Interrupt-driven I/O and Direct Memory Access (DMA)

Muhammad Afzaal

m.afzaal@nu.edu.pk

Book Chapter

- “Computer Organization and Architecture”
- Author “William Stallings”
- 8th Edition
- Chapter 7
  - Section 7.4
  - Section 7.5

Interrupt-driven I/O

- Overcomes CPU waiting in Programmed I/O
- No repeated CPU checking of devices
- I/O module interrupts when ready

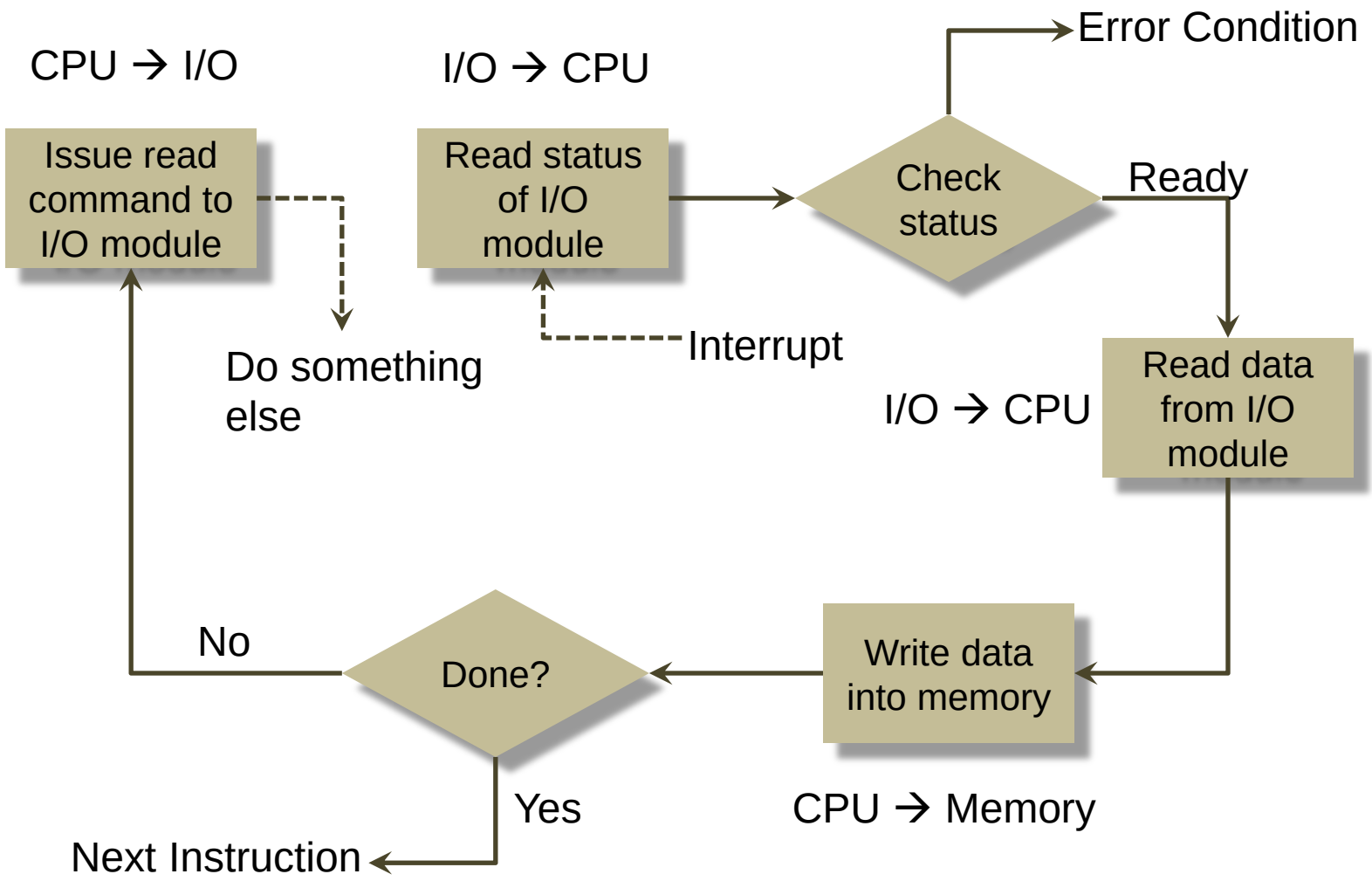
Basic Operation from I/O Module Viewpoint

