Introduction

Operating Systems

Text/Reference Books

- **Remzi: Operating Systems: Three Easy Pieces (http://pages.cs.wisc.edu/~remzi/OSTEP/)**
- **Thomas Anderson, Michael Dahlin: Operating Systems: Vol. I-IV, Recursive Books**
- ***Silberschatz,Galvin: Operating System Concepts, Addison Wesley.***
- ***Tanenbaum, Modern Operating Systems, Prentice Hall.***
- ***W. Stallings, Operating Systems, Prentice Hall.***
- ***Russinovich et al: Windows Internals: Part I/II, 6th, 7th editions***
- Tanenbaum: Operating Systems: Design and Implementation. PHI.
- Bach, Design of Unix O/S.

Quiz I

- Would $01011011_2+01000110_2$ generate carry for 8-bit arithmetic?
- What would be sign bit for above computation?
- How much memory is addressable by
  - 16-bit segment ?
  - 32-bit segment ?
  - 64-bit segment ?
- If segment register always points to an address divisible by 8, which of the following is an invalid address?

A) 0x4580 B) 0x3BC8 C) 0x80C0 D) 0x123A

Quiz I - Solution

- Would 01011011_2+01000110_2 generate carry for 8-bit arithmetic?

$$\begin{array}{r} 01011011_2 \\ 01000110_2 \\ \hline 10100001_2 \end{array}$$

Result is also 8-bit, no carry is generated

Quiz I - Solution

- What would be sign bit for above computation?

01011011_2

01000110_2

$\mathbf{1}0100001_2$

MSB is 1, sign bit is set.

Quiz I - Solution

- How much memory is addressable by
  - 16-bit segment ?
  - 32-bit segment ?
  - 64-bit segment ?

2^{16}: 2^{16} bytes = $2^{10}2^{6}$ bytes = 2^{6} KB = 64 KB

2^{32}: 2^{32} bytes = $2^{30}2^{2}$ bytes = 2^{2} GB = 4 GB

2^{64}: 2^{64} bytes = $2^{60}2^{4}$ bytes = 2^{4} EB = 16 EB

2^{10}: Kilo, 2^{20}: Mega, 2^{30}: Giga, 2^{40}: Tera, 2^{50}: Peta, 2^{60}: Exa

Quiz I - Solution

- If segment register always points to an address divisible by 8, which of the following is an invalid address?

A) 0x4580 B) 0x3BC8 C) 0x80C0 D) 0x123A

Multiplying a binary number by 2 ≡ Left shifting by one ≡ setting LSB to zero

Multiplying a binary number by 8 ≡ Last three LSBs are zero.

So LSD (least significant digit) is 0x0 (0000_2) or 0x8 (1000_2)

Option D) is invalid, last hex digit is A (1010_2)

Concurrent v/s Simultaneous

- Concurrent refers to the ability of multiple tasks or processes to run at the same time, even if they are not actually running at the same instant. This is possible because the tasks or processes can share the same resources, such as CPU, memory, and storage.

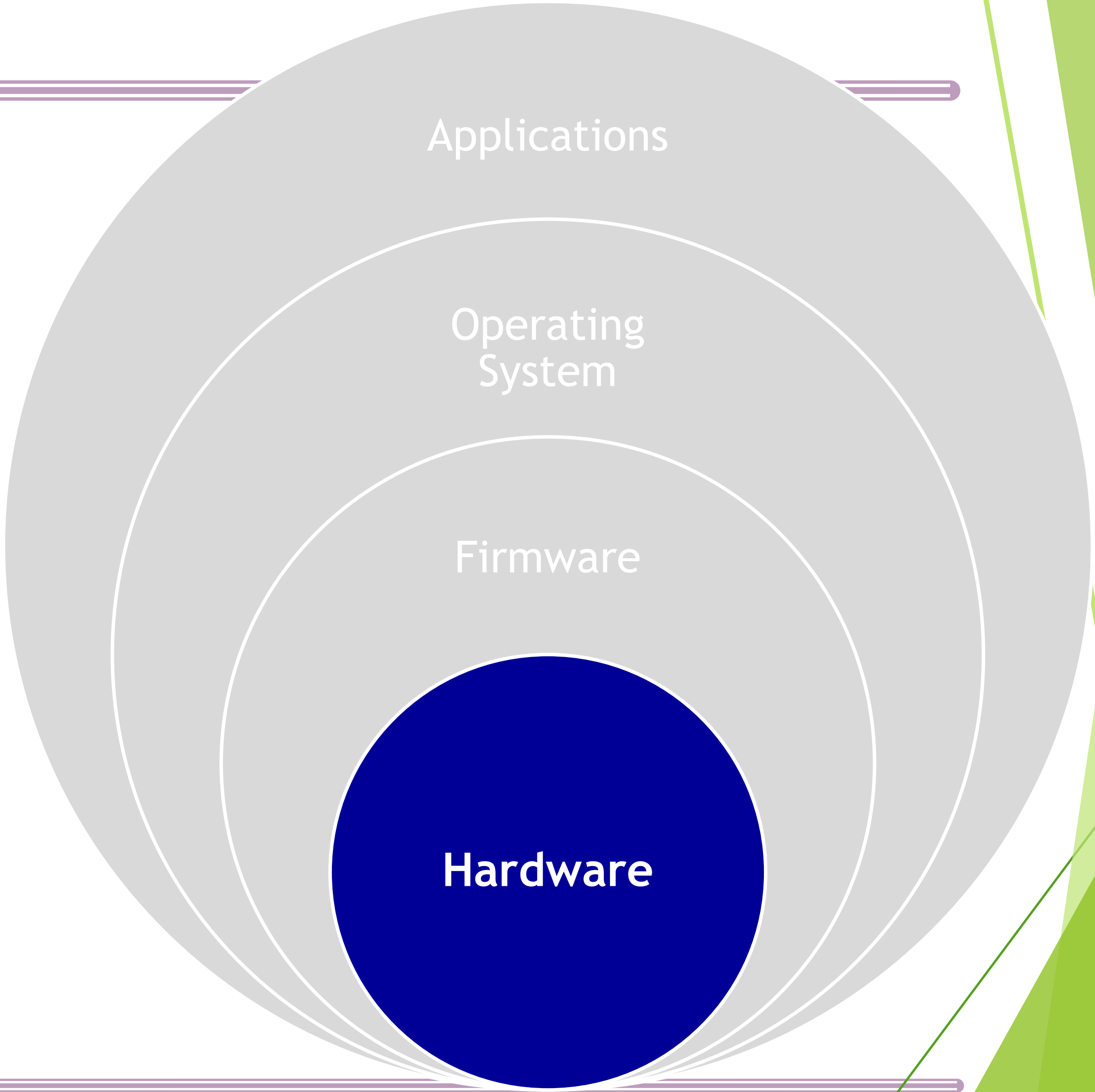
- Simultaneous refers to the ability of multiple tasks or processes to run at the same instant. This is possible if each task or process has its own dedicated resources.

