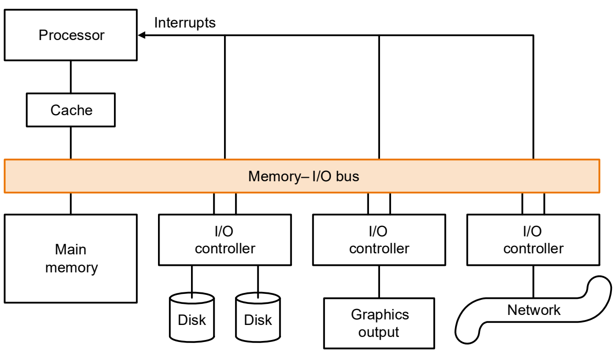
Peripherals (1/2)

Interfacing Processors and Peripherals

- I/O Design affected by many factors (expandability, resilience)
- Performance:
  - — access latency
  - — throughput
  - — connection between devices and the system
  - — the memory hierarchy
  - — the operating system
- A variety of different users (e.g., banks, supercomputers, engineers)

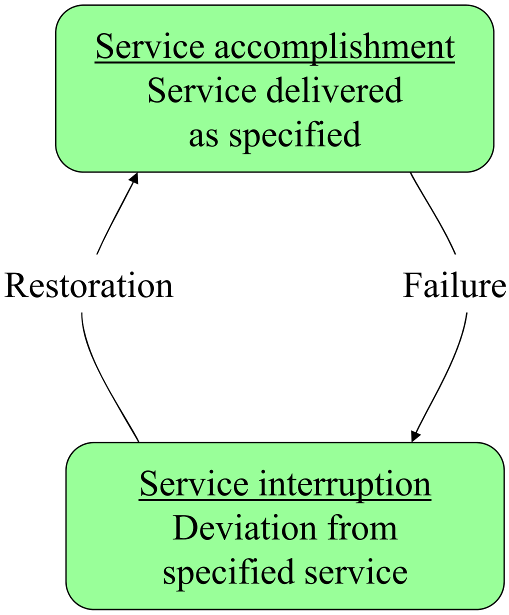


I/O System Characteristics

- **Dependability is important**
  - **Particularly for storage devices**
- **Performance measures**
  - **Latency (response time)**
  - **Throughput (bandwidth)**
  - **Desktops & embedded systems**
    - Mainly interested in response time & diversity of devices
  - **Servers**
    - Mainly interested in throughput & expandability of devices

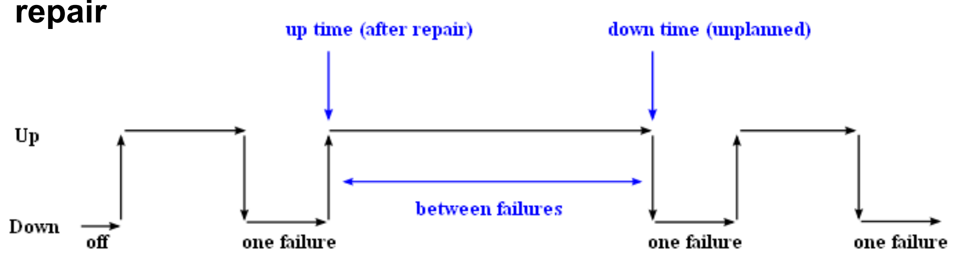
Dependability

- **Fault: failure of a component**
  - **May or may not lead to system failure**



Dependability Measures

- Reliability: mean time to failure (MTTF)
- Service interruption: mean time to repair (MTTR)
- Mean time between failures
  - MTBF = MTTF + MTTR
- Availability = MTTF / (MTTF + MTTR) = MTTF / MTBF
- Improving Availability
  - Increase MTTF: fault avoidance, fault tolerance, fault forecasting
  - Reduce MTTR: improved tools and processes for diagnosis and repair

