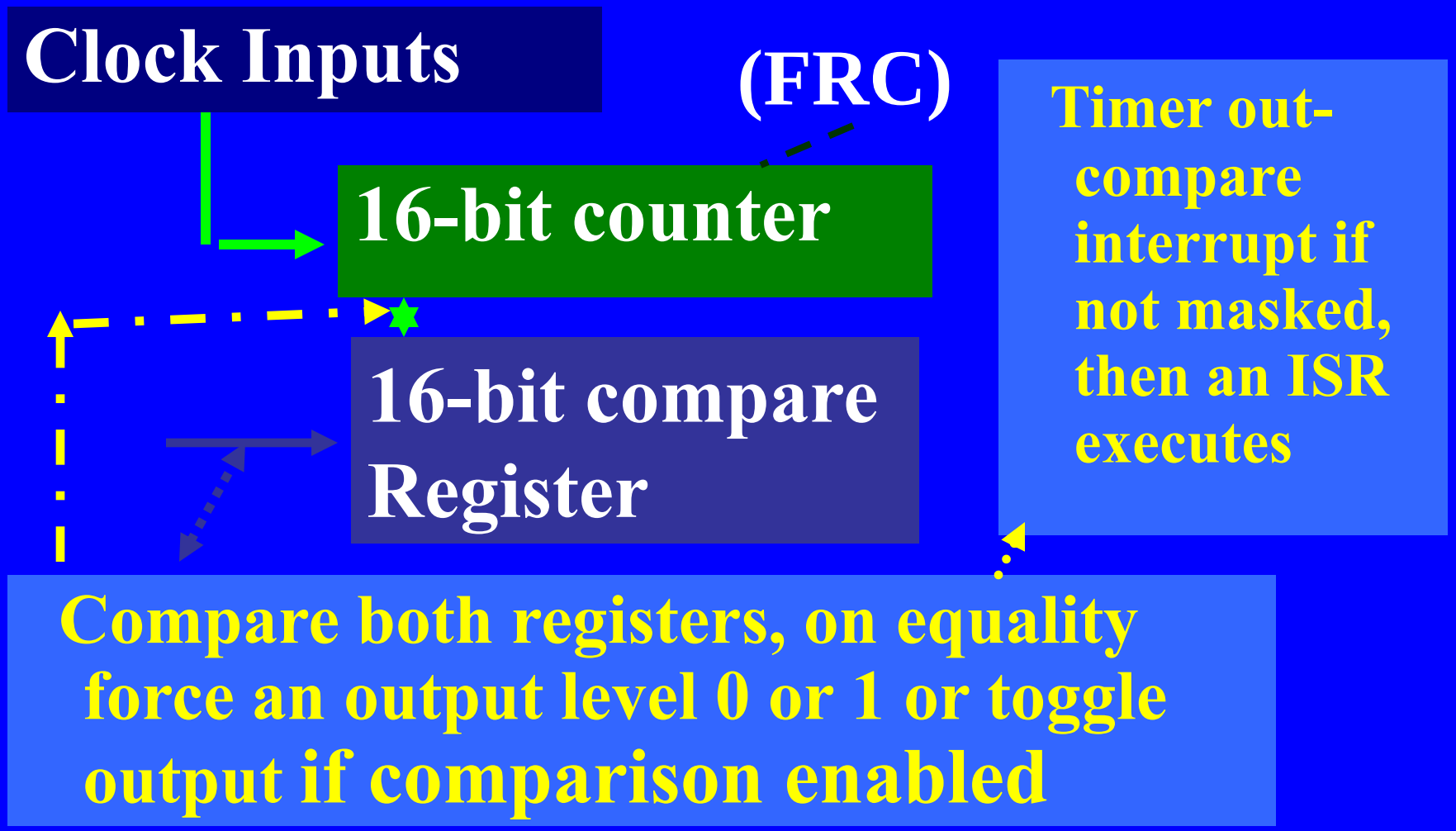
Advantage

Lets an event-time be captured in a register. Enables finding

- **pulse width**
- **pulse (event) frequency**
- **event-interval between the events**

