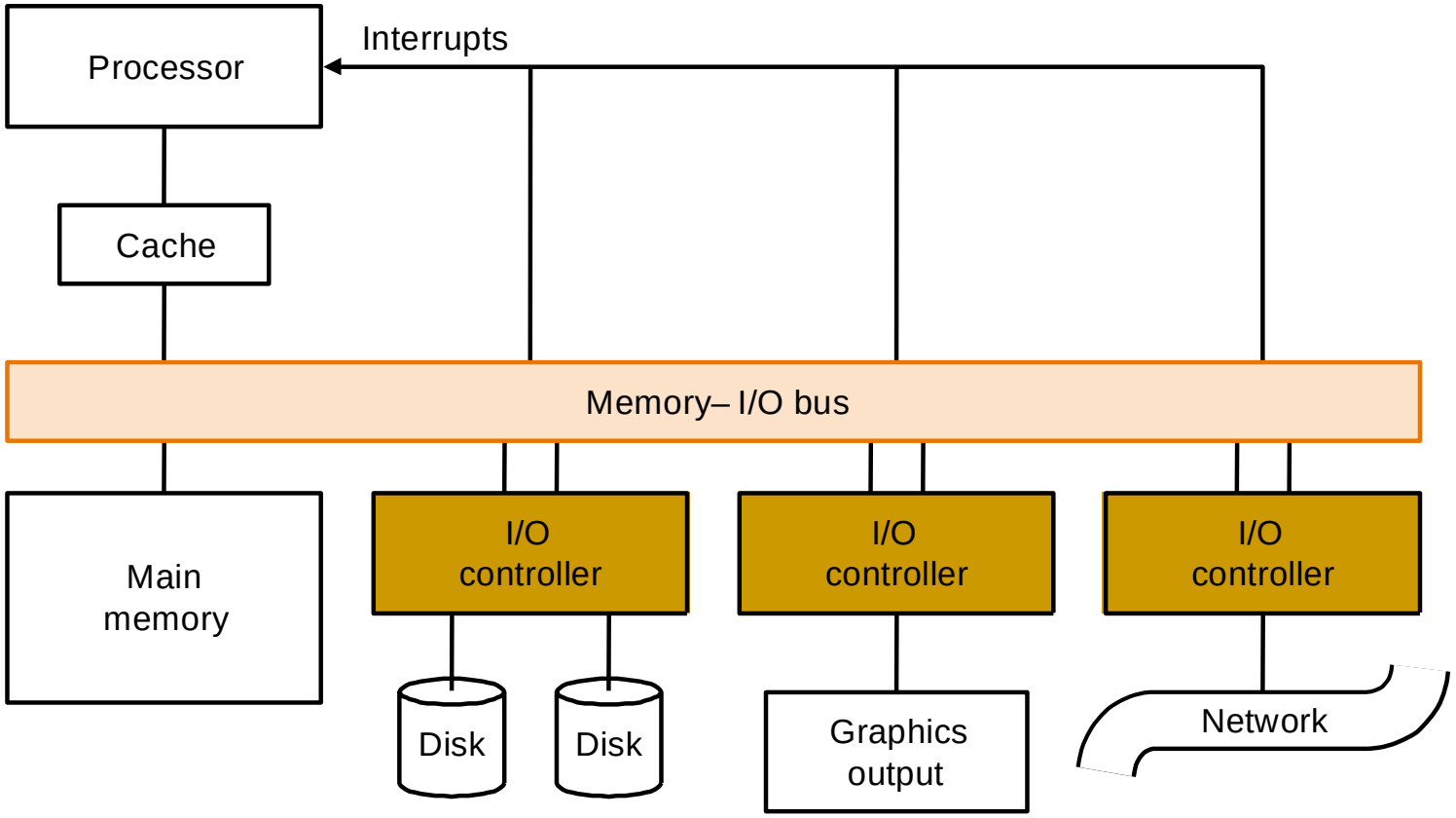
182.690 RECHNERSTRUKTUREN – INPUT/OUTPUT