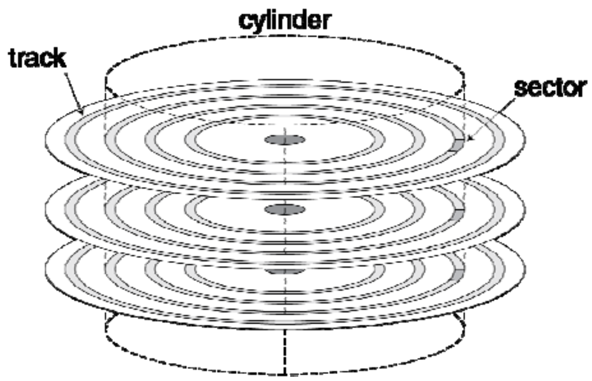


Time Between Failures = { down time - up time}

$$\text{Mean time between failures} = \text{MTBF} = \frac{\Sigma(\text{downtime} - \text{uptime})}{\text{number of failures}}.$$

Disk Storage

- Nonvolatile, rotating magnetic storage



Disk Sectors and Access

- **Each sector records**
  - **Sector ID**
  - **Data (512 bytes, 4096 bytes proposed)**
  - **Error correcting code (ECC)**
    - Used to hide defects and recording errors
  - **Synchronization fields and gaps**
- **Access to a sector involves**
  - **Queuing delay if other accesses are pending**
  - **Seek: move the heads**
  - **Rotational latency**
  - **Data transfer**
  - **Controller overhead**

Disk Access Example

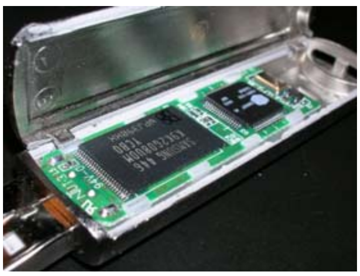
- **Given**
  - **512B sector, 15,000rpm, 4ms average seek time, 100MB/s transfer rate, 0.2ms controller overhead, idle disk**
- **Average read time**
  - **4ms seek time**
    + ½ / (15,000/60) = 2ms rotational latency
    + 512 / 100MB/s = 0.005ms transfer time
    + 0.2ms controller delay
    = 6.2ms
- **If actual average seek time is 1ms**
  - **Average read time = 3.2ms**

Disk Performance Issues

- Manufacturers quote average seek time
  - Based on all possible seeks
  - Locality and OS scheduling lead to smaller actual average seek times
- Smart disk controller allocate physical sectors on disk
  - Present logical sector interface to host
  - SCSI (Small Computer System Interface), ATA (Advanced Technology Attachment), SATA (Serial ATA)
- Disk drives include caches
  - Prefetch sectors in anticipation of access
  - Avoid seek and rotational delay

Flash Storage

- Nonvolatile semiconductor storage
  - 100× – 1000× faster than disk
  - Smaller, lower power, more robust
  - But more $/GB (between disk and DRAM)



Flash Types

- NOR flash: bit cell like a NOR gate
  - Random read/write access
  - Used for instruction memory in embedded systems
- NAND flash: bit cell like a NAND gate
  - Denser (bits/area), but block-at-a-time access
  - Cheaper per GB
  - Used for USB keys, media storage, …
- Flash bits wears out after 1000's of accesses
  - Not suitable for direct RAM or disk replacement
  - Wear leveling: remap data to less used blocks

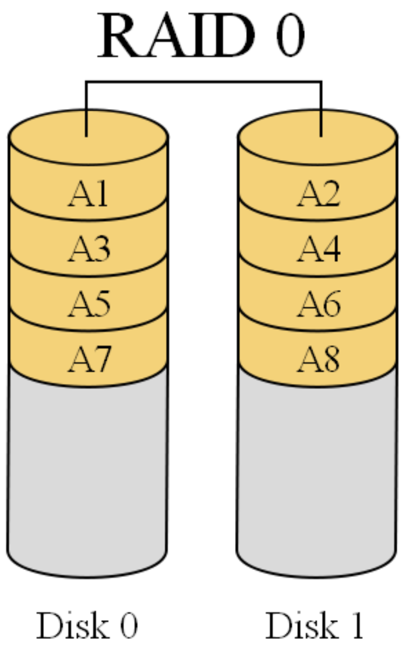
RAID

- Redundant Array of Inexpensive (Independent) Disks
  - Use multiple smaller disks (c.f. one large disk)
  - Parallelism improves performance
  - Plus extra disk(s) for redundant data storage
- Provides fault tolerant storage system
  - Especially if failed disks can be "hot swapped"
- It's a technology that enables greater levels of performance, reliability and/or large volumes when dealing with data.
- Mirroring, Stripping (of data) and Error correction techniques combined with multiple disk arrays give you the reliability and performance.

