OPERATING SYSTEMS

I/O Management, System Protection and Security

Kakoli Bora
Department of Computer Science

OPERATING SYSTEMS

I/O Systems - DMA

Kakoli Bora
Department of Computer Science

Slides Credits for all PPTs of this course

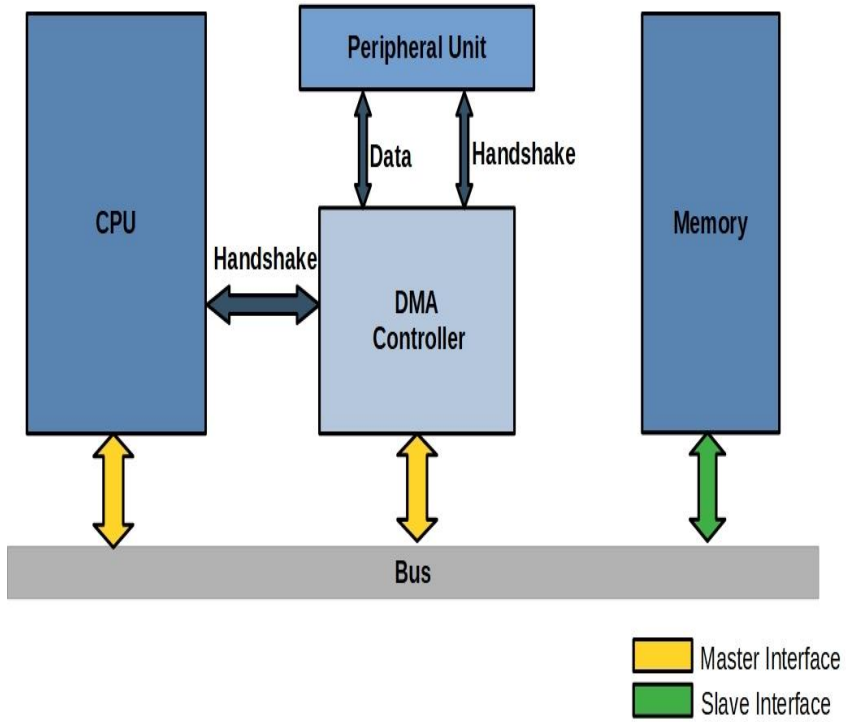
- The slides/diagrams in this course are an **adaptation**, **combination**, and **enhancement** of material from the following resources and persons:

1. Slides of Operating System Concepts, Abraham Silberschatz, Peter Baer Galvin, Greg Gagne - 9th edition 2013 and some slides from 10th edition 2018
2. Some conceptual text and diagram from Operating Systems - Internals and Design Principles, William Stallings, 9th edition 2018
3. Some presentation transcripts from A. Frank – P. Weisberg
4. Some conceptual text from Operating Systems: Three Easy Pieces, Remzi Arpaci-Dusseau, Andrea Arpaci Dusseau

OPERATING SYSTEMS

Direct Memory Access

- **Direct Memory Access** (DMA) transfers the block of data between the *memory* and *peripheral devices* of the system, **without the participation** of the **processor**.
  - The unit that controls the activity of accessing memory directly is called a **DMA controller**.
- Used to avoid **programmed I/O** (one byte at a time) for large data movement
- Bypasses CPU to transfer data directly between I/O device and memory
- A simple DMA controller is a standard component in all modern computers, from smartphones to mainframes.
- Handshaking between DMA controller and device controller is performed via a pair of wires called **DMA-request** and **DMA-acknowledge**