System Abstraction

- Most systems are designed with a layered architecture
- Hardware: Innermost layer is hardware or bare machine
- Firmware: software installed on a ROM so that it can not be modified
- Hypervisor: to support virtualization.
- Operating System (which itself follows a layered architecture)
  - Hardware abstraction layer: device drivers
  - Kernel: Resource management (CPU scheduler, VMM, File manager etc.)
  - Library routines and utilities
- Outermost layer consists of application software and user programs

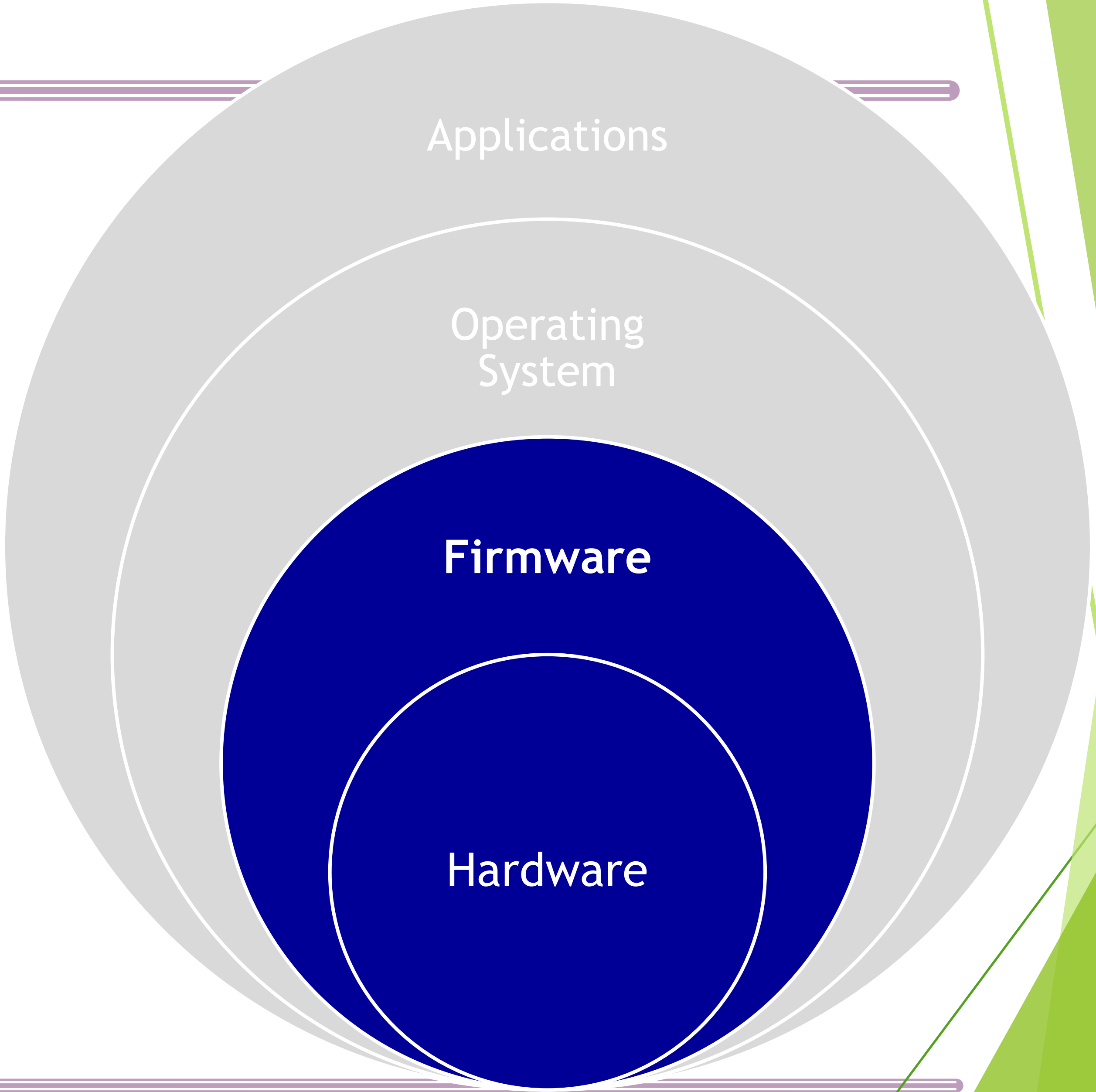
Abstraction

- Hardware:
  - CPU cores,
  - Memory cores,
  - IO processors or device controllers
  - A system of buses connecting these
  - All machine level instructions can be executed.
- Firmware
- Operating System
- Applications