RAID 0

- It splits data among two or more disks.
- Provides good performance.
- Lack of data redundancy means there is no fail over support with this configuration.
- In the diagram to the right, the odd blocks are written to disk 0 and the even blocks to disk 1 such that A1, A2, A3, A4, … would be the order of blocks read if read sequentially from the beginning.
- Used in read only NFS systems and gaming systems.

- JBOD = Just a bunch of disks



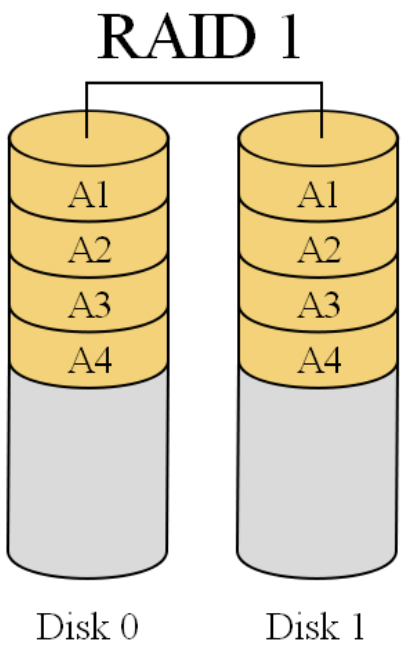
RAID 0 analysis

- Failure Rate:
  - MTBF of RAID0 is roughly proportional to the number of disks in the array.
  - Pr(disk.fail) = 5%, then
    Pr(at.least.one.fails) = 1 – Pr(none.fails) = 1 – $(1 - 0.05)^2$ = 9.75%

- Performance:
  - The fragments are written to their respective disks simultaneously on the same sector.
  - This allows smaller sections of the entire chunk of data to be read off the drive in parallel, hence good performance.

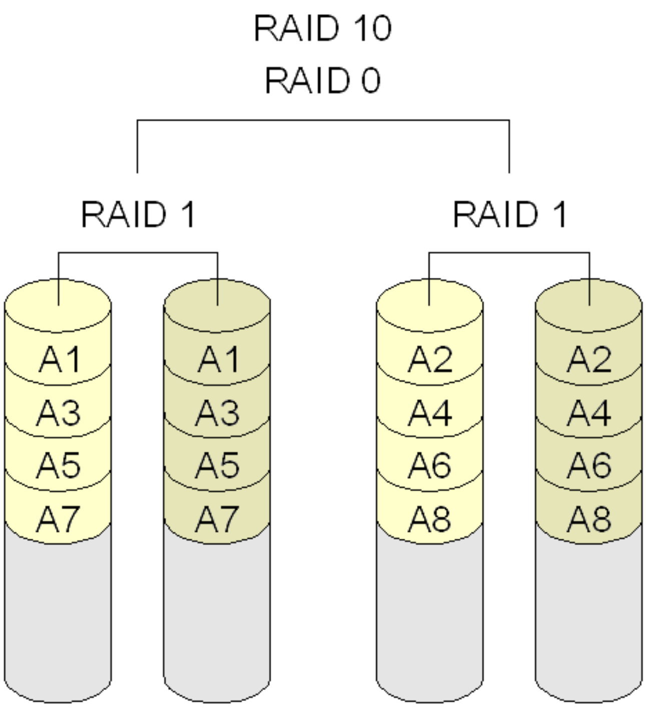
RAID 1

- RAID1 is 'data mirroring'.
- Two copies of the data are held on two physical disks, and the data is always identical.
- Twice as many disks are required to store the same data when compared to RAID 0.
- Array continues to operate so long as at least one drive is functioning.
- Failure Rate:
  – If Pr(disk.fail) = 5%, then the probability of both the drives failing in a 2-disk array is Pr(both.fail) = $(0.05)^2$ = 0.25%.
- Performance:
  – If we use independent disk controllers for each disk, then we can increase the read or write speeds by doing operations in parallel.