- CPU issues read command
- I/O module gets data from peripheral whilst CPU does other work
- I/O module reads data from the peripheral
- When ready, I/O module send an interrupt signal on control line to CPU and waits
- CPU requests data
- I/O module puts data on the data bus

Basic Operation from CPU Viewpoint

- Issues read command
- Does some other work
- Checks for interrupt at end of each instruction cycle
- If interrupted by I/O module
  - Save context (PC and other Registers) of current program
  - Process interrupt

Interrupt-driven I/O Block Diagram



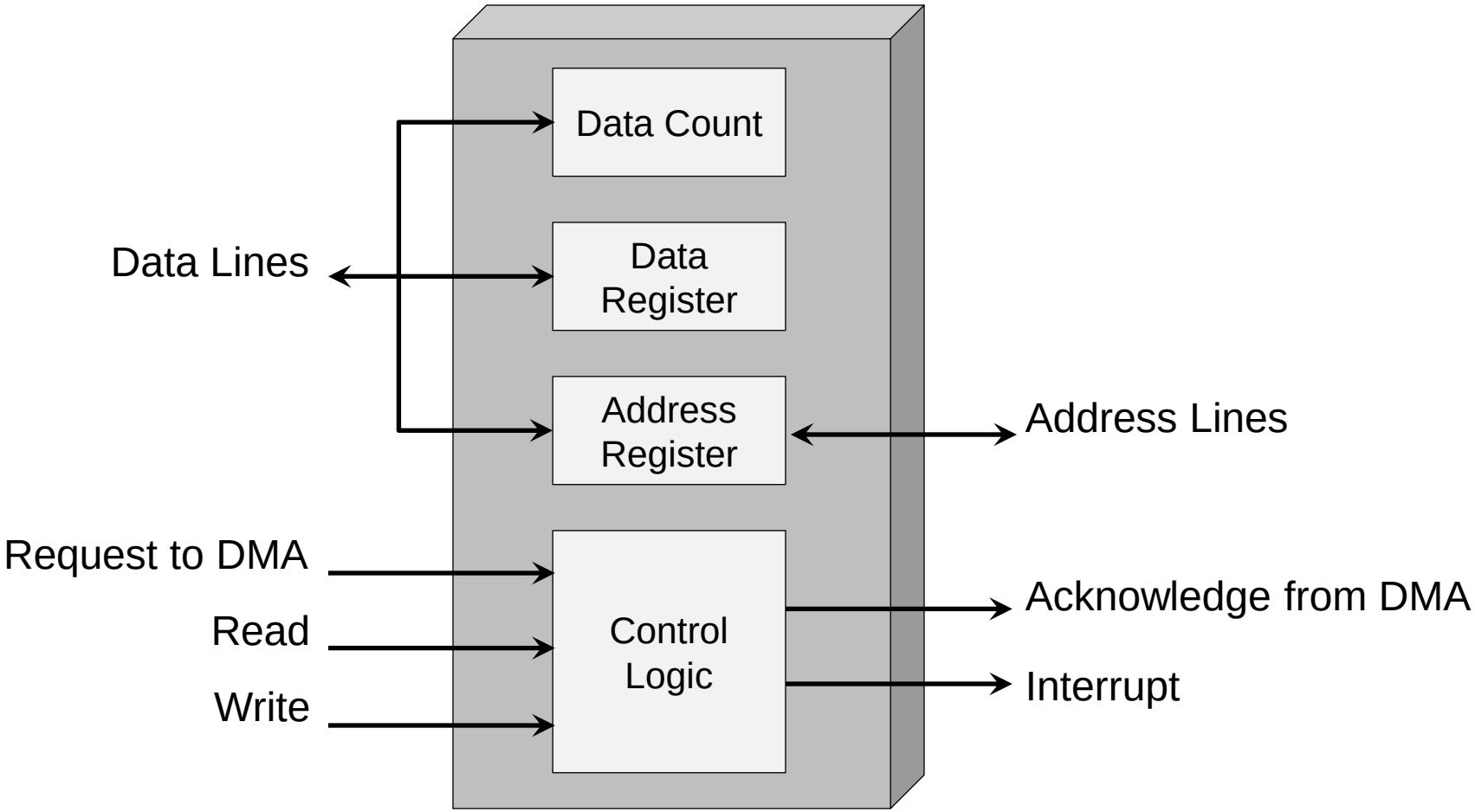
Design Issues

- How to identify the module issuing the interrupt?
  - Multiple Interrupt Lines between processor and I/O modules
  - Software Poll to determine which module caused the interrupt
  - Daisy Chain: hardware poll
  - Bus Arbitration: Module gains control of the bus
- How to deal with multiple interrupts?
  - Each interrupt line has a priority
  - Higher priority lines can interrupt lower priority lines