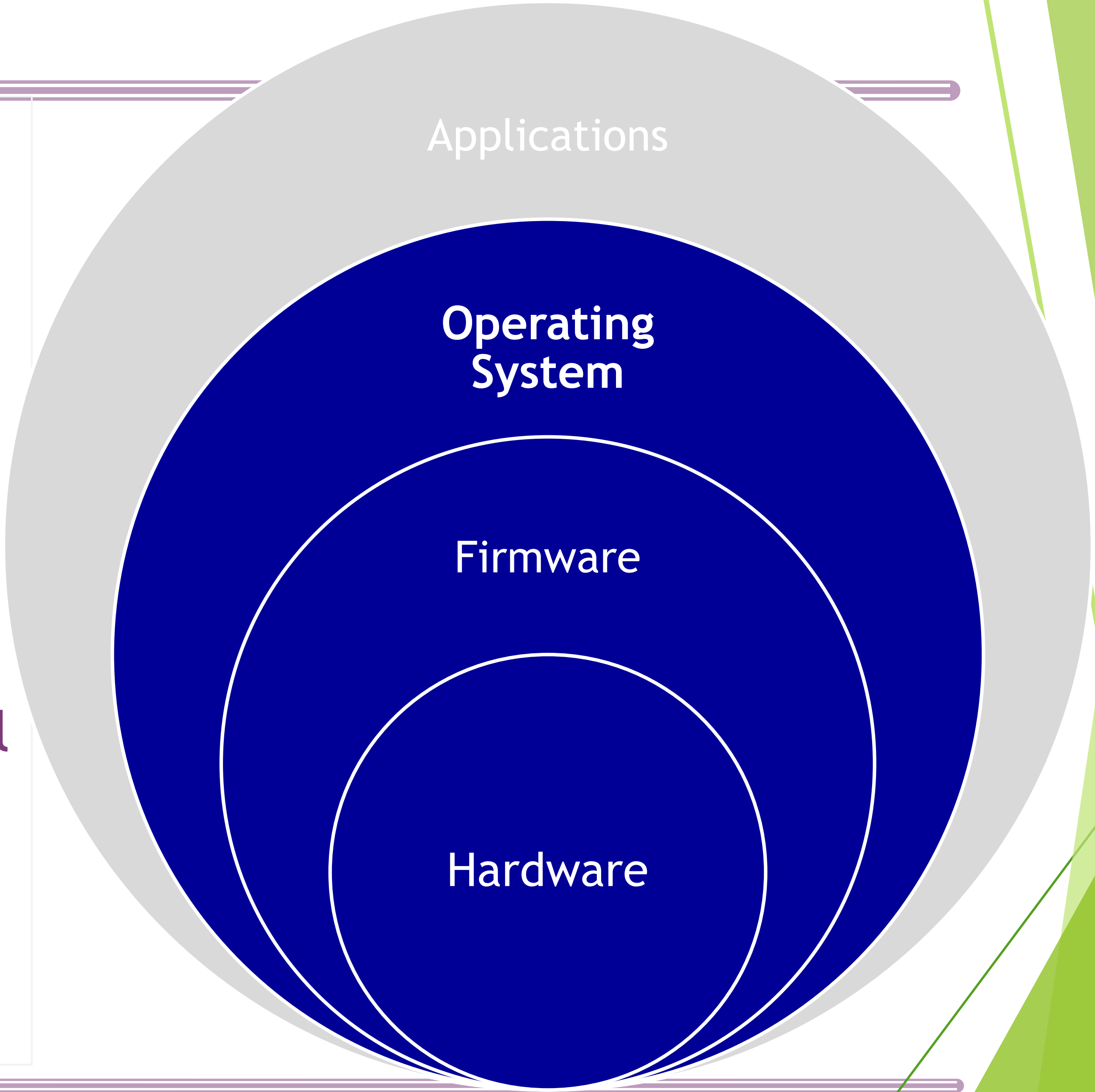
Abstraction

- Hardware:
- Firmware (BIOS)
  - Basic Input Output System
  - Executes POST (power on self test)
  - Checks if peripherals are connected
  - Checks if all memory locations are functioning well
  - Handlers for interrupts 13H / 11H
- Loads MBR (master boot record) from the hard disk to main memory and transfers control to it
- Operating System
- Applications



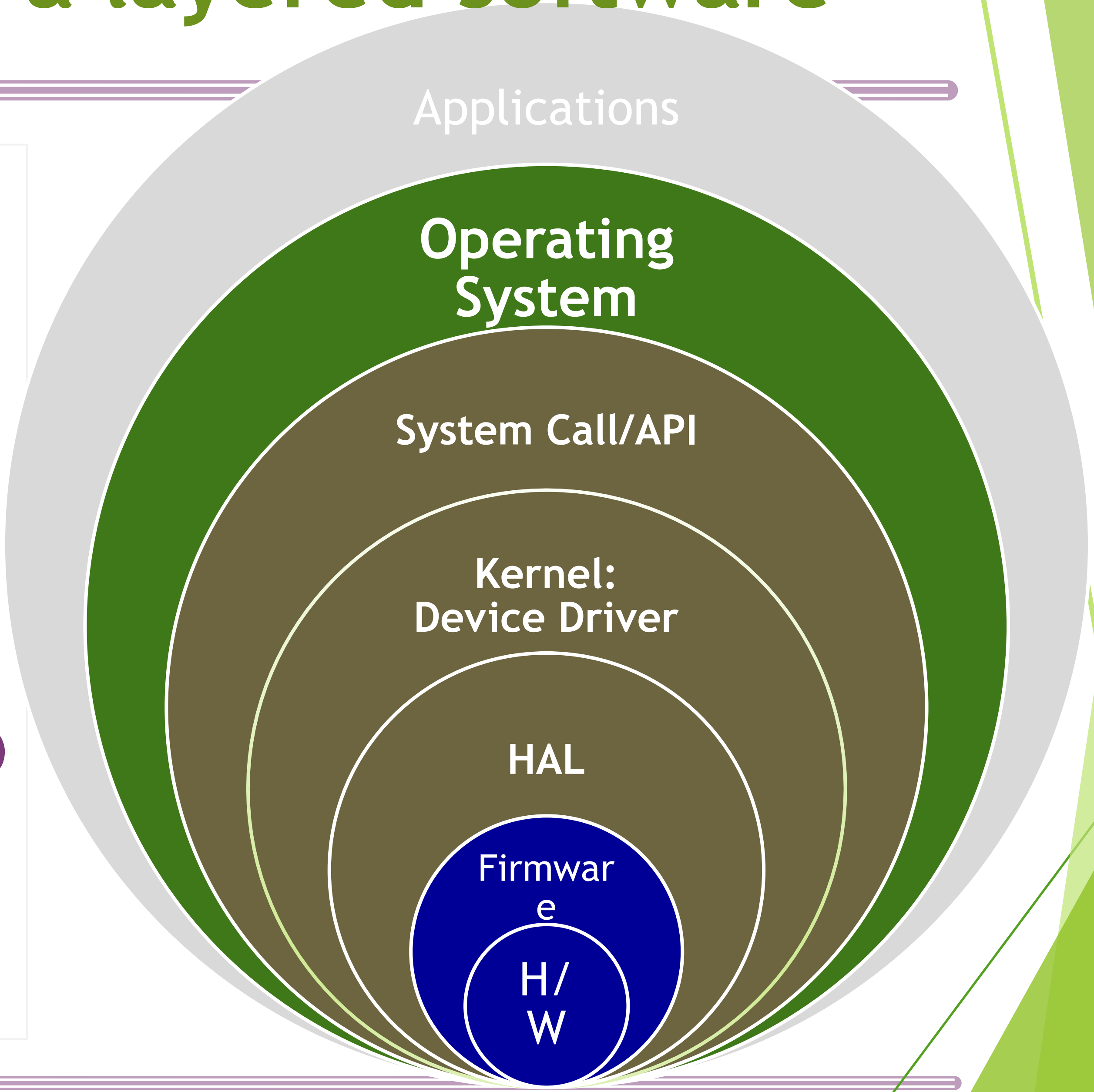
Abstraction

- Hardware
- Firmware
- Operating System
  - Interface between hardware and user
  - Complex software with millions of lines of code
  - Essential part is called kernel
  - Software: collection of many software/utilities?
- Applications