RAID 10

- Combines RAID 1 and RAID 0.
- Which means having the pleasure of both - good performance and good failover handling.
- Also called 'Nested RAID'.
- RAID 01
  – Mirroring of striped disks
  – 2^{nd} failure while repairing results in the system failure.



RAID 2: Bit-Interleaved with ECC

- Error correcting code (ECC)
  - N + E disks (e.g., 10 + 4)
  - Split data at bit level across N disks
  - Generate E-bit ECC
  - Too complex, not used in practice

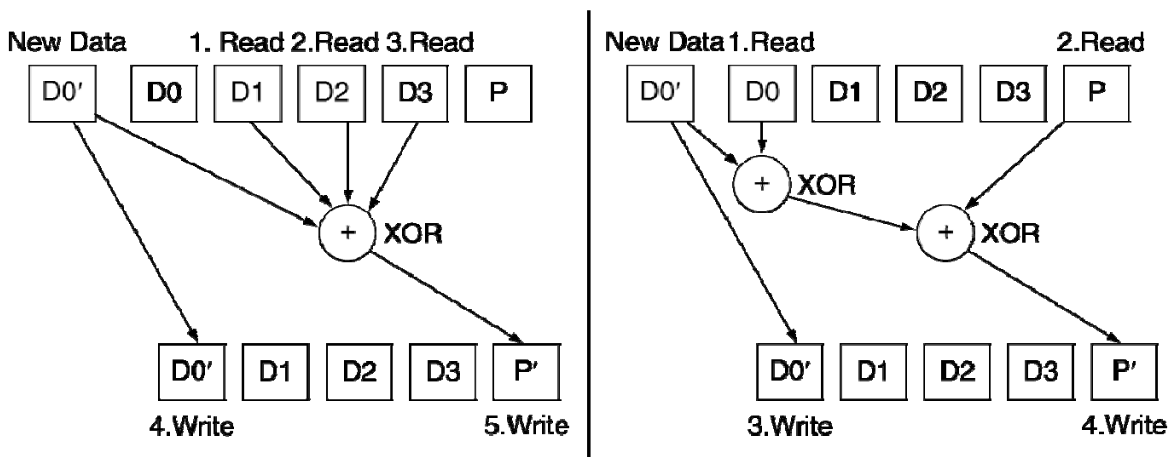
RAID 3: Byte-Interleaved Parity

- N + 1 disks
  - Data striped across N disks at byte level
  - Redundant disk stores parity
  - Read access
    - Read all disks
  - Write access
    - Generate new parity and update all disks
  - On failure
    - Use parity to reconstruct missing data
- Not widely used

RAID 4: Block-Interleaved Parity

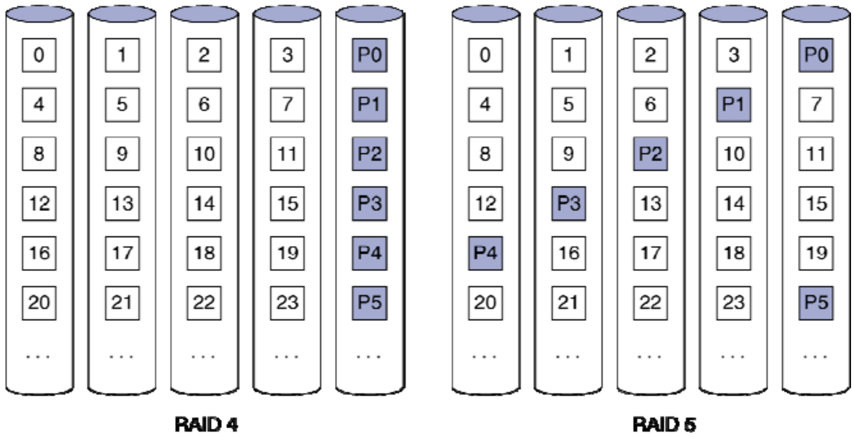
- **N + 1 disks**
  - **Data striped across N disks at block level**
  - **Redundant disk stores parity for a group of blocks**
  - **Read access**
    - Read only the disk holding the required block
  - **Write access**
    - Just read disk containing modified block, and parity disk
    - Calculate new parity, update data disk and parity disk
  - **On failure**
    - Use parity to reconstruct missing data
- **Not widely used**

