

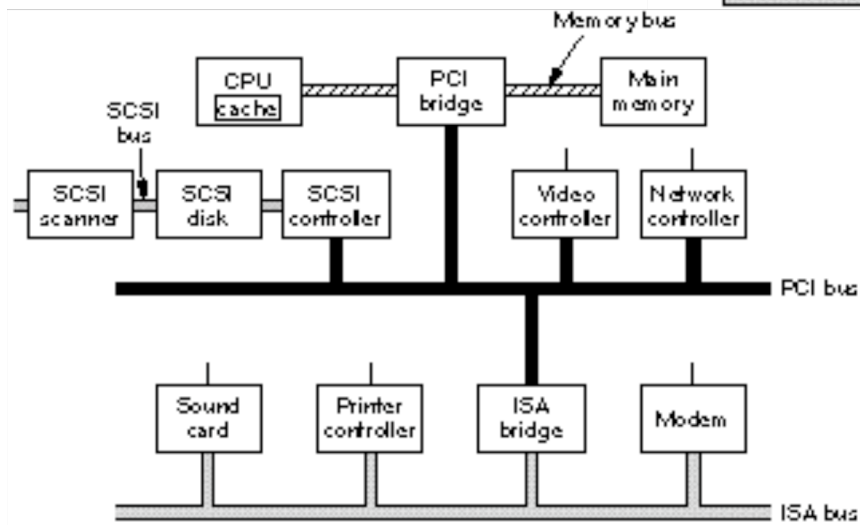
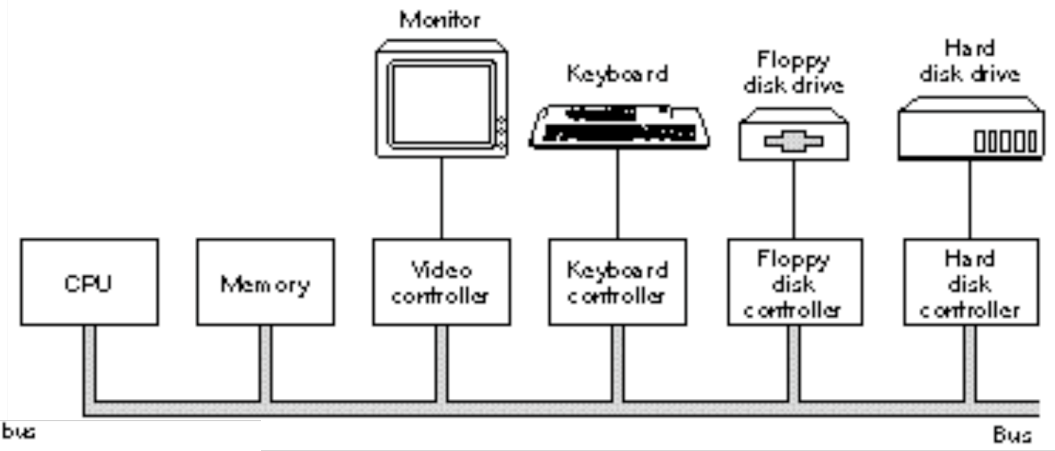
# Unit 7: The Input/Output System

## 7.1. Principles of I/O Systems

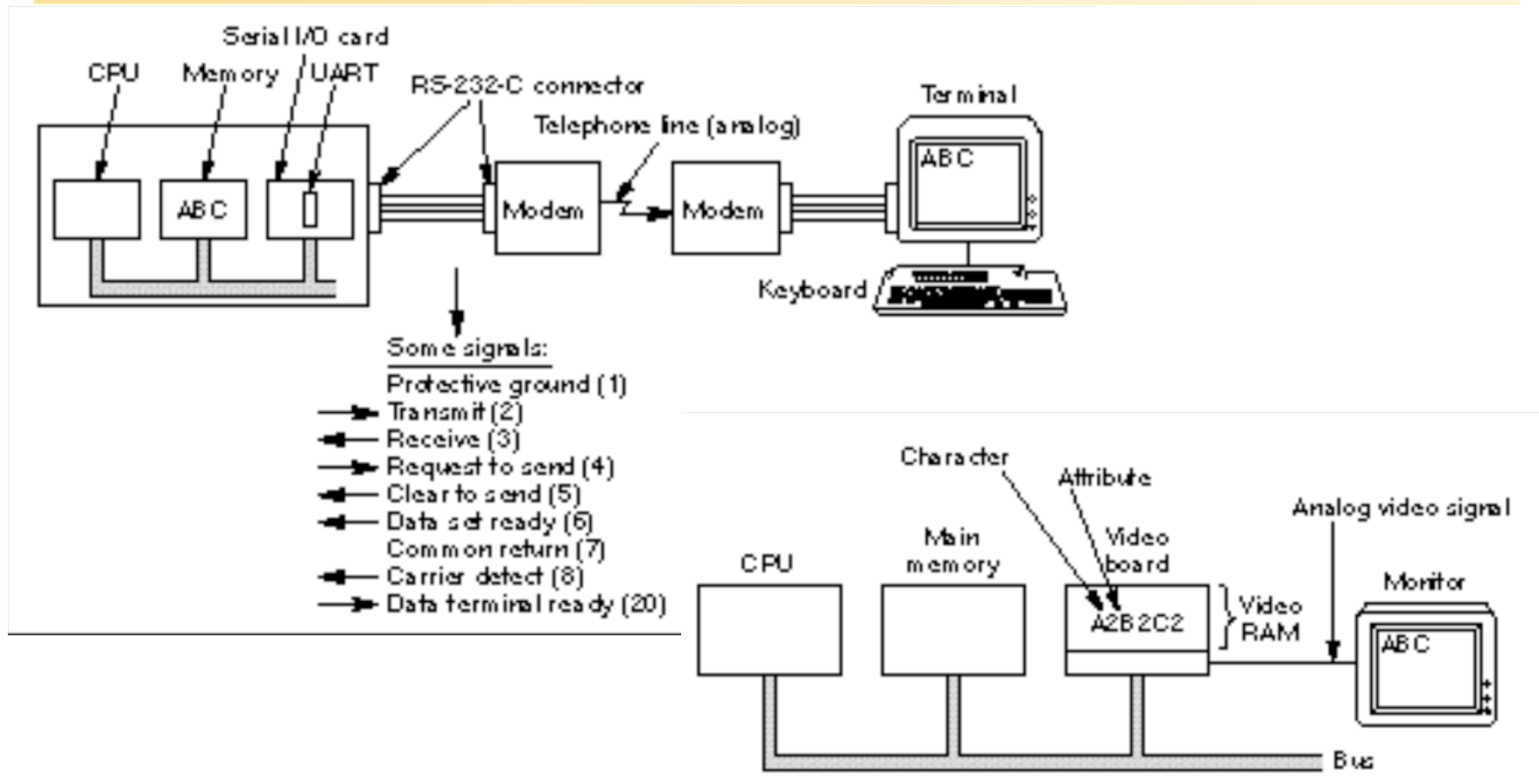
# Input/Output – Principles of I/O Hardware