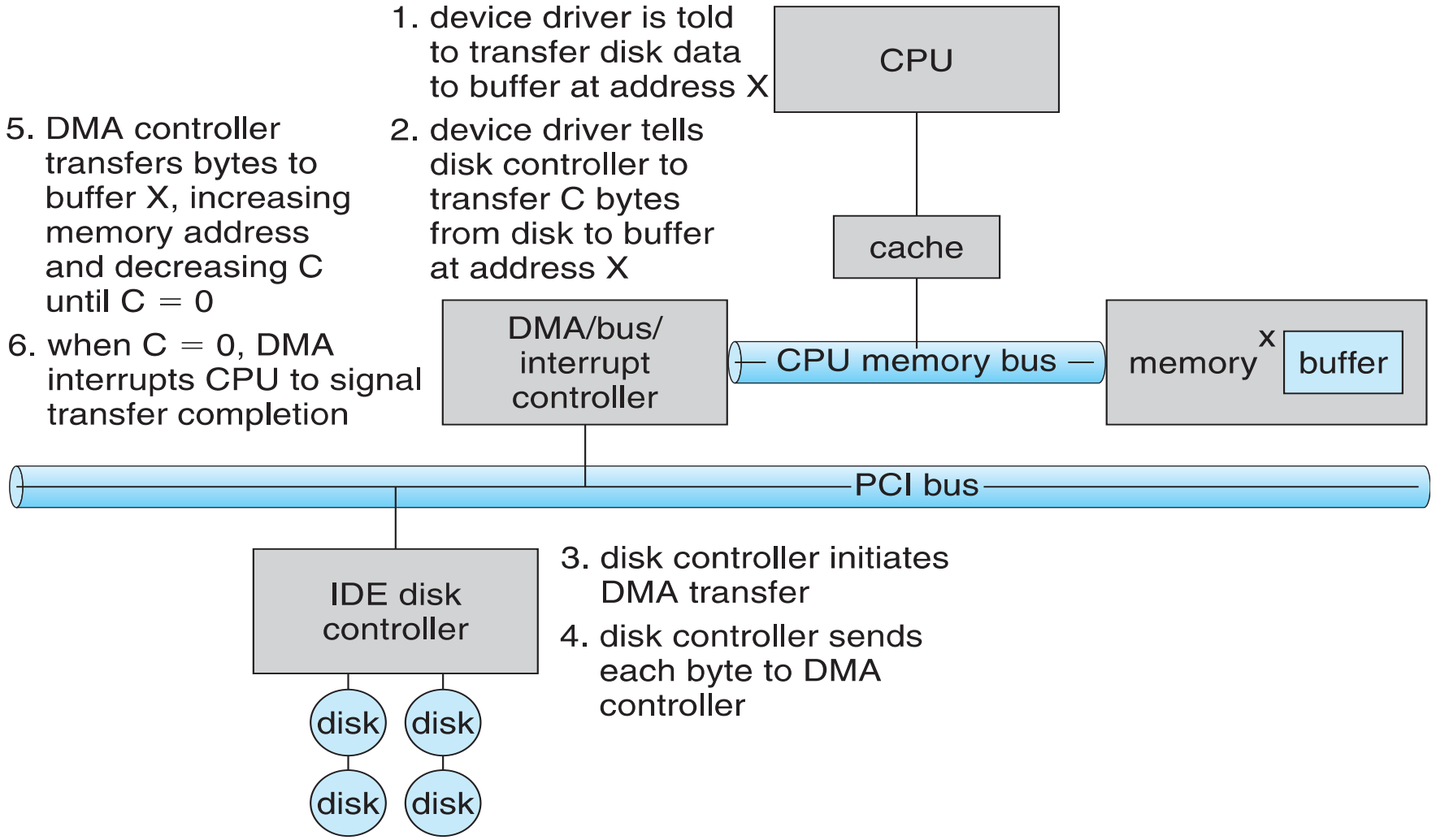
CPU and DMA controller have master interfaces to initiate transfers

Direct Memory Access (Cont.)

- OS writes DMA command block into memory
  - Source and destination addresses
  - Read or write mode
  - Count of bytes
  - CPU writes location of command block to DMA controller
  - Bus mastering of DMA controller – seizes the memory bus from CPU
    - **Cycle stealing** from CPU (i.e CPU is momentarily prevented from accessing main memory, but it can still access data items in its primary and secondary caches) but still much more efficient
  - When done, interrupts to signal completion
- Version that is aware of virtual addresses can be even more efficient – **DVMA (Direct Virtual Memory Access)**

