CS202 – Advanced Operating Systems

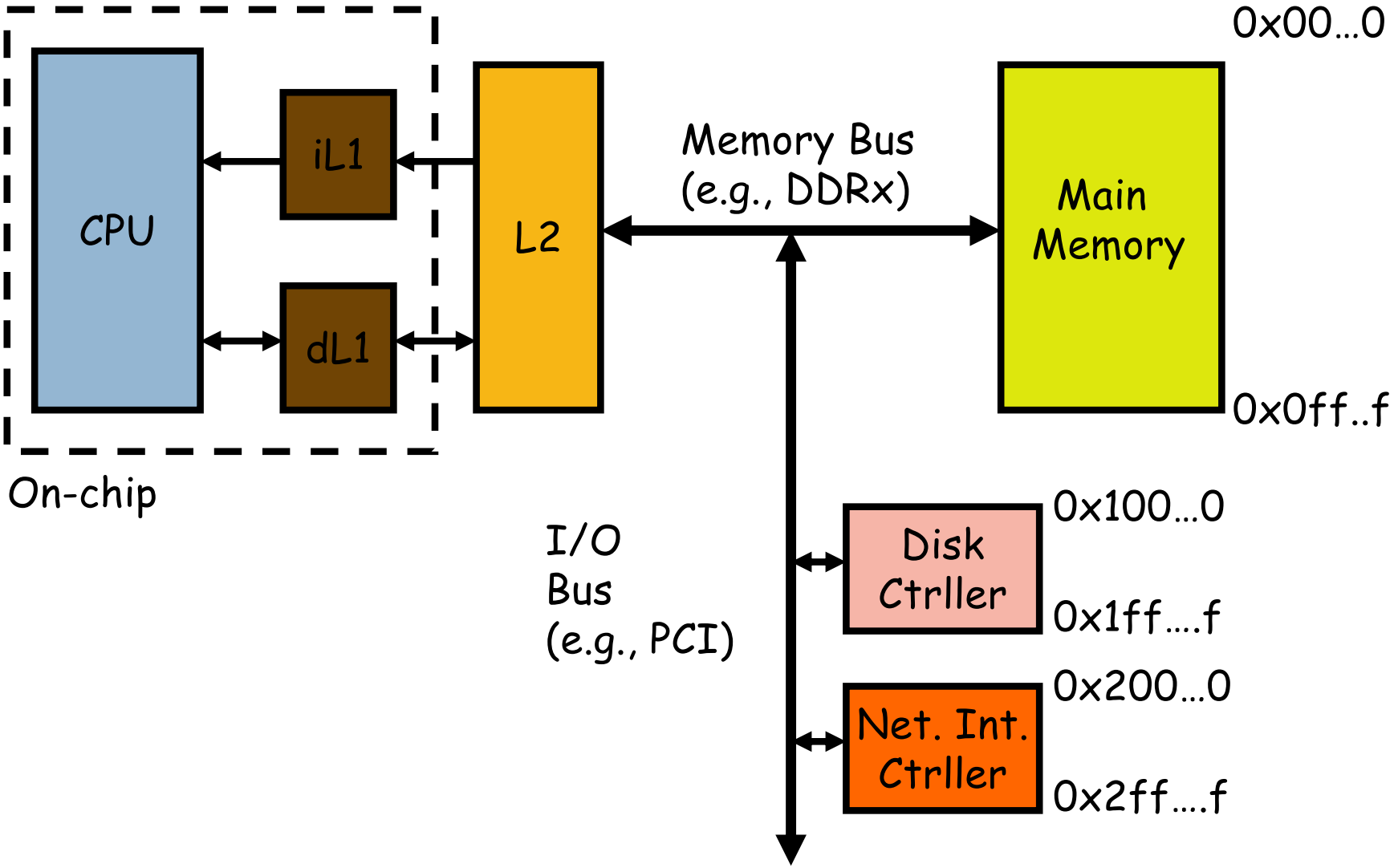
Input/Output

March 5, 2025

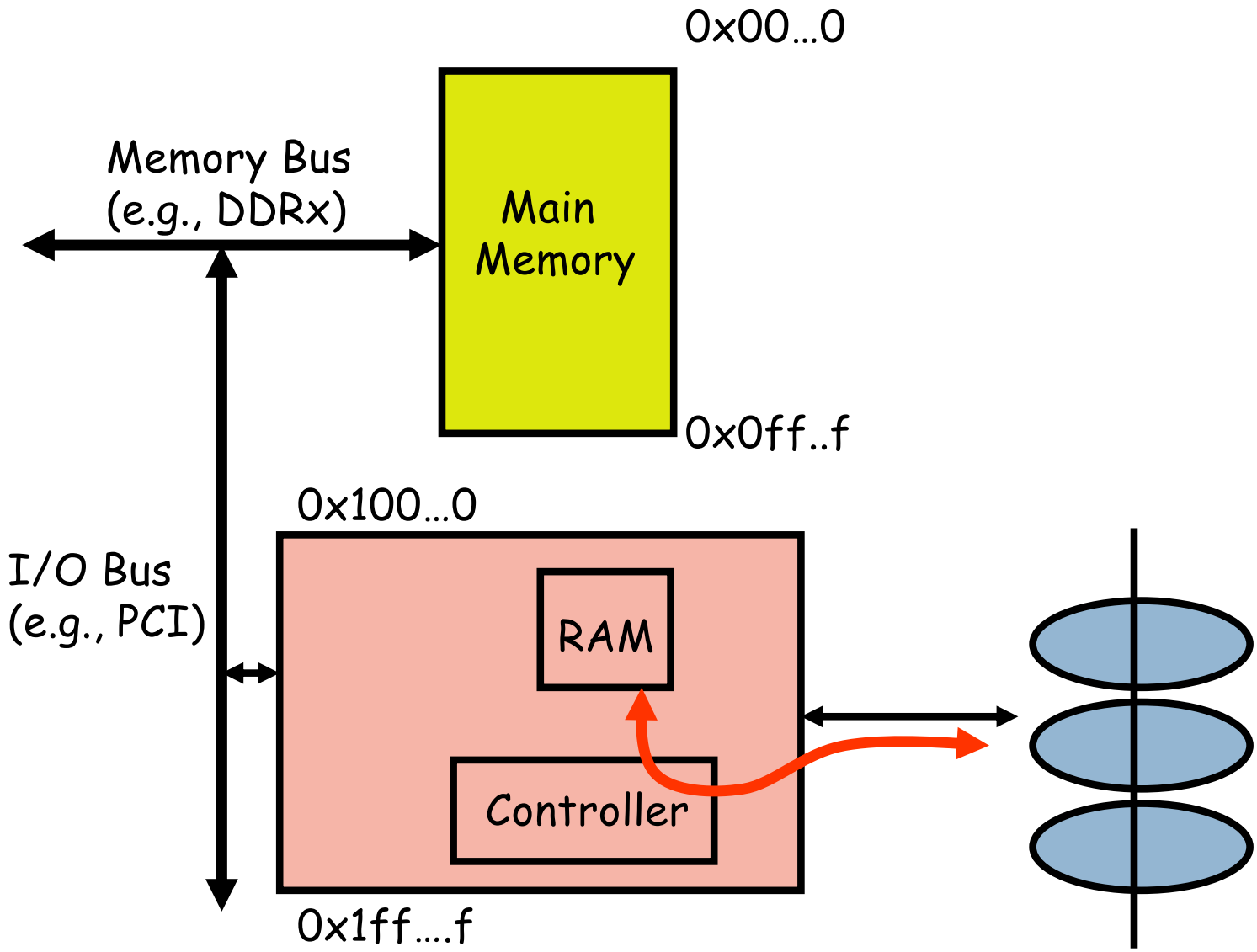
OS role in I/O

- Share the same device across different processes/users
- User does not see the details of how hardware works
- Device-independent interface to provide uniformity across devices.

