I/O Management

Categories of I/O Devices

- Human readable
  - used to communicate with the user
  - video display terminals
  - keyboard
  - mouse
  - printer

Categories of I/O Devices

- ✓ Machine readable
  - used to communicate with electronic equipment
  - disk drives
  - tape drives
  - controllers
  - actuators

Categories of I/O Devices

- Communication
  - used to communicate with remote devices
  - digital line drivers
  - modems

Differences in I/O Devices

- ✓ Data Transfer Rate
- ✓ Application-specific
  - disk used to store files must have file-management software
  - disk used to store virtual memory pages depends on virtual memory hardware; I/O ops may be scheduled differently than for disks used for file storage
  - terminal used by system administrator may have a higher priority

Differences in I/O Devices

- Complexity of control
- Unit of transfer
  - data may be transferred as a stream of bytes for a terminal or in larger blocks for a disk
- Data representation
  - encoding schemes: character encoding, parity may be different
- Error conditions
  - devices respond to errors differently

Techniques for Performing I/O

- Programmed I/O
  - process is busy-waiting for the operation to complete
- Interrupt-driven I/O
  - I/O command is issued
  - processor continues executing instructions
  - I/O module sends an interrupt when done

Techniques for Performing I/O

- ✓ Direct Memory Access (DMA)
  - DMA module controls exchange of data between main memory and the I/O device
  - processor interrupted only after entire block has been transferred

Evolution of the I/O Function

- ✓ Processor directly controls a peripheral device
- ✓ Controller or I/O module is added
  - processor uses programmed I/O without interrupts
  - processor does not need to handle details of external devices

Evolution of the I/O Function

- Controller or I/O module with interrupts
  - processor does not spend time waiting for an I/O operation to be performed
- Direct Memory Access
  - blocks of data are moved into memory without involving the processor
  - processor involved at beginning and end only

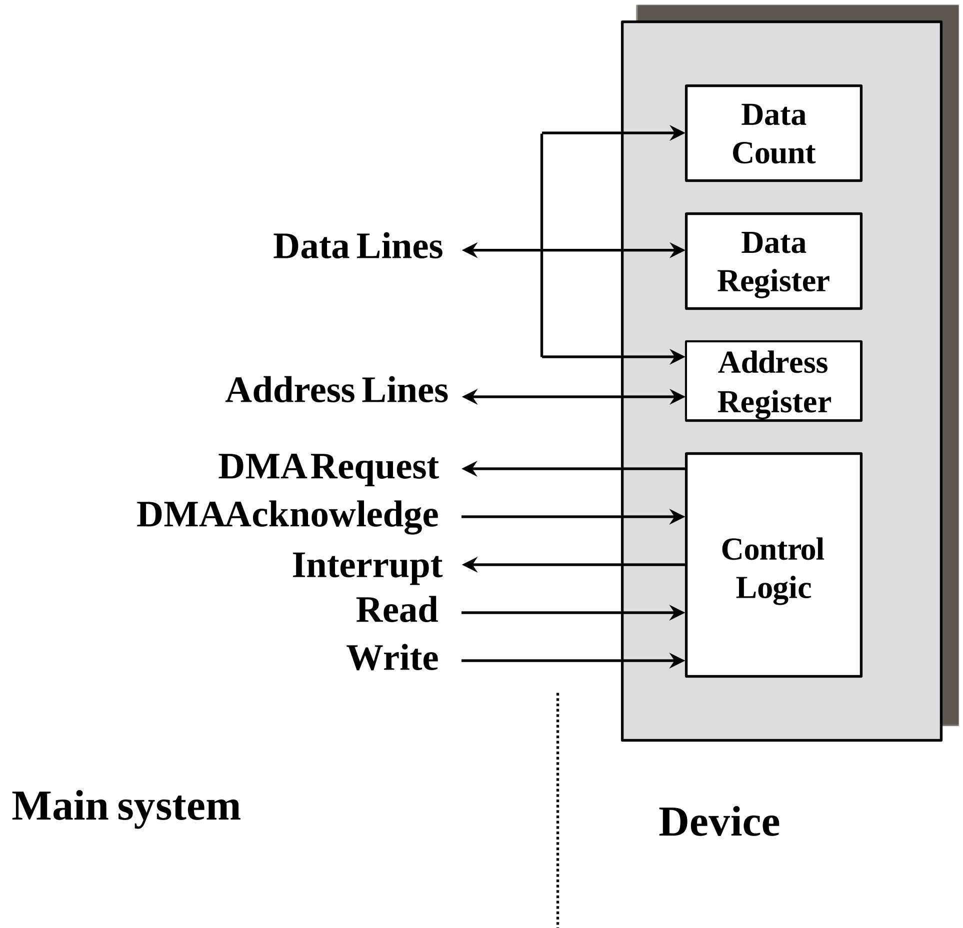
Evolution of the I/O Function

- I/O channel
  - I/O module is a separate processor
  - Uses computer's main memory
- I/O processor
  - I/O module is a processor with its own local memory
  - It's a computer in its own right

