

Interrupts

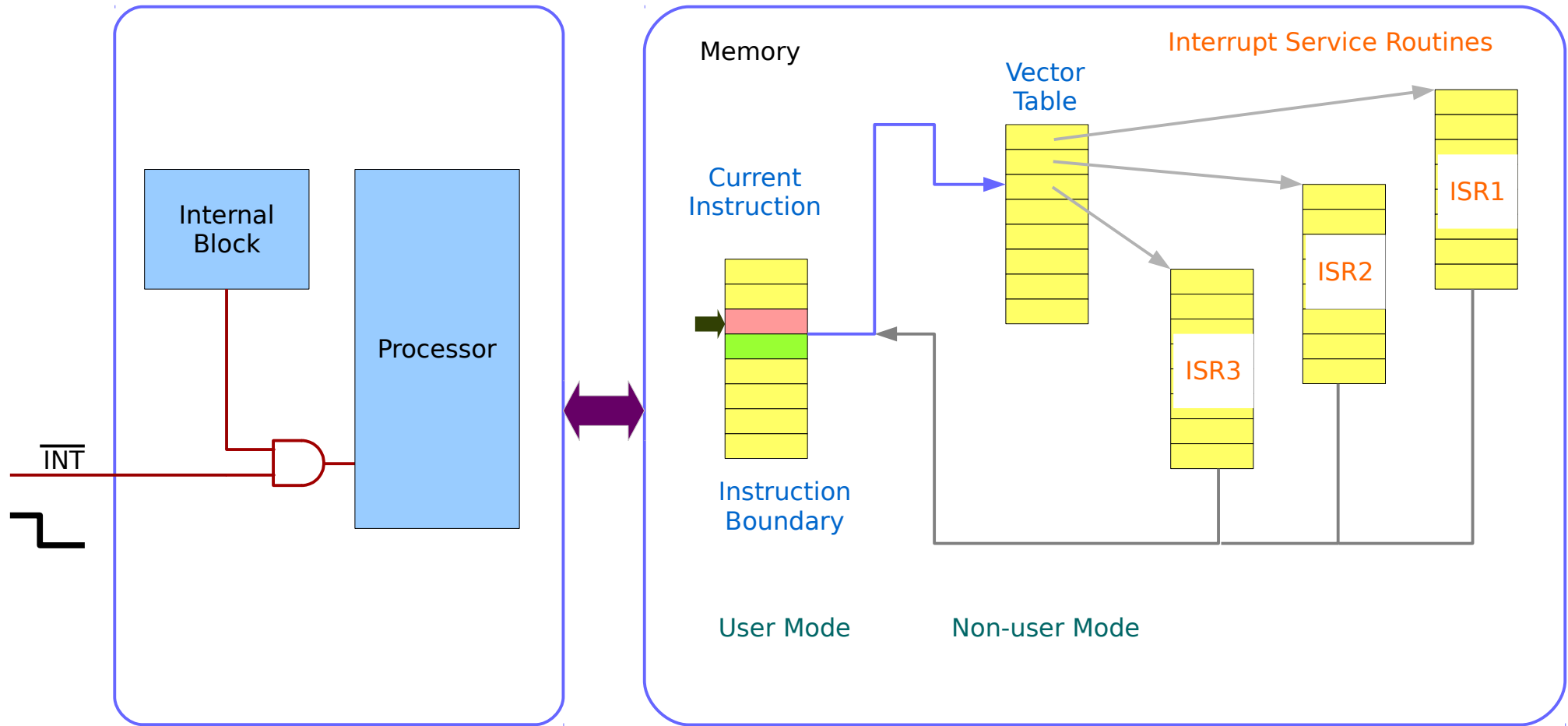
Copyright (c) 2010-2016 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice.

Interrupt Overview



Interrupt Overview

An event that requires the CPU to stop the current program execution and perform some service related to the event.

ISR : Interrupt Service Routine / Interrupt Handler

Coordinates I/O activities and prevent the CPU from continuous checking events

Performs Time critical applications

Interrupt Overview

Interrupt Vector

the starting address of an interrupt service routine

Interrupt Vector Table

Table storing several interrupt vectors

Each entry represents an address of an ISR

Interrupt Enable Bit

Processor's bit

Used to selectively enable a device

If the enable bit is 0, processor ignores the interrupt request

Interrupt Flag Bit

Devices' bit

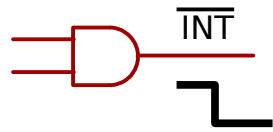
Used to get the processor's attention for it's service