RAID 3 vs RAID 4



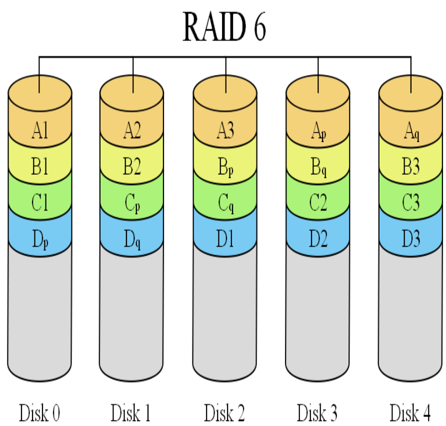
RAID 5: Distributed Block Parity

- **N + 1 disks**
  - **Like RAID 4, but parity blocks distributed across disks**
    - Avoids parity disk being a bottleneck
- **Widely used**



RAID 6: P + Q Redundancy

- **N + 2 disks**
  - **Like RAID 5, but two lots of parity**
  - **Greater fault tolerance through more redundancy**
- **It is seen as the best way to guarantee data integrity as it uses double parity.**
- **Lesser MTBF compared to RAID5.**
- **It has a drawback though of longer write time.**



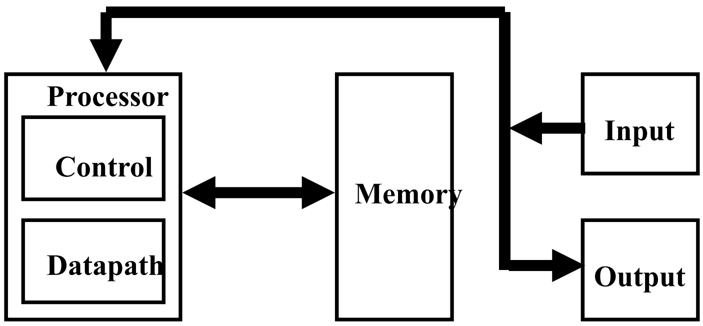
Implementations

- Software based RAID:
  - Software implementations are provided by many Operating Systems.
  - A software layer sits above the disk device drivers and provides an abstraction layer between the logical drives(RAIDs) and physical drives.
  - Server's processor is used to run the RAID software.
  - Used for simpler configurations like RAID0 and RAID1.

- Hardware based RAID:
  - A hardware implementation of RAID requires at least a special-purpose RAID controller.
  - On a desktop system this may be built into the motherboard.
  - Processor is not used for RAID calculations as a separate controller present.

A Bus is:

- shared communication link
- single set of wires used to connect multiple subsystems



- A Bus is also a fundamental tool for composing large, complex systems
  - systematic means of abstraction

Bus basics

- A bus consists of
  - A set of wires over which data and bus commands are transferred
  - *Communication protocols* for determining bus ownership, (which device can currently use the bus), bus commands, and responses to commands
- To use the bus, a device (cache, memory, or I/O) must first *arbitrate* for the usage of the bus in order to get exclusive usage for the required time period
- A bus operation may consist of sending
  - Bus commands (e.g., read, write)
  - Addresses (e.g., of the memory location to read or write)
  - Data (e.g., to be returned in response to a read command)

