

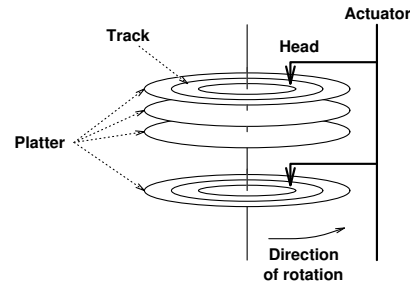
Storage Systems

I/O Systems: Fundamentals

- Input/output has traditionally been an ignored component of a computer architecture!
- Overall performance of a computer system depends on the performance of the I/O system
 - Amdahl's law!
 - Example: if CPU performance doubles but I/O performance is unchanged; how is the program performance affected?
- Two types of I/O:
 - [Storage I/O](#) (Chapter 6)
 - [Network I/O](#) (Chapter 7)

Fundamentals of Disk Storage

- Seek time
- Rotational latency
- Transfer rate
- Scheduling algorithms: FCFS, SCAN, SSTF, SATF



Disk Storage (Cont'd)

- Conventional disks: constant capacity tracks
- Current disks: **constant recording density** (or constant bit density) disks
 - Tracks have different capacities — outer track with larger capacity
 - Tracks have different data transfer rates — outer tracks with higher data transfer rate
 - Tracks are divided into **zones**; zones consist of consecutive tracks with the same capacity
- Disk performance depends on the seek time function, rotational rate, number of tracks per platter, number of platters per disk, and size of each track

Other Possibilities

- Solid state disk
 - DRAM + battery
 - Trivial access times; cost differential decreasing
 - Standard in future laptops??
- CD-ROMs
 - Potential for archival storage/transfer
- Disk still dominant
 - Represents nonvolatile storage
 - Costs/byte $< 1/20$ of DRAM

Interfacing Storage Devices to the CPU

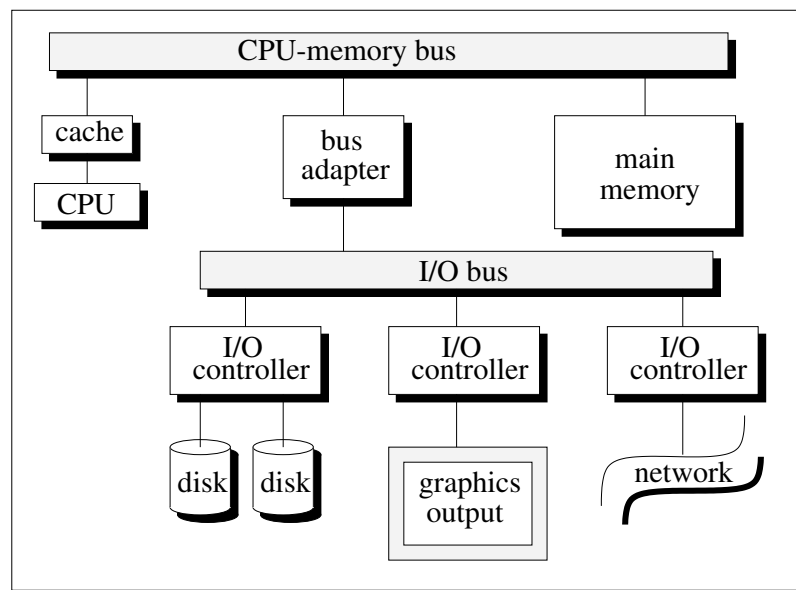
- Storage subsystem is commonly connected to the CPU/memory through a bus
- What is a bus?
 - Shared communication link between different subsystems
 - **Advantages:** simple, low cost, versatile
 - **Disadvantage:** creates a communication bottleneck — can limit the I/O throughput!
- Basic bus operation: (1) send the address of data to be accessed; and (2) receive or send data.