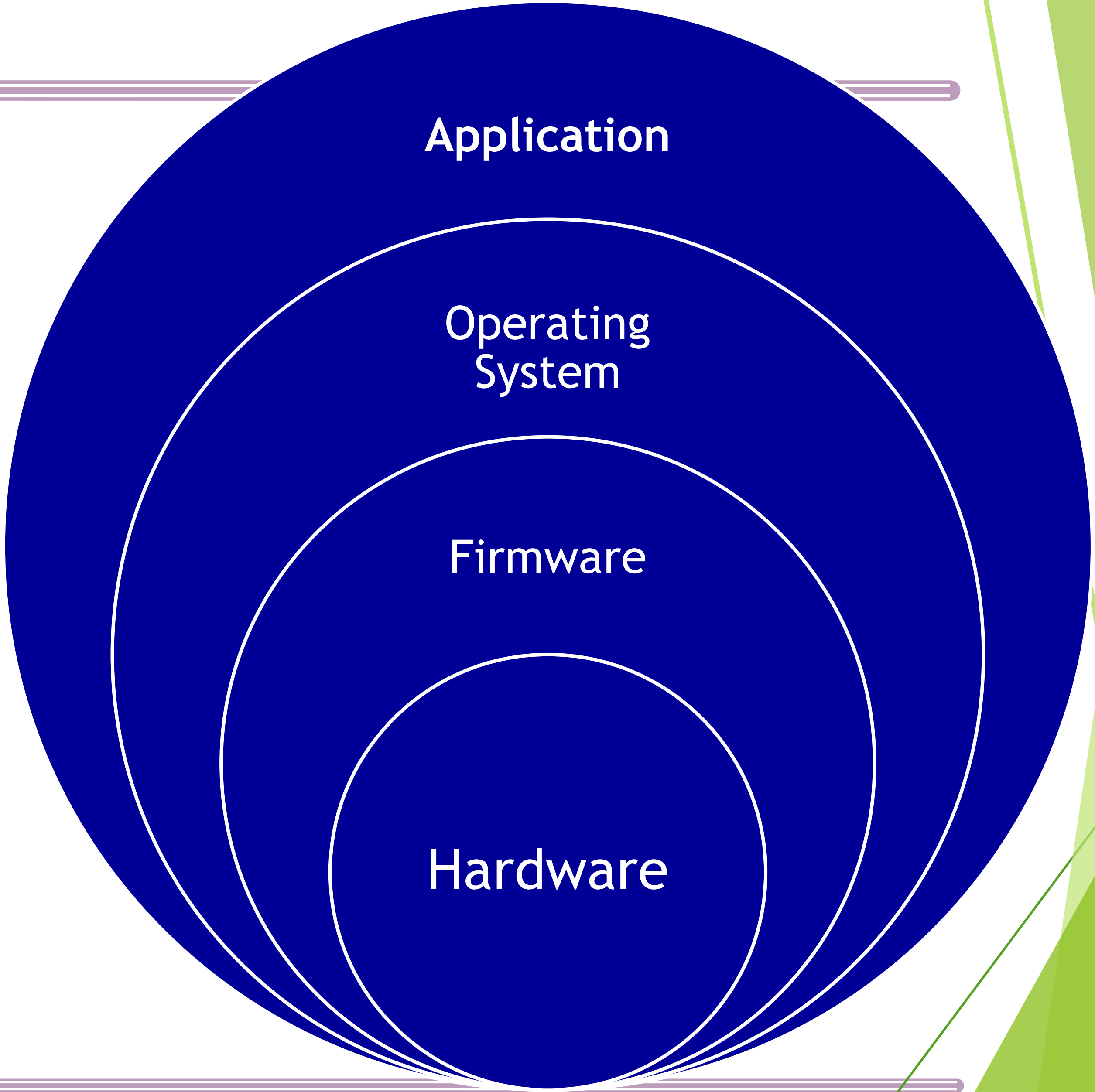
Abstraction: OS is a layered software

- Hardware
- Firmware
- Operating System
  - Interface between hardware and user
  - Complex software with millions of lines of code
  - Essential part is called kernel
  - HAL (Hardware Abstraction Layer)
  - Device Drivers (Third Party)
  - System Calls
- Applications



Abstraction

- Hardware
- Firmware
- Operating System
- Applications



Virtualization

- Virtualization (implemented through software) enables the hardware resources of a single computer—processors, memory, storage and more—to be divided into multiple virtual computers, called virtual machines (VMs). - IBM
- Facilitate sharing of physical resources among various users (process is also an user) to
  - Improve utilization of physical resources
  - Allow multiple users to use the system concurrently (not simultaneously as one user shall be allowed to access the resource at a given time)
  - Concurrently means that each

CPU Virtualization

- CPU is shared between processes.
- Each process has an illusion that it has control of CPU.
- Time sharing – sharing CPU time is the way to achieve it.
- Timer interrupt ensures that CPU is not monopolized by any process.
- Time slice needs to be small so that CPU sharing is not visible to any process.
- Concurrent execution
    - Sharing
    - Change or context switch fast enough to be noticeable
    - Process execution (though in slices) should be same as if there were no interruptions.

Memory Virtualization

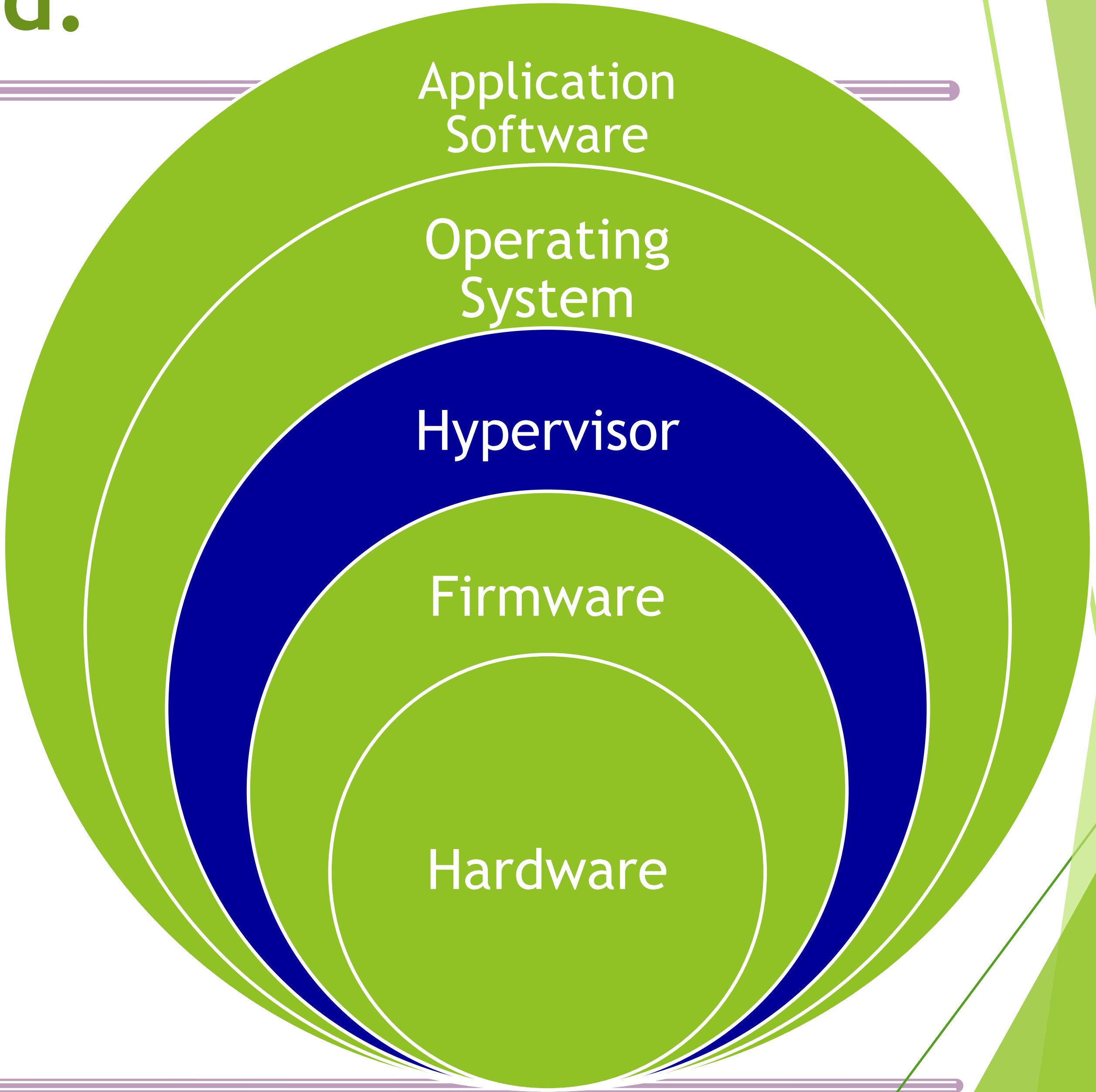
- Memory is shared between processes.
- No process is allowed to access address space of other processes
  - Memory corruption
  - Code and/or data of another process may be corrupted
  - Incorrect, inconsistent execution, abnormal termination of other process
- Space sharing – multiple processes can share memory but not the same region
- Base and limit registers used to ensure that address space violation do not take place.
- Same strategy shall be used for sharing storage – pen drive, SSD, hard disk.