Interconnecting Components

- Need interconnections between
  - CPU, memory, I/O controllers
- Bus: shared communication channel
  - Parallel set of wires for data and synchronization of data transfer
  - Can become a bottleneck
- Performance limited by physical factors
  - Wire length, number of connections
- More recent alternative: high-speed serial connections with switches
  - Like networks

Bus Types

- Processor-Memory buses
  - Short, high speed
  - Design is matched to memory organization
- I/O buses
  - Longer, allowing multiple connections
  - Specified by standards for interoperability
  - Connect to processor-memory bus through a bridge

Bus Signals and Synchronization

- Data lines
  - Carry address and data
  - Multiplexed or separate
- Control lines
  - Indicate data type, synchronize transactions
- Synchronous
  - Uses a bus clock
- Asynchronous
  - Uses request/acknowledge control lines for handshaking

Advantages and Disadvantage of Buses

- **Advantages**
  - **Versatility:**
    - New devices can be added easily
    - Peripherals can be moved between computer systems that use the same bus standard
  - **Low Cost:**
    - A single set of wires is shared in multiple ways
    - Manage complexity by partitioning the design
- **Disadvantages**
  - **It creates a communication bottleneck**
    - The bandwidth of that bus can limit the maximum I/O throughput
  - **The maximum bus speed is largely limited by:**
    - The length of the bus
    - The number of devices on the bus
    - The need to support a range of devices with:
      - Widely varying latencies
      - Widely varying data transfer rates

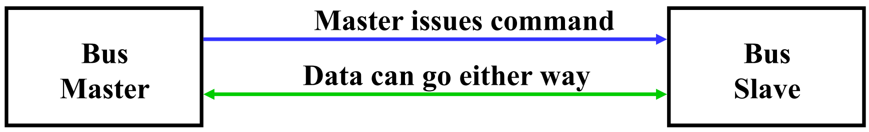
The General Organization of a Bus

- **Control lines:**
  - **Signal requests and acknowledgments**
  - **Indicate what type of information is on the data lines**
- **Data lines:**
  - **carry information between the source and the destination:**
    - Data and Addresses
    - Complex commands

- **What defines a bus?**
  - **Transaction Protocol**
  - **Timing and Signaling Specification**
  - **Bunch of Wires**
  - **Electrical Specification**
  - **Physical / Mechanical Characterisics – the connectors**

Master versus Slave

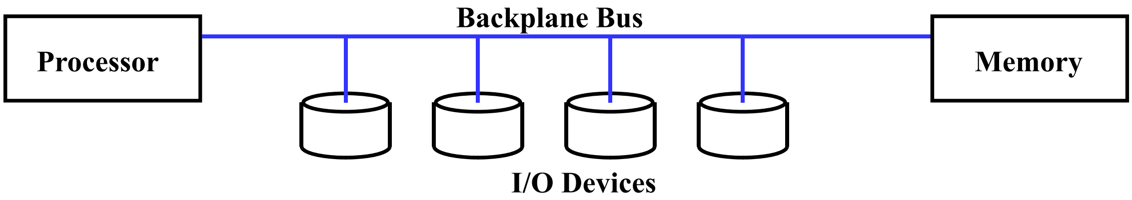
- A bus transaction includes two parts:
  - Issuing the command (and address) – request
  - Transferring the data – action
- Master is the one who starts the bus transaction by:
  - issuing the command (and address)
- Slave is the one who responds to the address by:
  - Sending data to the master if the master ask for data
  - Receiving data from the master if the master wants to send data