Design Options for a Bus

Option	High performance	Low cost
Bus width	Separate Address and data lines	Multiplex address and data lines
Data width	Wider is faster (e.g., 64 bits)	Narrow is cheaper (e.g., 8 bits)
Transfer size	Multiple words have less bus overhead	Single-word transfer is simpler
Bus masters	Multiple (requires arbitration)	Single master (no arbitration)
Split transaction?	Yes—separate request and reply packets get higher bandwidth (need multiple masters)	No—continuous connection is cheaper and has lower latency
Clocking	Synchronous	Asynchronous

Interfacing Storage Devices to the CPU

Typical architecture consists of two buses: a **CPU-memory bus** and an **I/O bus**.



Accessing I/O Devices

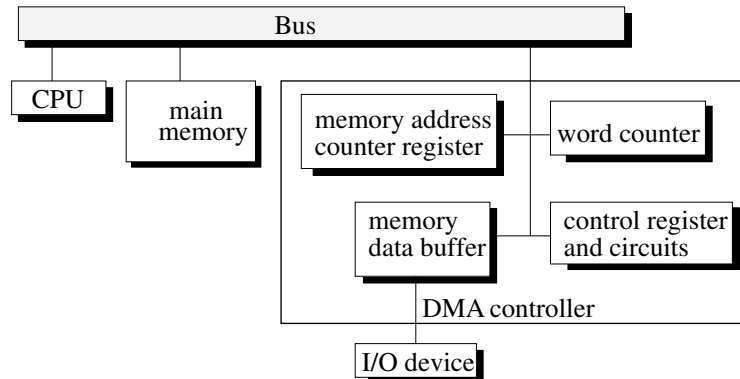
- How are I/O devices accessed?
 - Memory-mapped I/O
 - Separate I/O operations
- In both architectures, CPU needs to initiate an I/O and take actions when an I/O is completed
- Two possible implementations:
 - Programmed I/O
 - Direct Memory Access

Programmed I/O

- Two possible implementations: polling or interrupts
- Drawbacks:
 - Very slow; ties up the CPU
 - Speed mismatch between CPU and I/O device makes one of them wait
- Still, included as an option in most computers because of little hardware costs.
- Useful in small, low-speed systems.

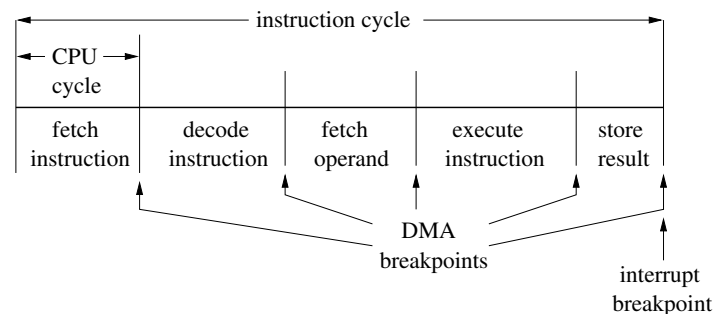
Direct Memory Access

- Dedicated hardware (DMA controller) for
 - High speed data transfer between I/O and memory
 - Minimizing involvement of CPU



DMA Access

- I/O operation initiated by CPU
 - Example: to transfer a block of data, need four instructions: (1) load MAR, (2) load word count, (3) read/write, and (4) GO. On task completion DMA informs CPU through an interrupt.
- I/O operation initiated by I/O device
 - DMA request sent to CPU
 - Request granted at the next DMA breakpoint
- DMA must have (occasional) control of system bus



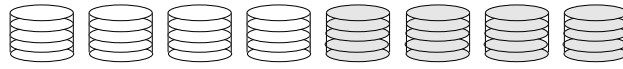
Storage Systems: Performance Trends

- Disk Components
 - density (bits/area) 2x per 3 years
 - transfer time 2x per 5 years
 - access time 2x per 10 years
- Others
 - MIPS 2x per year
 - memory size 2x per 2 years

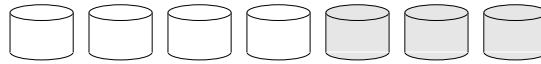
RAID

- Redundant Arrays of Inexpensive/Independent Disks
- Two orthogonal concepts
 - Data striping for improved performance
 - Redundancy for improved reliability
 - * 1 disk MTTF 23 years → 100 disk array MTTF 3 months
- Distinguish different RAIDs by:
 - Granularity of interleaving
 - * Fine grain: high latency, high transfer rate
 - * Coarse grain: for multiple small requests
 - How is redundant information computed and distributed across the disk array?