Questions to ponder?

- How would virtualization be enacted for following resources?
  - Keyboard
  - Mouse
  - Display
  - Printer
  - Network Interface Card

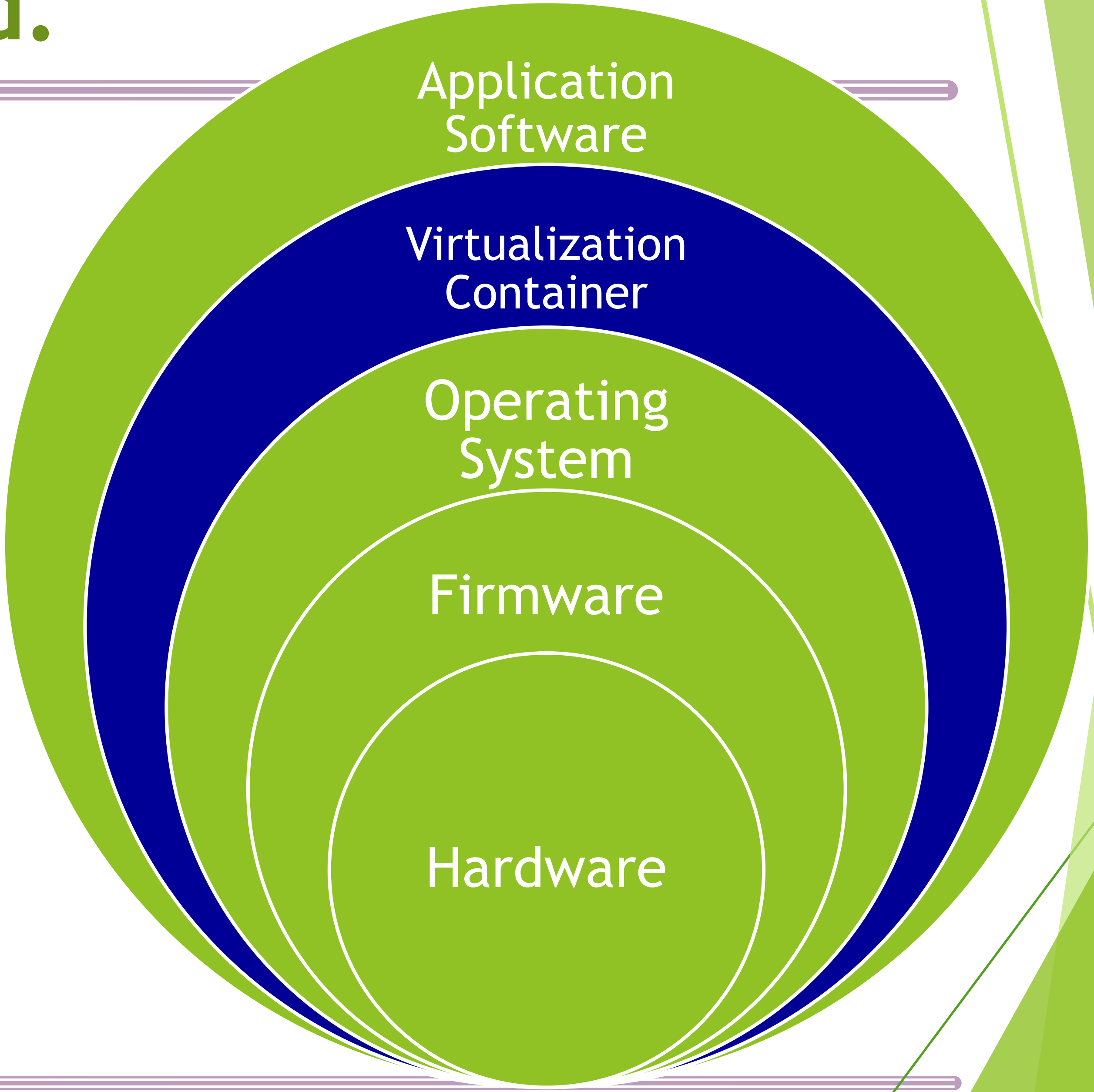
Abstraction Contd.

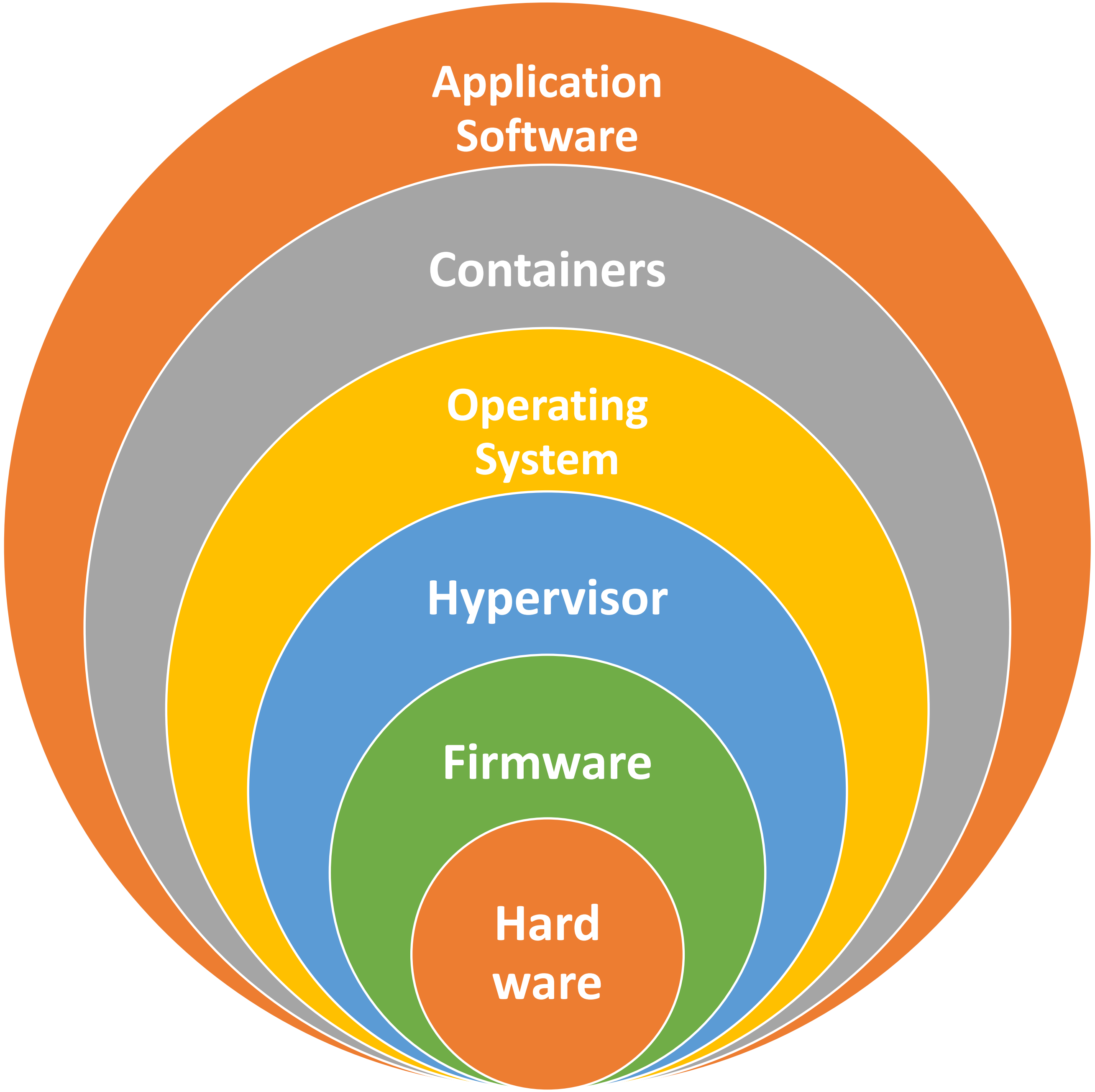
- Virtualization (support for multiple operating systems) in modern systems achieved
- **Type I Hypervisor (bare metal) that resides between OS and firmware**
- Type II: Application on host OS (Oracle VirtualBox)