Direct Memory Access

- Drawbacks of Programmed and Interrupt-driven I/O
  - Limited I/O transfer rate
  - Processor tied up in managing an I/O transfer
- Direct Memory Access is a better approach when transferring large amount of data
- Involves additional module on system bus called DMA module
- DMA module takes over control from CPU to transfer data to/from memory over system bus

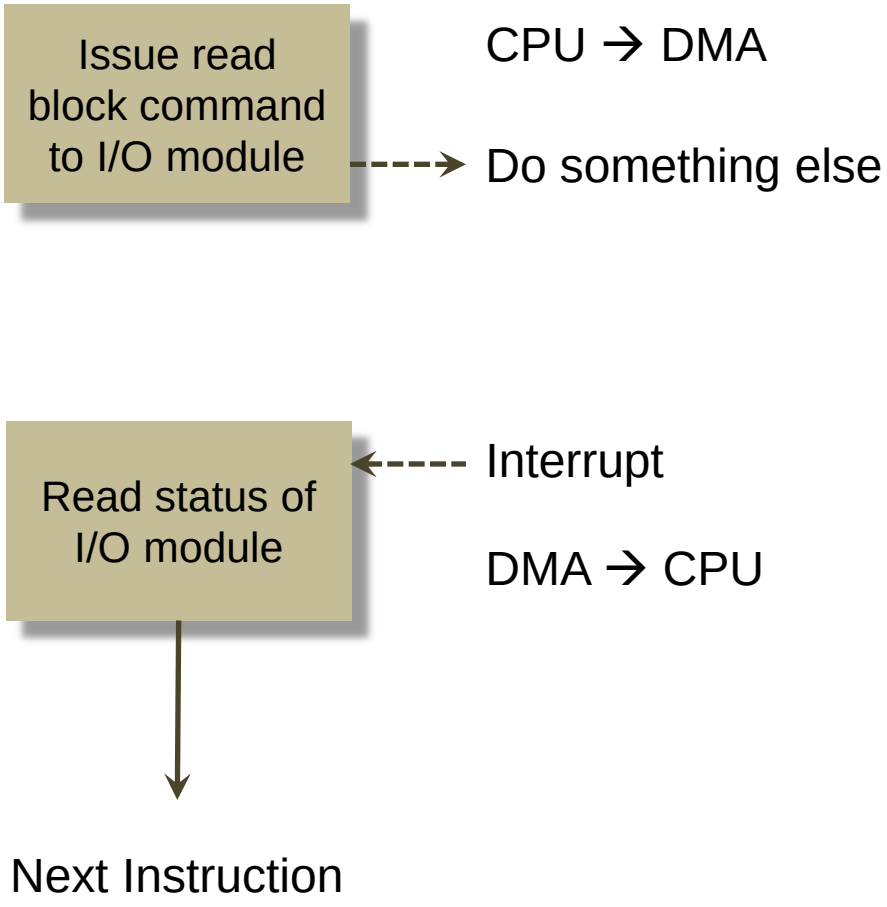
DMA Module Diagram



DMA Operation

- CPU tells DMA module
  - Read/Write
  - Device Address
  - Starting address of memory block for data
  - Amount of data to be transferred
- CPU carries on with other work
- DMA module deals with transfer
- DMA module sends interrupt to processor when finished