Timer out-compare feature

An Out-Compare in a Timer or FRC

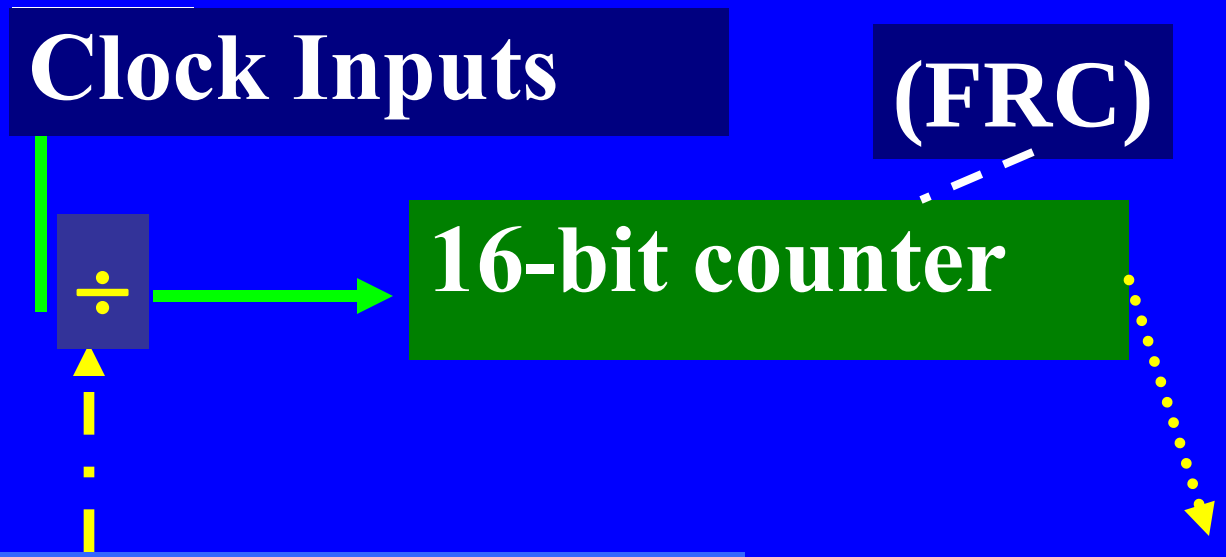


Advantage

- **Lets an output or sequences of outputs fire at a preset time (s)**
- **Lets at a preset time (s), an interrupt(s) occur to initiate an action(s) or initiate sequences of actions, using the ISR.**