Abstraction Contd.

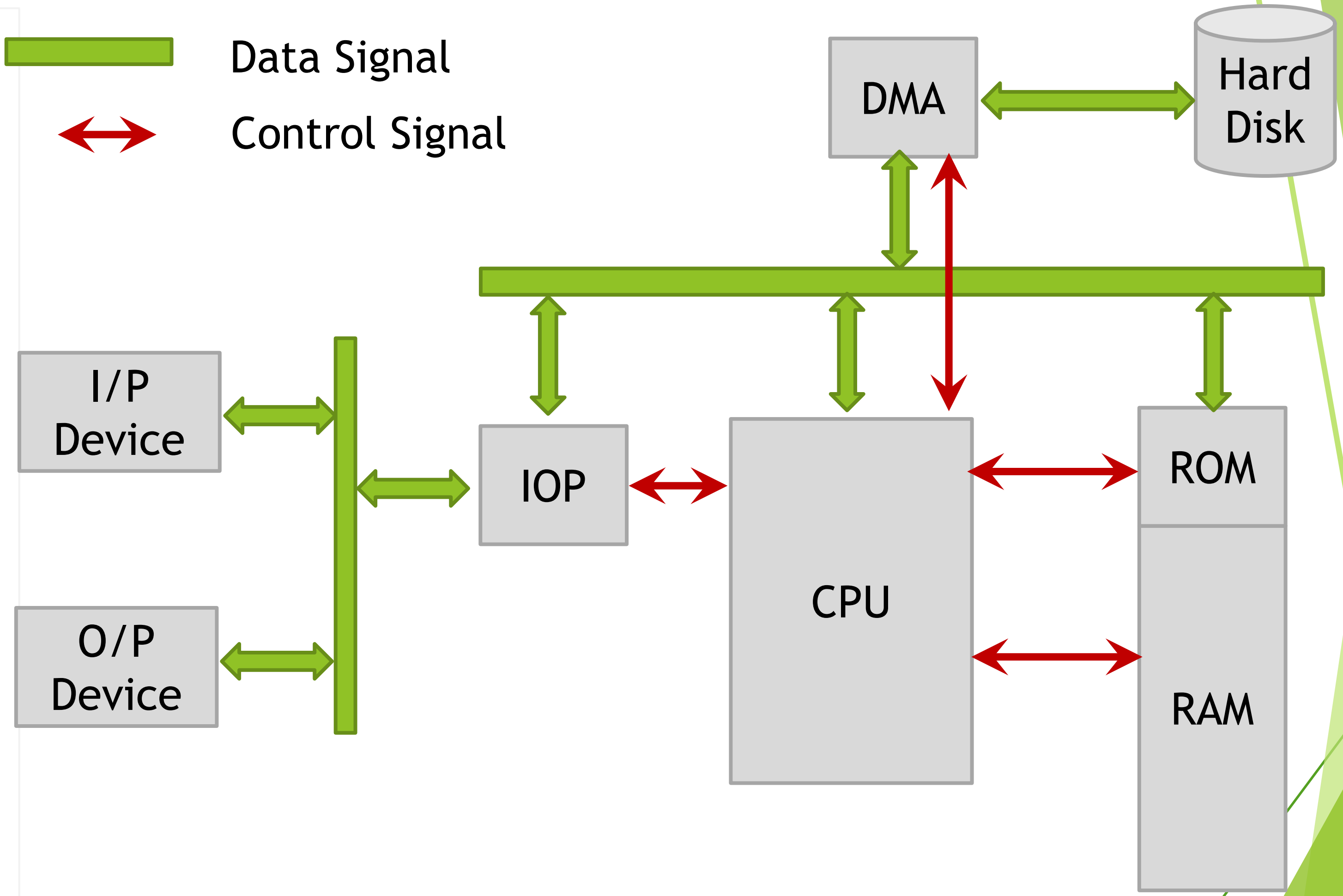
- Virtualization (support for multiple operating systems) in modern systems achieved
- Type I Hypervisor (bare metal) that resides between OS and firmware
- **Type II: Application on host OS (Oracle VirtualBox)**




Application Software
Containers
Operating System
Hypervisor
Firmware
Hard ware

Hardware Abstraction

- Processor can read data directly only from main memory i.e.
  - ROM: Read Only Memory, non-volatile (persistent)
  - RWM: more commonly called as RAM, volatile
  - SSD (solid state device): flash memory, non-volatile
- Data from secondary storage device (Hard Disk, CDROM, DVD, USB drive etc.) needs to be loaded into main memory.
- IOP: input output processor or IO controller is hardware for device-CPU interaction; also has buffer for data storage