Thomas Polzer
tpolzer@ecs.tuwien.ac.at
Institut für Technische Informatik

Typical Computer Architecture



Properties of I/O Devices

- Behavior
  - Input (Reading data from the environment)
  - Output (Writing data to the environment)
  - Storage (Repetitive writing and reading of data)
- Data rate
  - Transfers/second, bytes/second
- Communication partners
  - Human – machine
  - Machine - machine

Classification of Peripherals

Device	Behavior	Partner	Data rate (Mbit/sec)
Keyboard	Input	Human	0.0001
Mouse	Input	Homan	0.0038
Voice input	Input	Human	0.264
Sound input	Input	Machine	3
Scanner	Input	Human	3.2
Voice output	Output	Human	0.264
Sound output	Output	Human	8
Laser printer	Output	Human	3.2
Graphics display	Output	Human	800-8000
Cable modem	Input or output	Machine	0.128-6
Network/LAN	Input or output	Machine	100-10000
Network/WLAN	Input or output	Machine	11-54
Optical disk	Storage	Machine	80-220
Magnetic tape	Storage	Machine	5-120
Flash memory	Storage	Machine	32-200
Magnetic disk	Storage	Machine	800-3000

I/O System Characteristics

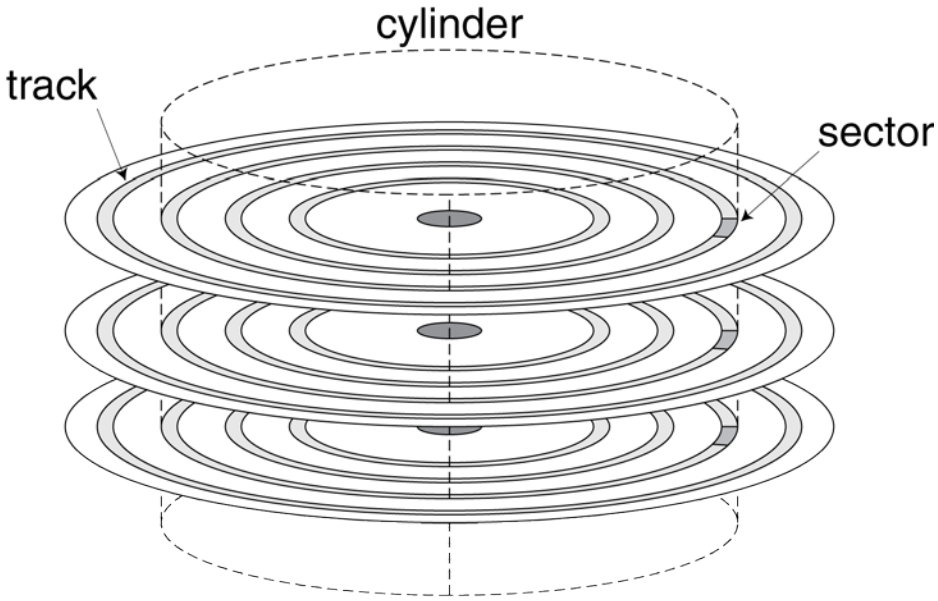
- Dependability is important
  - Particularly for storage devices
- Performance measures
  - Latency (response time)
  - Throughput (bandwidth)
- Desktops and embedded systems
  - Mainly interested in response time
- Servers
  - Mainly interested in throughput & expandability of devices

Dependability Measures

- Reliability: measure of continuous service accomplishment → mean time to failure (MTTF)
- Service interruption: mean time to repair (MTTR)
- Mean time between failure:
  - MTBF = MTTF + MTTR
- Availability: MTTF/MTBF
- Improving availability
  - Increasing MTTF: fault avoidance, fault tolerance, fault forecasting
  - Reduce MTTR: improved tools and processes for diagnostic and repair