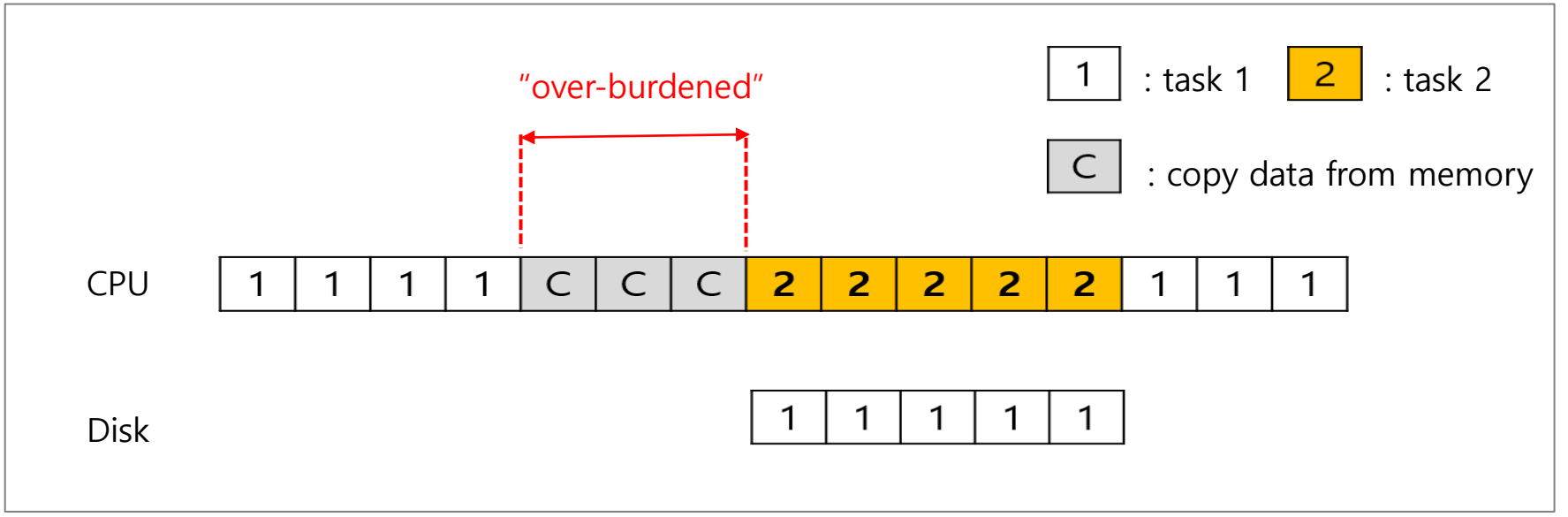
OPERATING SYSTEMS

Six Step Process to Perform DMA Transfer



CPU utilization

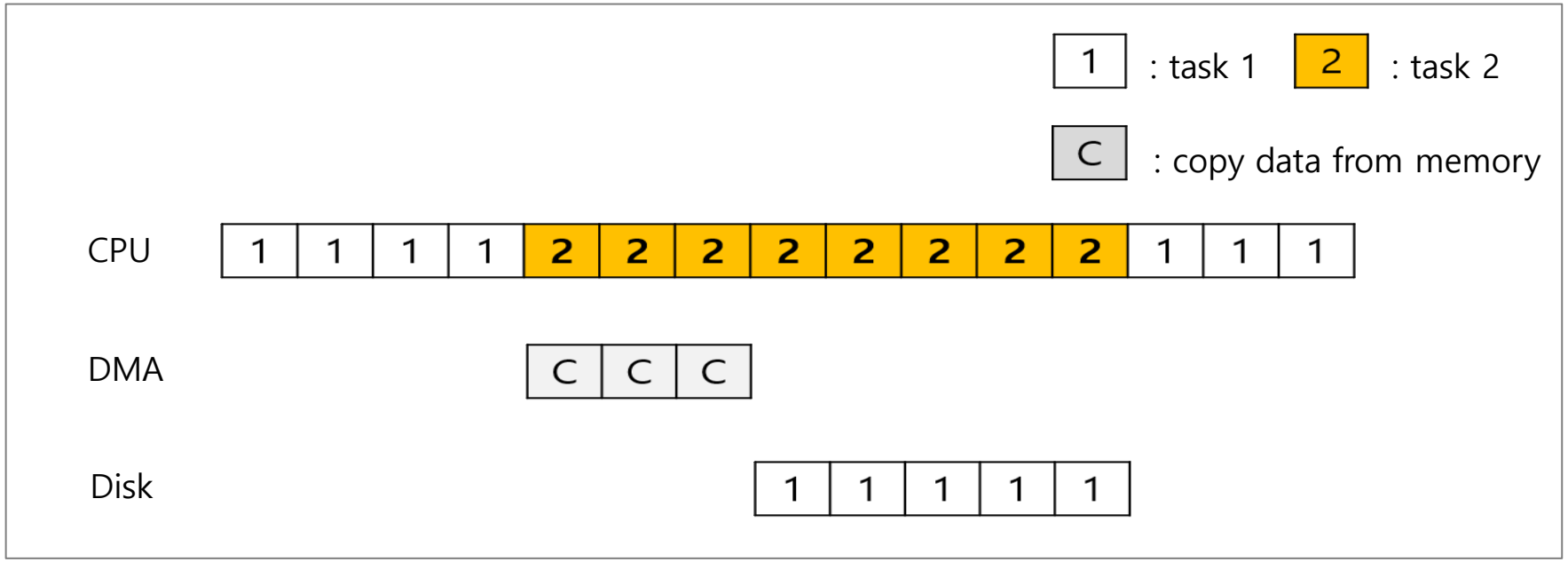
- CPU **wastes a lot of time** to copy a large chunk of data from memory to the device.