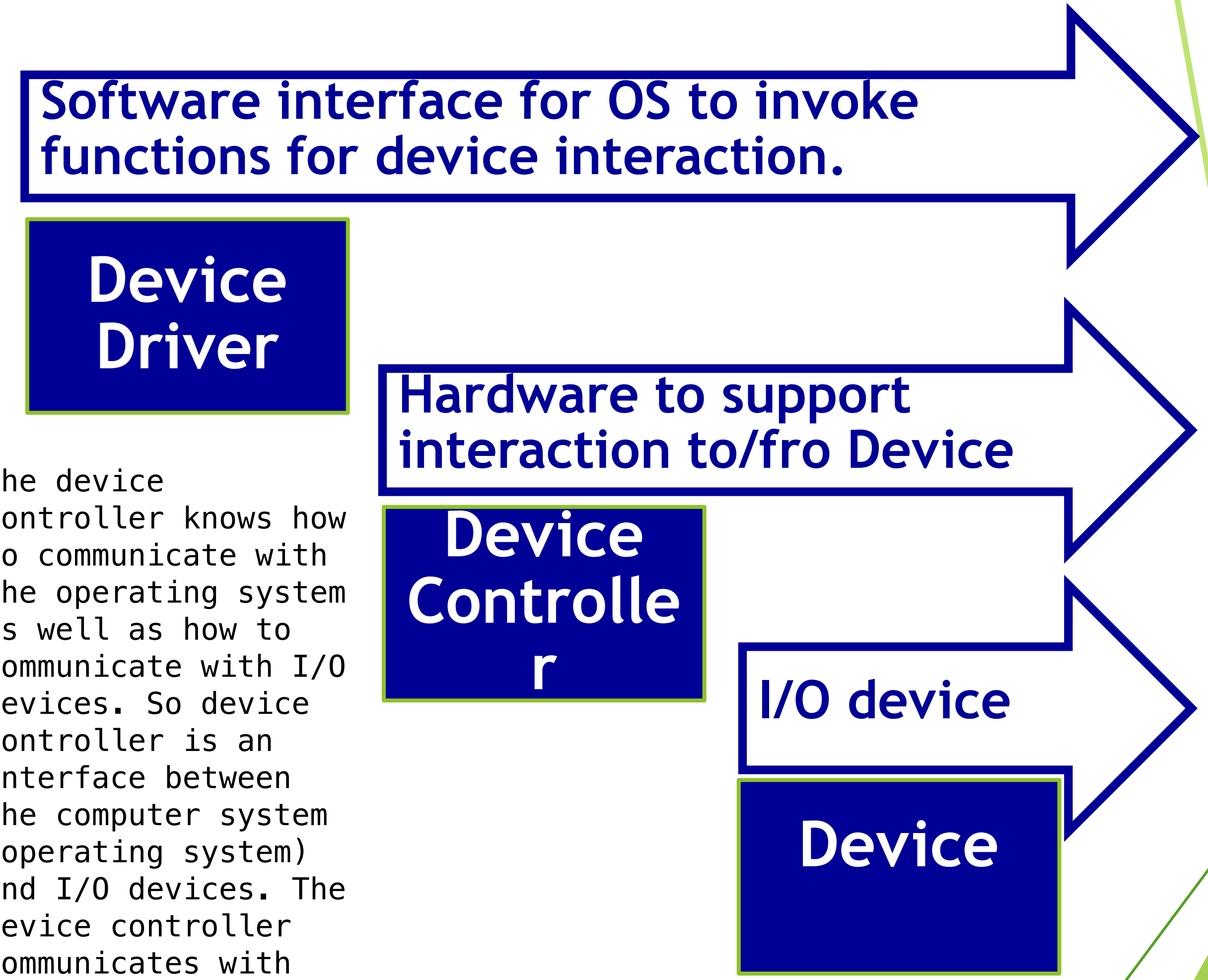


Hardware Abstraction

- ACC: accumulator, a special register where results are stored unless explicitly specified
- PC: program counter that stores address of current instruction
- IP: stores instruction under execution
- SP: stack pointer – top of the stack
- BP: base pointer
- SI: source index
- DI: destination index
- FLAG register: each bit is set/reset depending on outcome of the instruction just executed
  - Zero, Sign, Parity, Carry
- Segment registers:
  - CS: code segment
  - DS: data segment
  - SS: stack segment
  - ES: extra data segment

Device Abstraction

- Each device: minimum of three addresses
    - Status Register
    - Command Register
    - Data Register
- Device controller is hardware for managing data exchange between device and processor
- Device driver is software for managing device controller



Software interface for OS to invoke functions for device interaction.

Device Driver

Hardware to support interaction to/fro Device

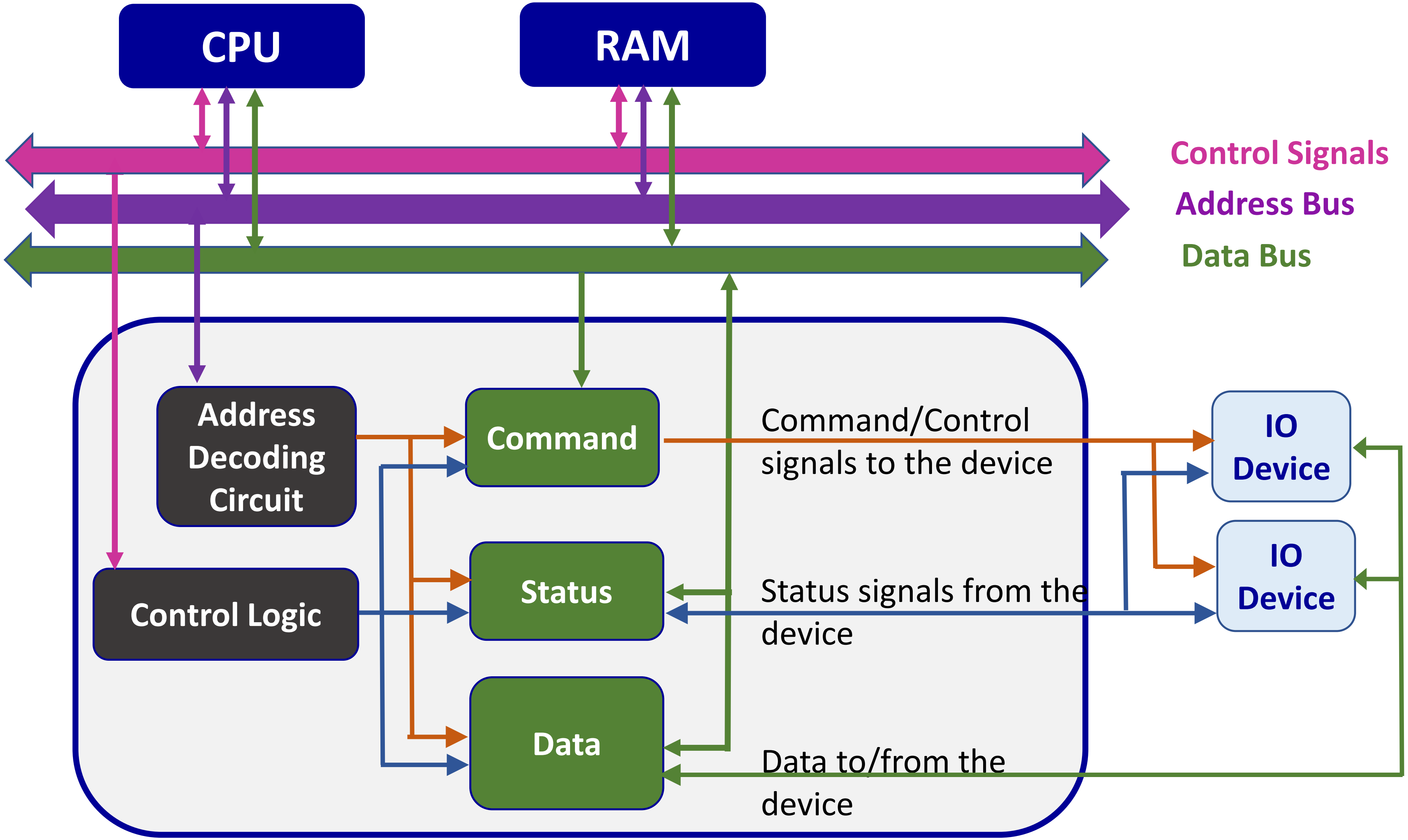
Device Controller

I/O device

Device

The device controller knows how to communicate with the operating system as well as how to communicate with I/O devices. So device controller is an interface between the computer system (operating system) and I/O devices. The device controller communicates with the system using the system bus.