- Major components of a computer system:  
CPU, memories (primary/secondary), I/O system
- I/O devices:
  - **Block devices** – store information in fixed-sized blocks;  
typical sizes: 128-1024 bytes
  - **Character devices** – delivers/accepts stream of characters
- Device controllers:
  - Connects physical device to system bus (Minicomputers, PCs)
  - Mainframes use a more complex model:  
Multiple buses and specialized I/O computers (I/O channels)
- Communication:
  - Memory-mapped I/O, controller registers
  - Direct Memory Access - DMA

# I/O Hardware - Buses



# I/O Hardware – Terminals



# Principles of I/O Software

- Layered organization
- Device independence
- Uniform naming (e.g. UNIX file names)
- Error handling
  - Error should be handled as close to the hardware as possible
  - Transparent error recovery at low level
- Synchronous vs. Asynchronous transfers
  - Most physical I/O is asynchronous
- Sharable vs. Dedicated devices
  - Disk vs. printer

## Structuring of I/O software

1. Interrupt handlers
2. Device drivers
3. Device-independent OS software
4. User-level software

# Interrupt Handlers

- Should be hidden by the operating system
- Every process starting an I/O operation should block until I/O has completed and interrupt occurs
- Interrupt handler transfers data from device (controller) and un-blocks process

# Device Drivers

- Contains all device-dependent code
- Handles one type (class) of devices
- Translates abstract requests into device commands
  - Writes controller registers
  - Accesses mapped memory
  - Queues requests
- Driver may block after issuing a request:
  - Interrupt will un-block driver (returning status information)

# Device-independent I/O Software

Functions of device-independent I/O software:

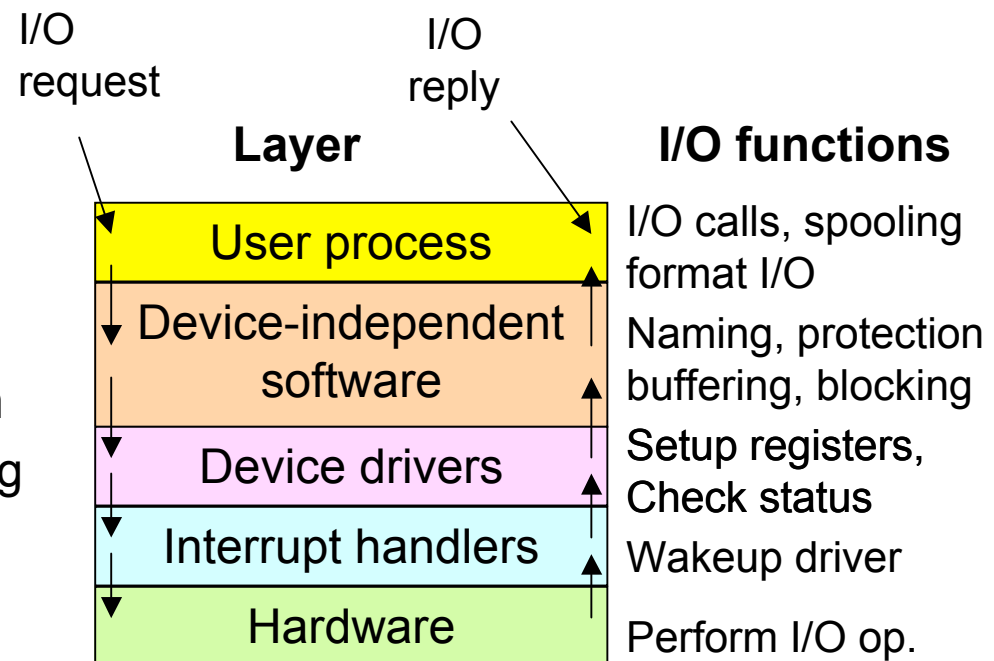
- Uniform interfacing for the device drivers
- Device naming
- Device protection
- Providing a device-independent block size
- Buffering
- Storage allocation on block devices
- Allocating and releasing dedicated devices
- Error reporting



# Layers of the I/O System

## User-Space I/O Software

- System call libraries (read, write,...)
- Spooling
  - Managing dedicated I/O devices in a multiprogramming system
  - Daemon process, spooling directory
  - lpd – line printer daemon, sendmail – simple mail transfer protocol



# Example:

## 4.3BSD kernel I/O structure