Interrupt Overview

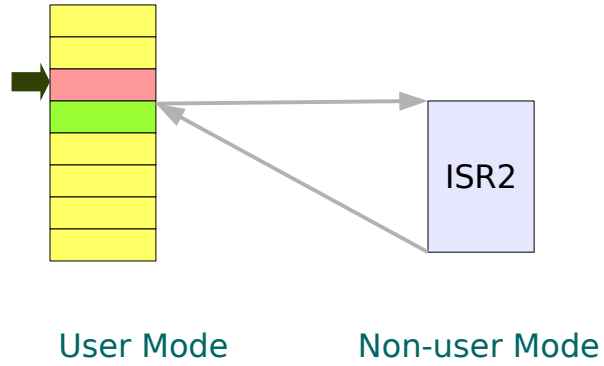
1. The device sets its flag bit
2. Upon detecting a flag is set, the microprocessor triggers an interrupt if the enable bit is also set
3. The processor status is saved automatically on the stack.
4. The Processor searches the appropriate interrupt vector
5. The processor jumps to the ISR.
6. At completing ISR, the special return instruction must be used to restore the saved processor status

HW & SW Interrupt

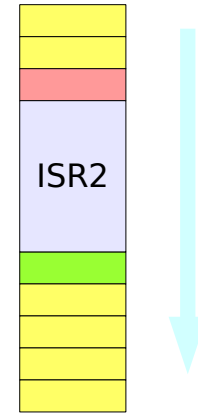
Asynchronous



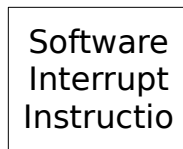
Hardware Interrupt



≡

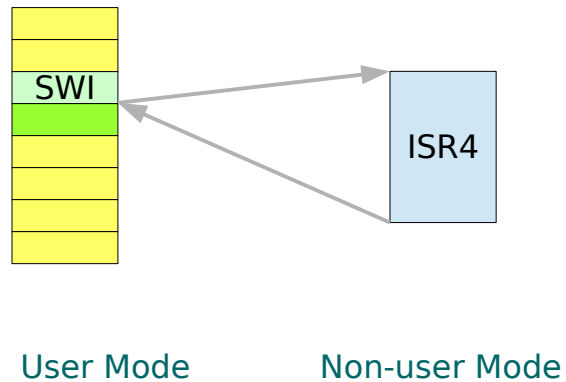


Synchronous

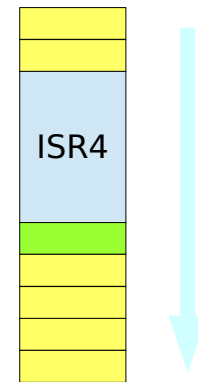


INT 0x80...