I/O Controller

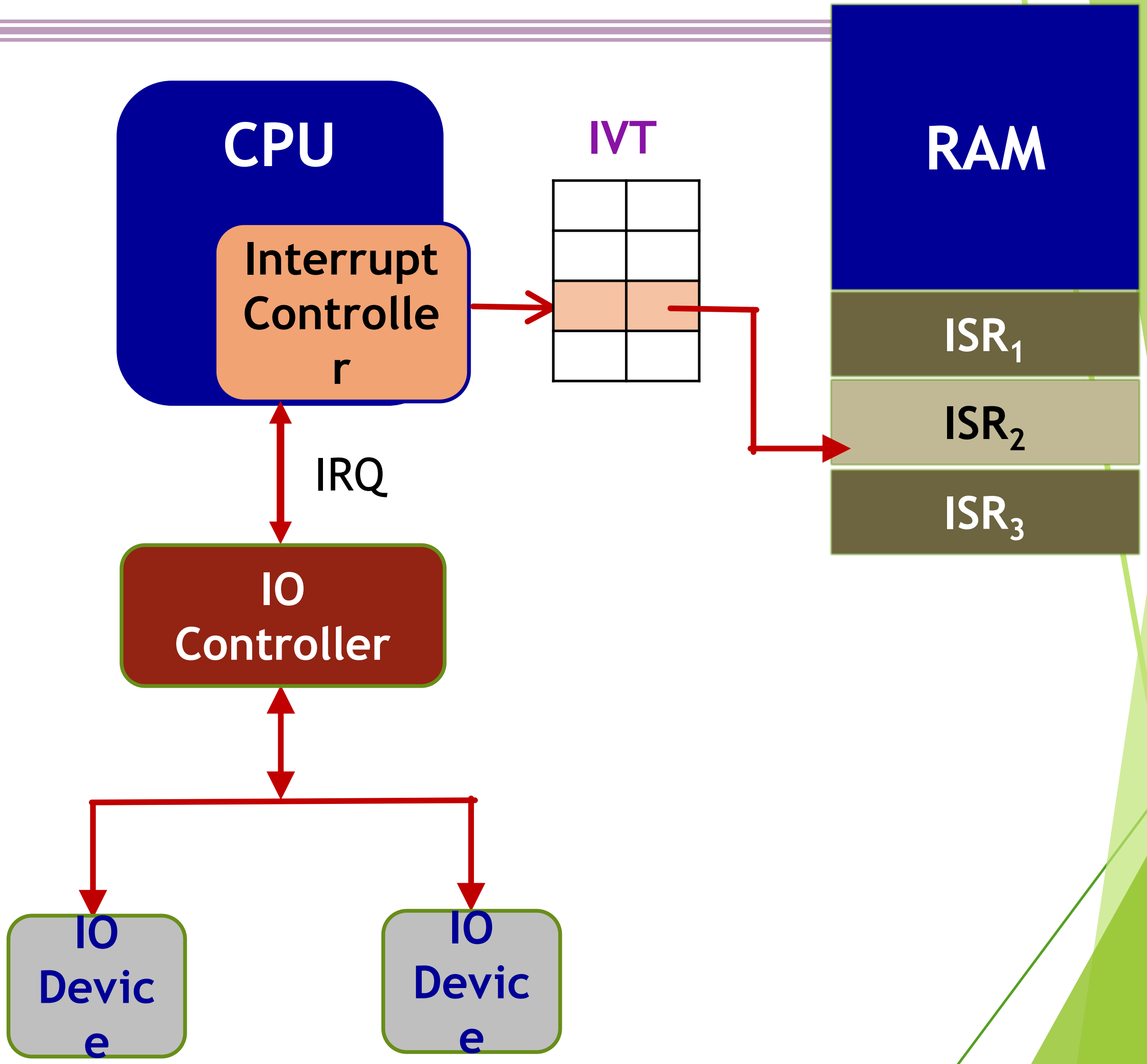


DMA

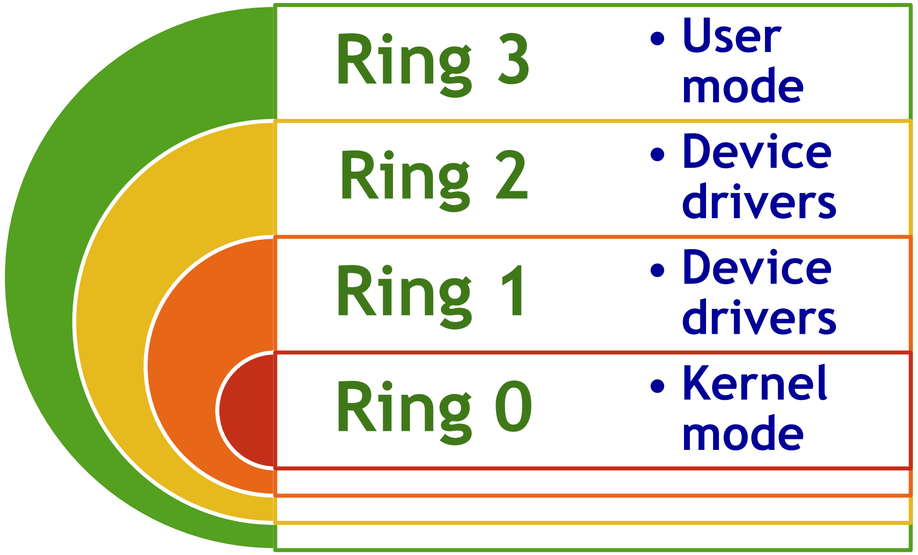
- DMA – Direct Memory Access
- Device controller to manage data transfers between hard disk and RAM
- Data transferred directly between disk and memory (through DMA) without any intervention of CPU
- Data and address bus under DMA's control when the transfer takes place
  - Generally done through cycle stealing (use bus when CPU not using it)

Interrupts

- Device sends IRQ (interrupt request) signal to interrupt controller
- Each request has a code
- Stores different requests from different devices
- ISR (interrupt service routine) invoked as per the code in the request
- Nested interrupt handling may or may not be supported



Privileged Mode



- Instruction Set divided partitioned as per privilege level (protection rings); a process can execute an instruction provided its priority/privilege level allows it
- x86 architecture supports four protection levels or rings
- Innermost ring or Ring 0 (Highest priority) - all instructions can be executed; also called kernel mode
- Outermost ring or Ring 3 (least priority) – only some instructions can be executed; also called user mode
- Most OS use only two levels – kernel mode (supervisor / privileged mode) and user mode

QUIZ 2

- Minimum number of address bits required for 2GB RAM ?
- How many hexadecimal digits in a 32-bit address?
- If CPU is shared between 10 processes and switches to another process after 0.5 ms in a round robin manner, each process has to wait for how much time between consecutive runs?
- Interface provided by OS to allow a user process to access hardware device is called ….
- Interrupts are generated by CPU or I/O devices?
- Where are Interrupt service routines loaded?