OPERATING SYSTEMS

CPU utilization by DMA

- **Copy data** in memory by knowing “where the data lives in memory, how much data to copy”
- When completed, DMA raises an interrupt, I/O begins on Disk.



DMA Transfer Modes

The DMA controller transfers the data in three modes:

1.Burst Mode: Here, once the DMA controller gains the charge of the system bus, then it releases the system bus only after **completion** of data transfer. Till then the CPU has to wait for the system buses.

2.Cycle Stealing Mode: In this mode, the DMA controller **forces** the CPU to stop its operation and **relinquish the control over the bus** for a **short term** to DMA controller. After the **transfer of every byte**, the DMA controller **releases** the **bus** and then again requests for the system bus. In this way, the DMA controller steals the clock cycle for transferring every byte.

3.Transparent Mode: Here, the DMA controller takes the charge of system bus only if the **processor does not require the system bus**.

OPERATING SYSTEMS

DMA Controller Data Transfer

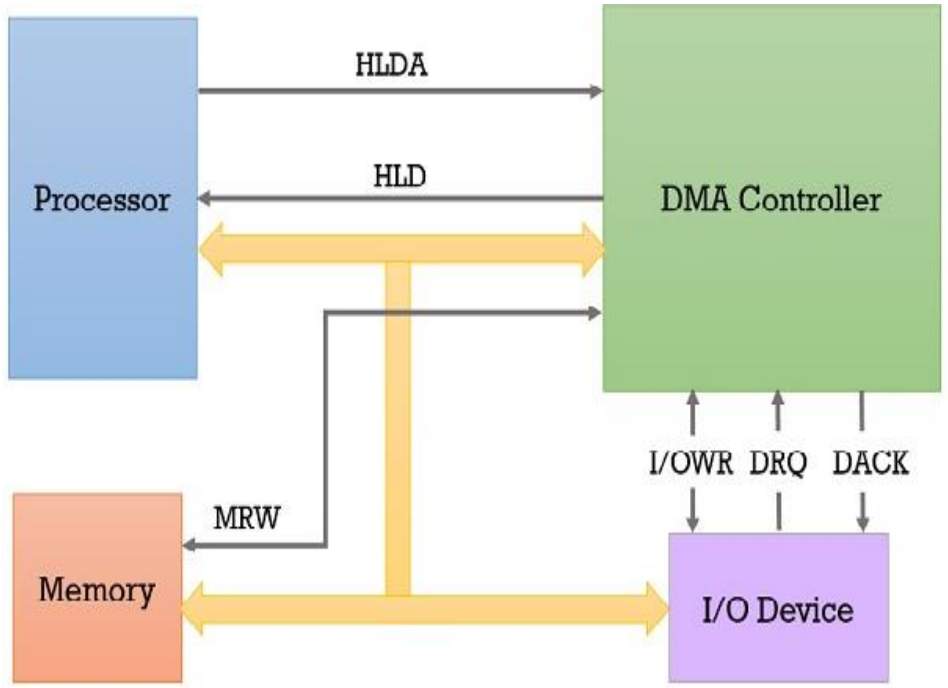
❖I/O device sends the DMA request (**DRQ**) to the DMA controller.

❖DMA controller accepts this DRQ and asks the CPU to hold for a few clock cycles by sending it the Hold request (**HLD**).

❖CPU receives the Hold request (HLD) from DMA controller and relinquishes the bus and sends the Hold acknowledgement (**HLDA**) to DMA controller.

❖DMA controller acknowledges I/O device **(DACK)** that the data transfer can be performed and DMA controller takes charge of the system bus and transfers the data to or from memory.

❖When the data transfer is accomplished, the DMA sends an **interrupt** to the processor



THANK YOU

Kakoli Bora
Department of Computer Science Engineering

K_bora@pes.edu