I/O Peripherals



Consider a disk device …



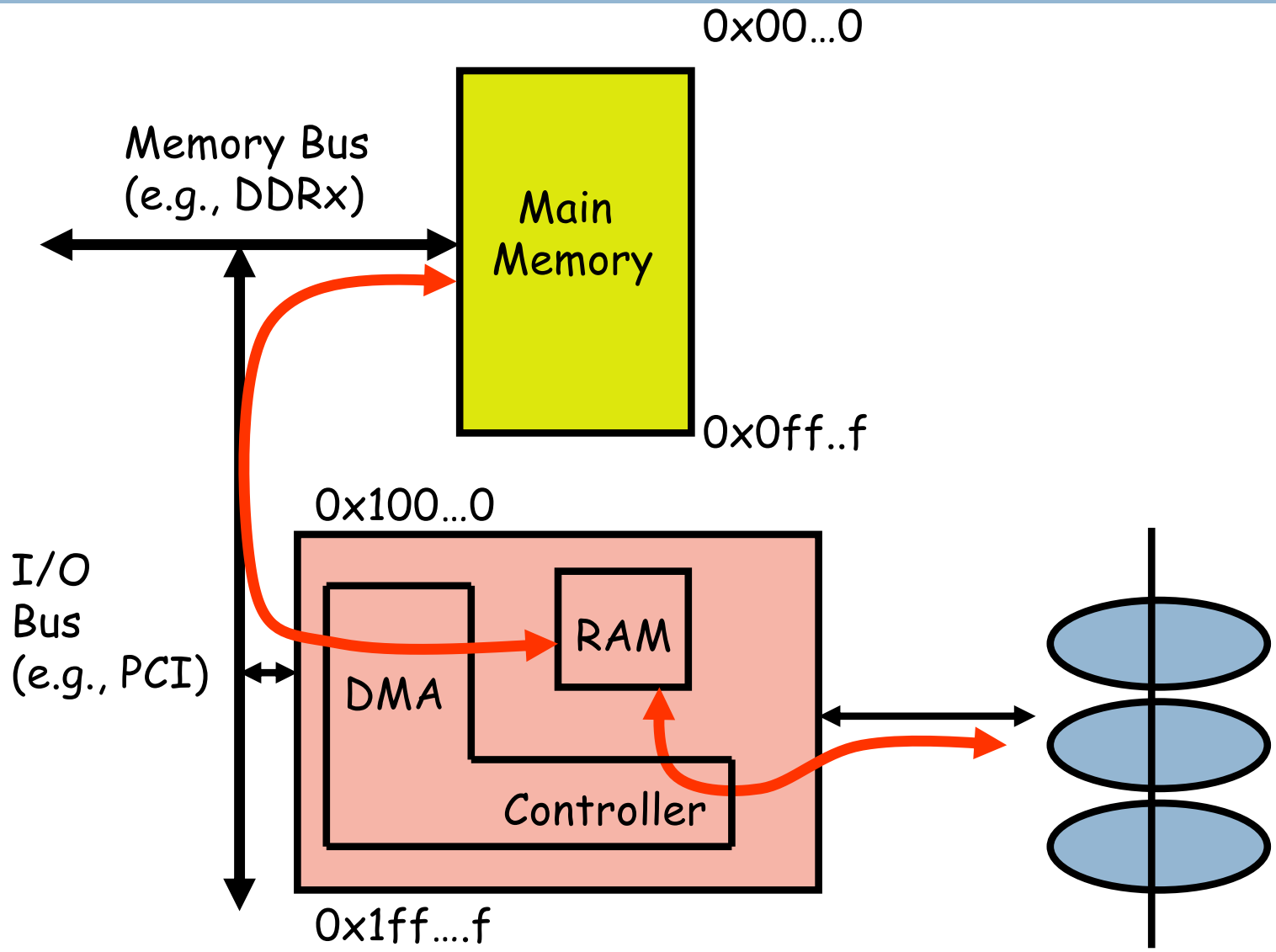
Reading a sector from disk

```
Store [Command_Reg], READ_COMMAND
Store [Track_Reg], Track #
Store [Sector_Reg], Sector #

/* Device starts operation */


L:   Load R, [Status_Reg]
     cmp R, 0
     jeq

/* Data now on memory of card */

For i = 1 to sectorsize
    Memtarget[i] = MemOnCard[i]
```

You don't want to do this! Instead, block/switch to other process and let an interrupt wake you up.

This is again a lot of overhead to ask the main CPU to do!

DMA engine to offload work of copying



```
Store [Command_Reg], READ_COMMAND
Store [Track_Reg], Track #
Store [Sector_Reg], Sector #
Store [Memory_Address_Reg], Address

/* Device starts operation */

P(disk_request);

/* Operation complete and data is
   now in required memory locations*/
```

Assuming an integrated DMA and disk ctrller.

Called when DMA raises interrupt after Completion of transfer

```
ISR() {
    V(disk_request);
}
```

Issues to consider

- What is purpose of RAM on card?
  - To address the speed mismatch between the bit stream coming from disk and the transfer to main memory.

Issues to consider (contd.)

- When we program the DMA engine with address of transfer (Store[Memory_Address_Reg], Address), is Address virtual or physical?
  - It has to be a physical address, since the addresses generated by the DMA do NOT go through the MMU (address translation).
  - But since it is the OS programming the DMA, the physical address is available and NOT a problem.
  - You do NOT want to give this option to user programs.
  - Also, the address needs to be “pinned” (cannot be evicted) in memory.

Conclusions

- Brief look at I/O support for operating systems
  - In particular Direct Memory Access (DMA)
- While CPUs can communicate with devices
  - It removes the ability to perform useful work
- DMA enables the OS to setup a device->memory op
  - Or memory->device op
- Can be performed between the device and memory
  - CPU can do other things and wait for an interrupt

Questions