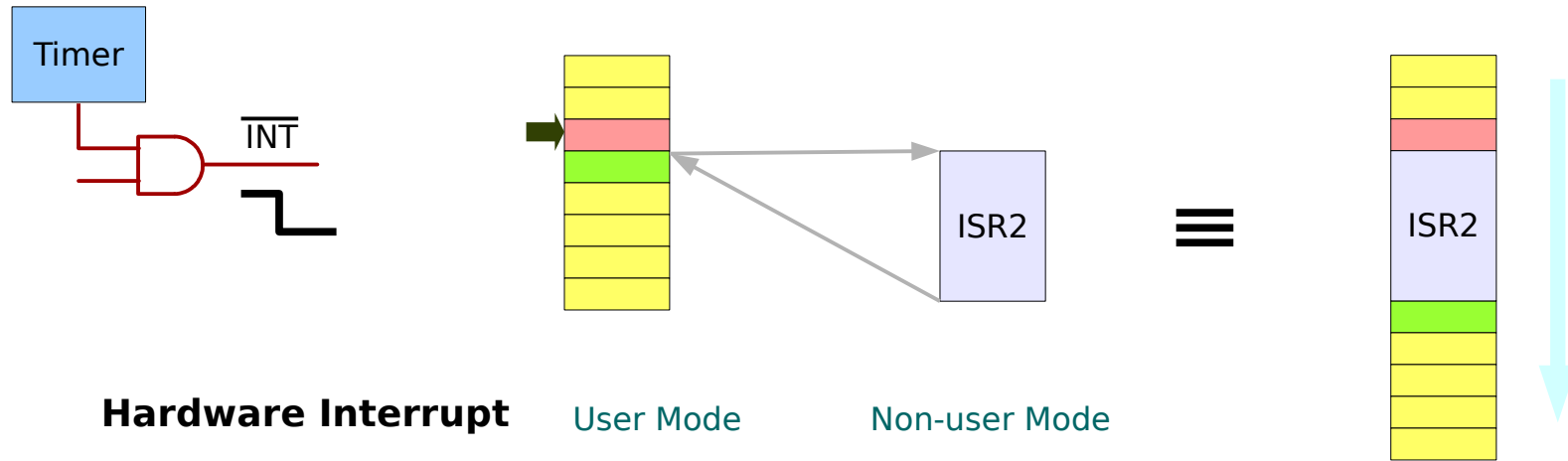
Software Interrupt



≡

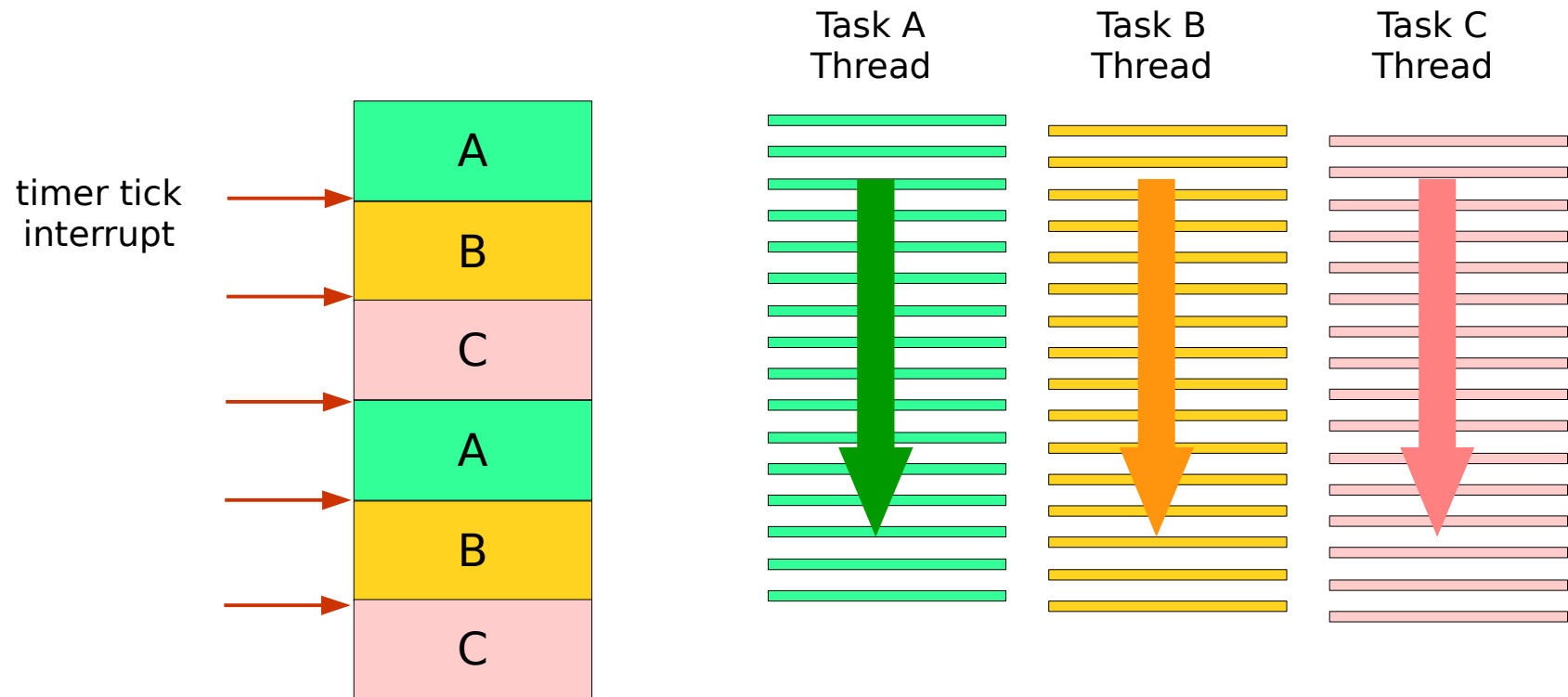


Timer & Interrupt



Time Sharing

- **Time Slice**
- **Round Robin Scheduling**
- **Context Switching**



References

- [1] http://wiki.osdev.org/ARM_RaspberryPi_Tutorial_C
- [2] <http://blog.bobuhiro11.net/2014/01-13-baremetal.html>
- [3] <http://www.valvers.com/open-software/raspberry-pi/>
- [4] <https://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/os/downloads.html>