Direct Memory Access

- ✓ Takes control of the system form the CPU to transfer data to and from memory over the system bus
- ✓ Cycle stealing is used to transfer data on the system bus
- ✓ The instruction cycle is suspended so data can be transferred
- ✓ The CPU pauses one bus cycle
- ✓ No interrupts occur
  - does not need to save context

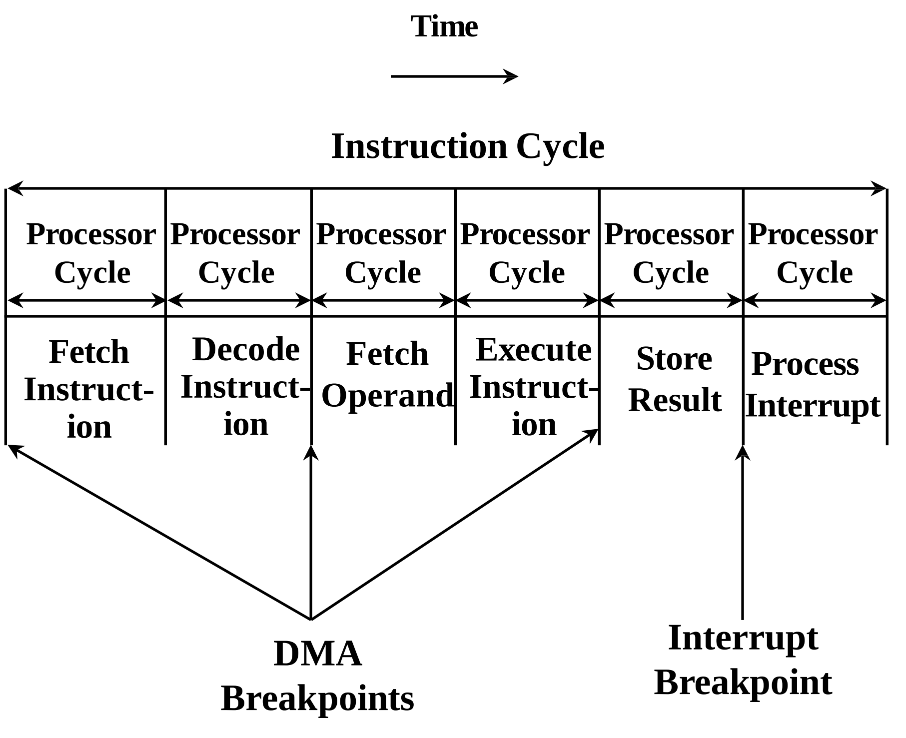
Typical DMA Block Diagram



Direct Memory Access

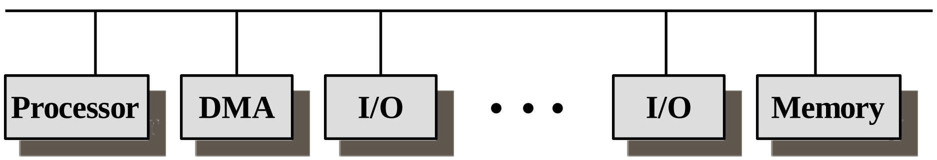
- ✓ Cycle stealing causes the CPU to execute more slowly
- ✓ Number of required busy cycles can be cut by integrating the DMA and I/O functions
- ✓ Try to use path between DMA module and I/O module that does not include the system bus