DMA Block Diagram



Cycle Stealing

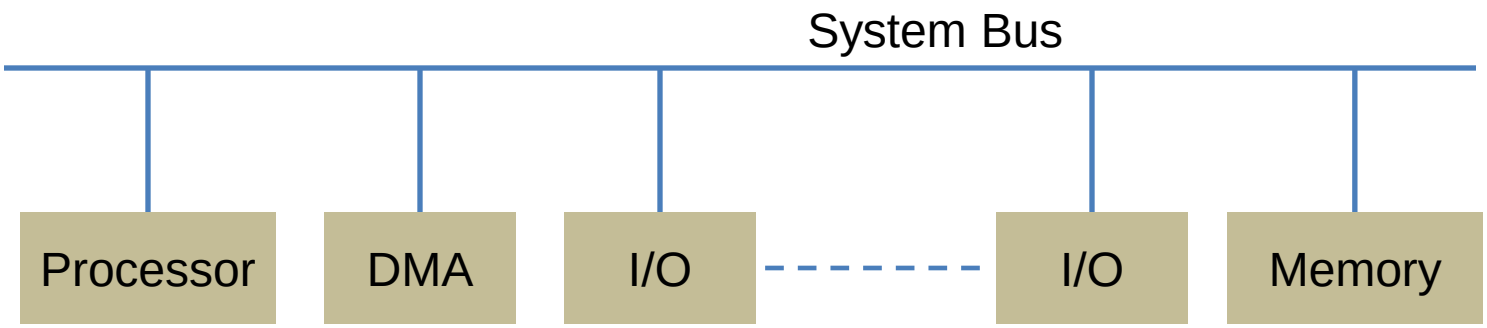
- DMA module takes over the system bus to transfer data to and from memory
- DMA must use the bus when processor does not need it
- DMA can force the processor to suspend operation temporarily
  - Not an interrupt so CPU does not switch context
- CPU suspends just before it accesses bus
  - i.e. before an operand or data fetch or a data write
- Called cycle stealing because DMA module steals a clock cycle

DMA Configuration

- Three configuration mechanisms
  - Single bus, detached DMA
  - Single bus, Integrated DMA-I/O
  - I/O bus

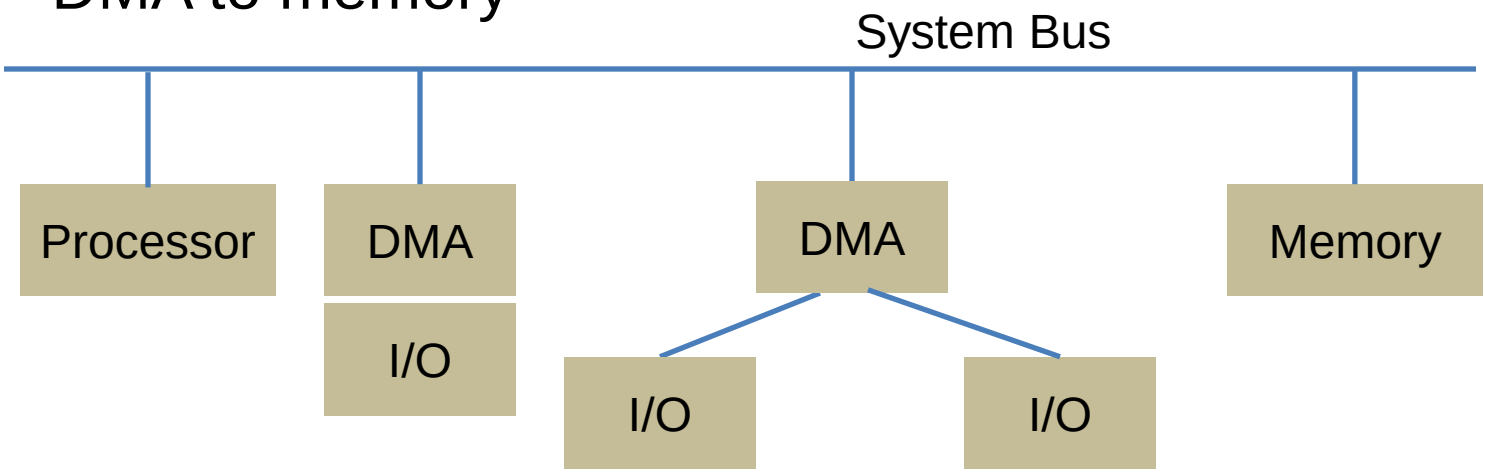
Single Bus, Detached DMA

- Single bus, detached DMA module
- Module may support more than one devices.
- Each transfer uses bus twice
  - First from I/O to DMA
  - Then from DMA to memory



Single Bus, Integrated DMA

- Path between DMA module and I/O module
- Module may support more than one devices
- I/O module is not on address bus
- Each transfer uses bus once
  - DMA to memory



I/O Bus

- Separate I/O bus
- Bus supports all DMA enabled devices
- Each transfer uses bus once
  - From DMA to memory