Types of Buses

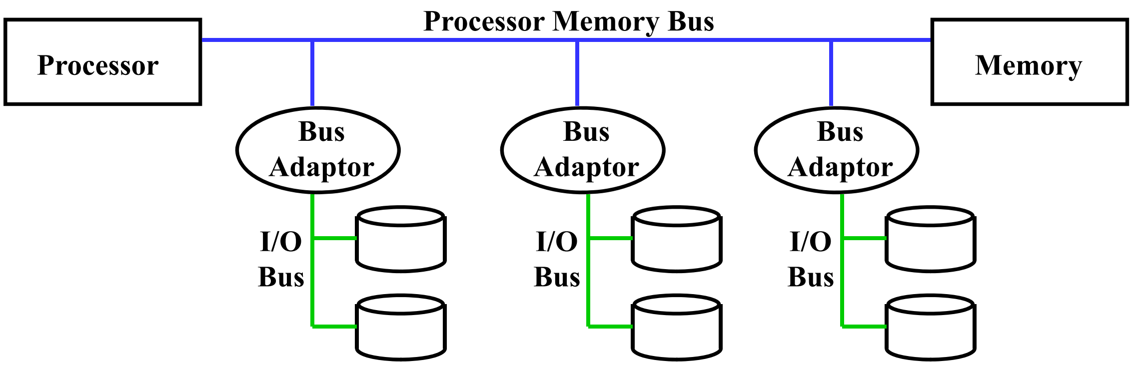
- Processor-Memory Bus (design specific)
  - Short and high speed
  - Only need to match the memory system
    - Maximize memory-to-processor bandwidth
  - Connects directly to the processor
  - Optimized for cache block transfers
- I/O Bus (industry standard)
  - Usually is lengthy and slower
  - Need to match a wide range of I/O devices
  - Connects to the processor-memory bus or backplane bus
- Backplane Bus (standard or proprietary)
  - Backplane: an interconnection structure within the chassis
  - Allow processors, memory, and I/O devices to coexist
  - Cost advantage: one bus for all components

A Computer System with One Bus: Backplane Bus



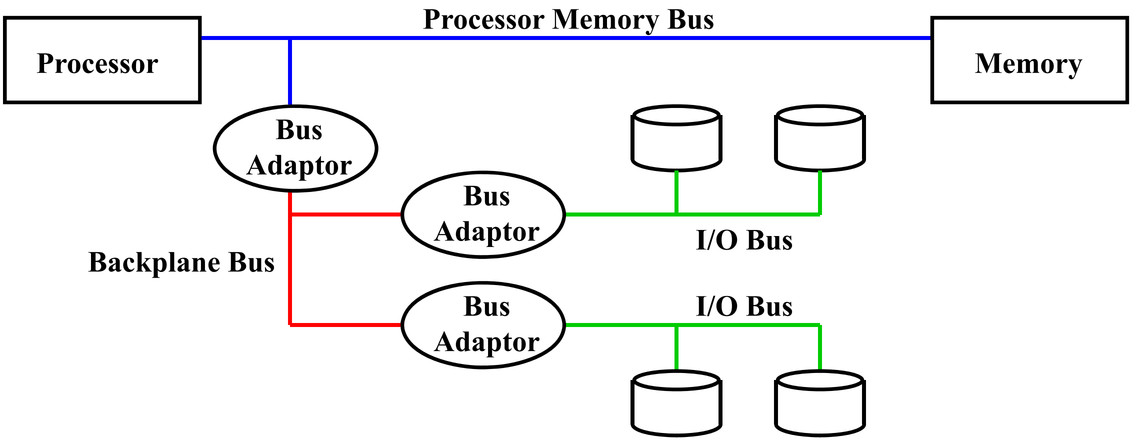
- A single bus (the backplane bus) is used for:
  – Processor to memory communication
  – Communication between I/O devices and memory
- Advantages: Simple and low cost
- Disadvantages: slow and the bus can become a major bottleneck
- Example: IBM PC - AT

A Two-Bus System



- I/O buses tap into the processor-memory bus via bus adaptors:
  – Processor-memory bus: mainly for processor-memory traffic
  – I/O buses: provide expansion slots for I/O devices
- Apple Macintosh-II
  – NuBus: Processor, memory, and a few selected I/O devices
  – SCSI Bus: the rest of the I/O devices

A Three-Bus System



- A small number of backplane buses tap into the processor-memory bus
  - Processor-memory bus is used for processor memory traffic
  - I/O buses are connected to the backplane bus
- Advantage: loading on the processor bus is greatly reduced