DMA and Interrupt Breakpoints

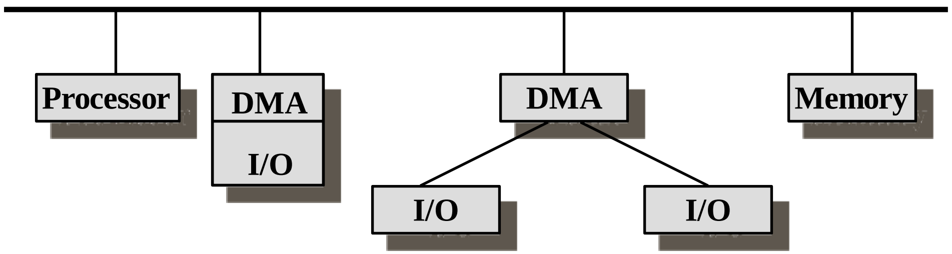


Breakpoints where CPU can be suspended to let the DMA module use the buss

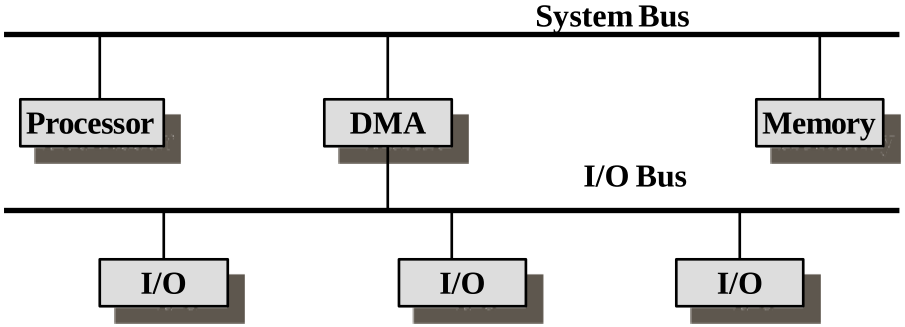
Single-bus, Detached DMA



Single-bus, Integrated DMA-I/O



I/O Bus



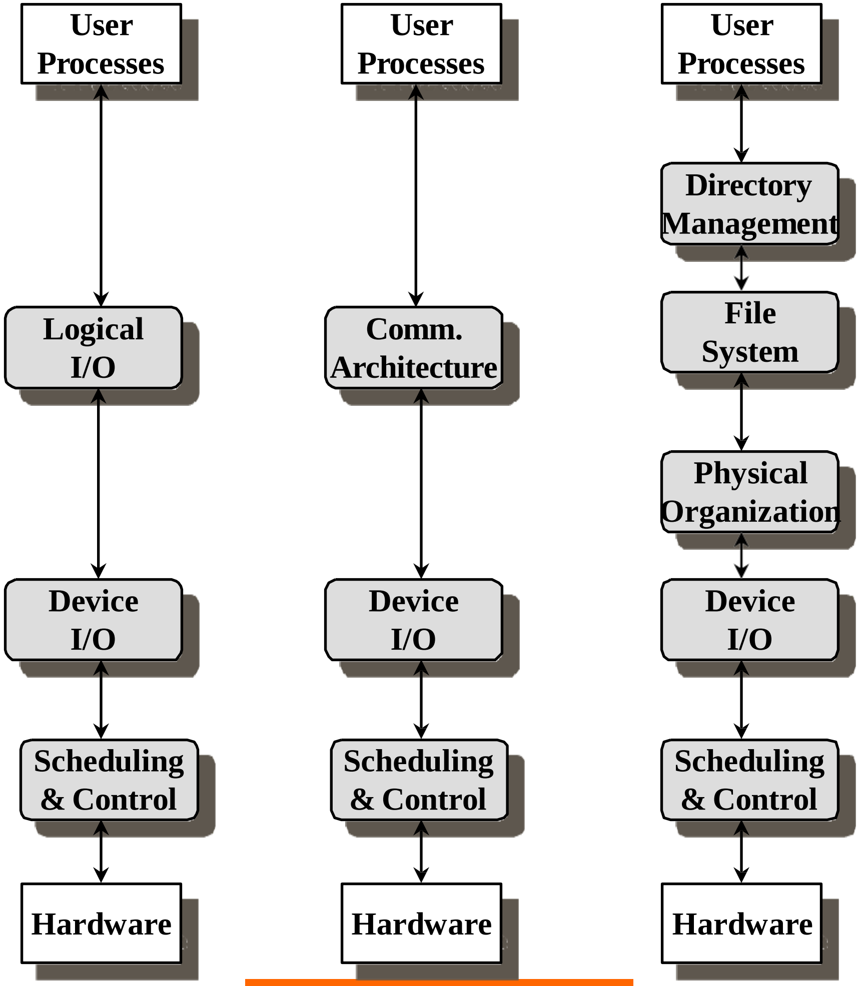
Operating System Design Objectives

- ✓ I/O is extremely slow compared to main memory
- ✓ Use of multiprogramming allows that some processes will be waiting on I/O while another process executes
- ✓ I/O cannot keep up with processor speed
- ✓ Swapping is used to bring in additional Ready processes, which is an I/O operation
- ✓ Efficiency of I/O is an important issue, since this is a bottleneck

Operating System Design Objectives

- Desirable to handle all I/O devices in a uniform manner
- Hide most of the details of device I/O in lower-level routines so that processes and upper levels see devices in general terms such as Read, Write, Open, and Close
- Generality is an important issue