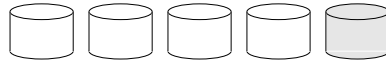
RAID Levels (for 4 disk capacity)



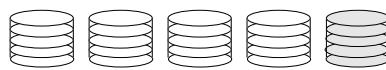
Mirrored (RAID Level 1)



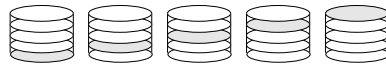
Memory-style ECC (RAID Level 2)



Bit-Interleaved Parity (RAID Level 3)



Block-Interleaved Parity (RAID Level 4)

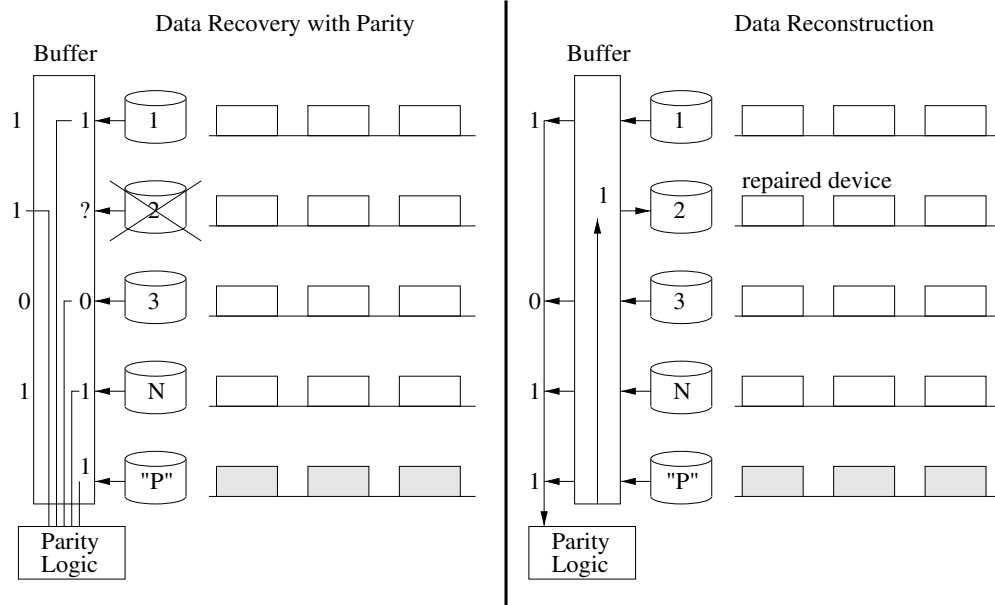


Block-Interleaved Distributed-Parity (RAID Level 5)

Popular RAID S

- **RAID-1**
 - Better availability
 - Faster reads! (choose disk set with shorter Q)
- **RAID-3**
 - Byte interleaving with spindle synch.
 - High data rate
 - Low throughput on small requests
- **RAID-5**
 - Independent actuators
 - High throughput and data rate
 - Adaptive, reliable

How Parity Works



Left Symmetric Parity Placement (RAID-5)

- Reduces disk conflicts for large requests

0	1	2	3	P0
5	6	7	P1	4
10	11	P2	8	9
15	P3	12	13	14
P4	16	17	18	19

- Each square corresponds to a stripe unit. Each column of squares corresponds to a disk.
- P0 computes the parity over stripe units 0, 1, 2, and 3; P1 computes the parity over stripe units 4, 5, 6, and 7; etc.