Real time clock interrupts and software timer interrupts

Real time clock interrupts using a Timer or FRC



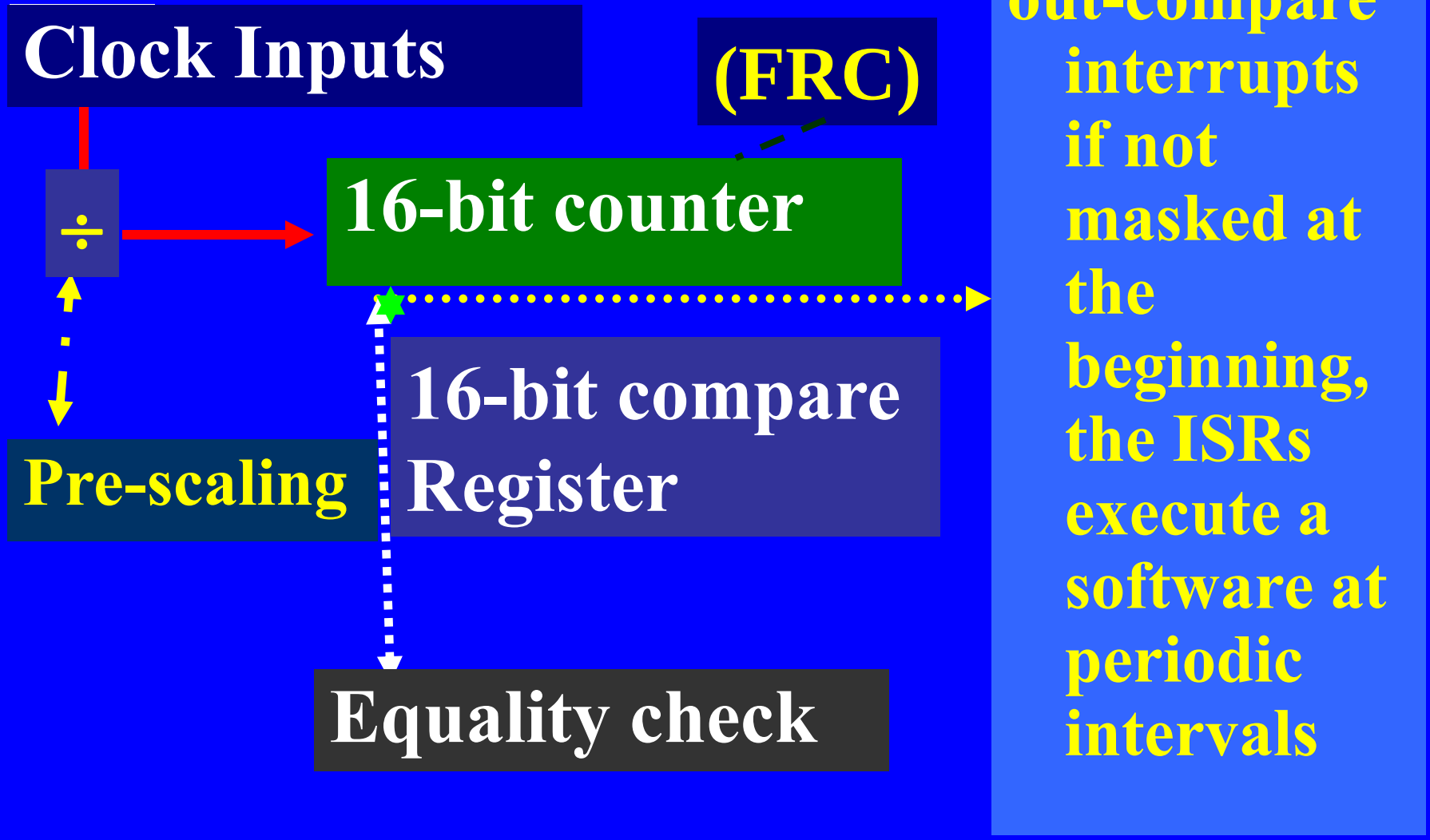
Pre-scaling to control real time clock time tick interval 4 ms or 8 ms

Real time clock interrupts if not masked at the beginning, the ISRs execute at periodic intervals

Advantage

- **Lets the system periodically supervise and schedule the different processes**
- **Used for repeated calls to a task or program or calls to sequences of tasks or programs**

Software Timer interrupts in a Timer or FRC



Advantage

- Lets the system periodically supervise and schedule the different processes at pre-select times
- Used for repeated calls to a task or program or calls to sequences of tasks or programs at pre-select times

Watchdog timer

Watchdog

- Watchdog timer forces system reset to save from system stuck up state (hanging state) during running
- Timeout period can be set at the beginning
- Rewriting can be done to initiate next WDT cycle before a timeout to enable extension of the period by another cycle

Advantage

- Lets the system come out of hang or stuck up situation
- Lets the system recover to initial state after a watched period

Summary

We learnt

Timing Devices

- Timer-counter, overflow and interrupt ISR
- Timer function - as counter
- Timer, reset, load, reload, auto-reload, stop and start

We learnt

Timing Devices

- Free running counter overflow and on interrupt an ISR run
- Input capture function
- Out compare function
- Real time clock interrupts and software timer interrupts

We learnt

Timing devices

- System reset after a watched period using a watchdog timer