A Model of I/O Organization



Local peripheral device

Communications port

Device with a File System

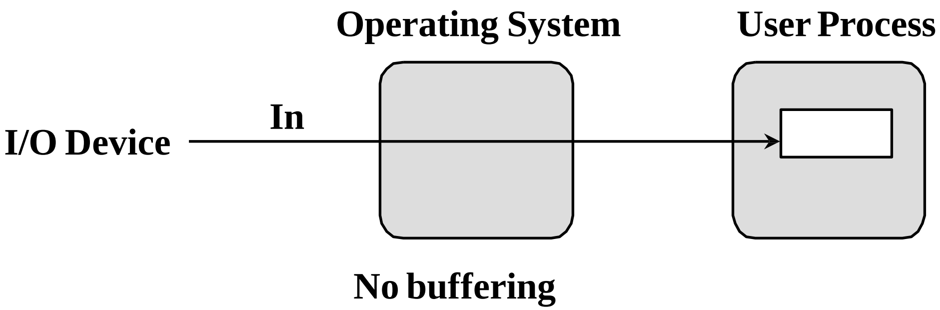
I/O Buffering

- ✓ Reasons for buffering: to find a solution to these problems:
  - Processes must wait for I/O to complete before proceeding
  - Certain pages must remain in main memory during I/O – interferes with page replacement

I/O Buffering

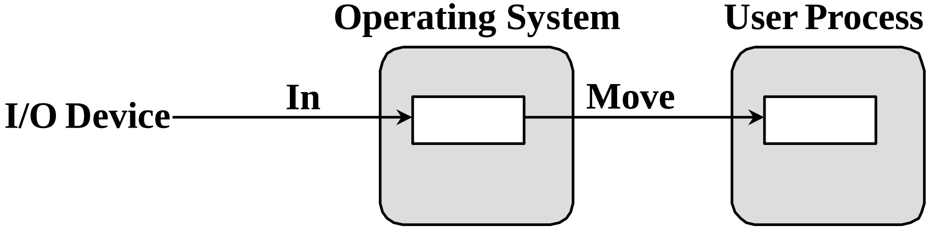
- Block-oriented
  - information is stored in fixed sized blocks
  - transfers are made a block at a time
  - used for disks and tapes
- Stream-oriented
  - transfer information as a stream of bytes
  - used for terminals, printers, communication ports, mouse, and most other devices that are not secondary storage

No Buffering



Single Buffer

- ✓ Operating system assigns a buffer in main memory for an I/O request
- ✓ Block-oriented
  - input transfers are made to buffer
  - block moved to user space when needed
  - another block is moved into the buffer
    - *read ahead*



Single buffering

Single Buffer

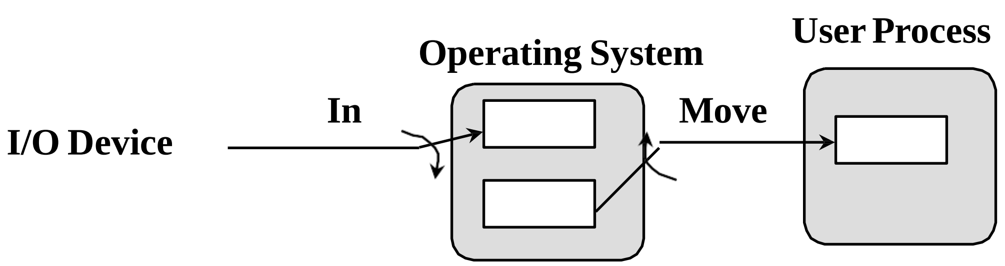
- ✓ *Block-oriented I/O:*
  - user process can work on one block of data while next block is being read in
  - process waiting for I/O can be swapped out, since input is taking place in system memory, not user memory
  - operating system keeps track of assignment of system buffers to user processes
  - output is accomplished by the user process writing a block to the buffer and later actually written out

Single Buffer

- *Stream-oriented:*
  - used one line at a time
  - user input from a terminal is one line at a time with carriage return signaling the end of the line
  - output to the terminal is one line at a time

Double Buffer

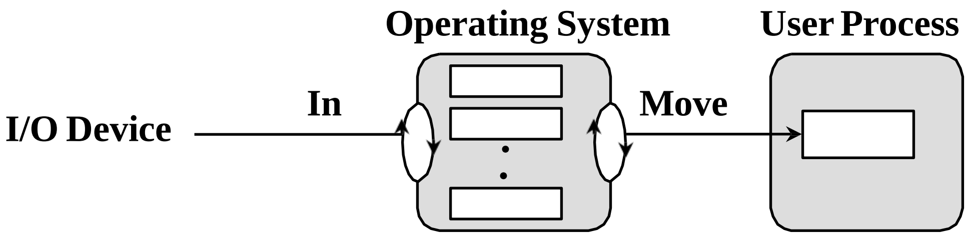
- ✓ Use two system buffers instead of one
- ✓ A process can transfer data to or from one buffer while the operating system empties or fills the other buffer



Double buffering

Circular Buffer

- More than two buffers are used
- Each individual buffer is one unit in a circular buffer
- Used when I/O operation must keep up with process



Circular buffering