Disk Storage

- Nonvolatile, rotating magnetic storage



Magnetic Disk

- A rotating platter coated with a magnetic surface
- A moveable read/write head to access the information on the disk
- 1 to 4 platters (each with two recordable surfaces) per disk (1” to 3.5” diameter)
- Rotational speed: 5400 to 15000 rpm
- 10000 – 50000 tracks/surface
- 100 – 500 sectors/track
  - The smallest unit read or writeable, typically between 512B and 4096B)

Disk Sectors and Access

- Each sector consists of
  - Sector ID
  - Data (512bytes – 4096bytes)
  - Error correction code (ECC, used to hide defects and record errors)
  - Synchronization fields and gaps
- Access to a sector
  - Queuing delay if other accesses are pending
  - Seek: move the heads
  - Rotational latency
  - Data transfer
  - Controller overhead

Magnetic Disk Characteristics

- Seek time: position the head over the proper track (3 – 13ms in average)
- Rotational latency: wait for the desired sector to rotate under the head (half of the 1/RPMs)
  - 0.5/5400RPM=5.6ms, 0.5/15000RPM = 2.0ms
- Transfer time: transfer a block of bits to the controller cache (70 – 125MB/s typical in 2008)
- Controller time: overhead of the controller (typ. < 0.2ms)

Disk Access Example

- 512B sectors, 15000RPM, 4ms avg. seek time, 100MB/s transfer rate, 0.2ms controller overhead, disk in idle state
- Average read time:
  - Seek time + rotational latency + transfer time + controller overhead = 5ms + ½ / (15000RPM / 60) + 512 / 100MB/s + 0.2ms = 4ms + 2ms + 0.005ms + 0.2ms = 6.2ms
- If average seek time is 1ms:
  - Average read time = 3.2ms

Disk Performance Issues

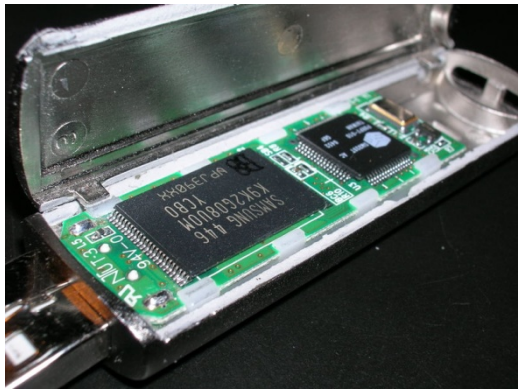
- Manufactures quote average seek time
  - Based on all possible seeks
  - Locality and OS scheduling lead to smaller actual average seek time (only 22% - 33% of quoted number)
- Smart disk controller allocate physical sectors on disk but present logical sectors to host (e.g. SATA)
- Disk drive include caches
  - Prefetching of anticipated accesses
  - Can avoid seeks and rotational delays

Write and Read Operations

- Write: Current flow within a inductor creates a magnetic field which polarized the magnetic coating of the platter
  - The direction of the current determines the polarization direction
- Read (1): The polarized bits pass by the head and induce a voltage spike into the inductor. Its polarity depends on the polarization of the bits
- Read (2): A magneto-resistive sensor in the head changes its resistance depending on the magnetic field.

Flash Storage

- Nonvolatile semiconductor storage
  - 100x – 1000x faster than disk
  - Smaller, lower power and more robust
  - But more expansive



Flash Types

- NOR flash: bit cell is like a NOR gate
  - Random read/write access
  - Used for instruction memory in embedded systems
- NAND flash: bit cell is like a NAND gate
  - Denser (more bits/area), but can only access a block at a time
  - Cheaper
  - Used for USB-sticks, media storage, …
- Flash bits wear out!
  - After 1000's of accesses
  - Not suitable to replace RAM
  - Wear leveling: remap data to less used blocks

Interconnecting Components

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

Bus Types

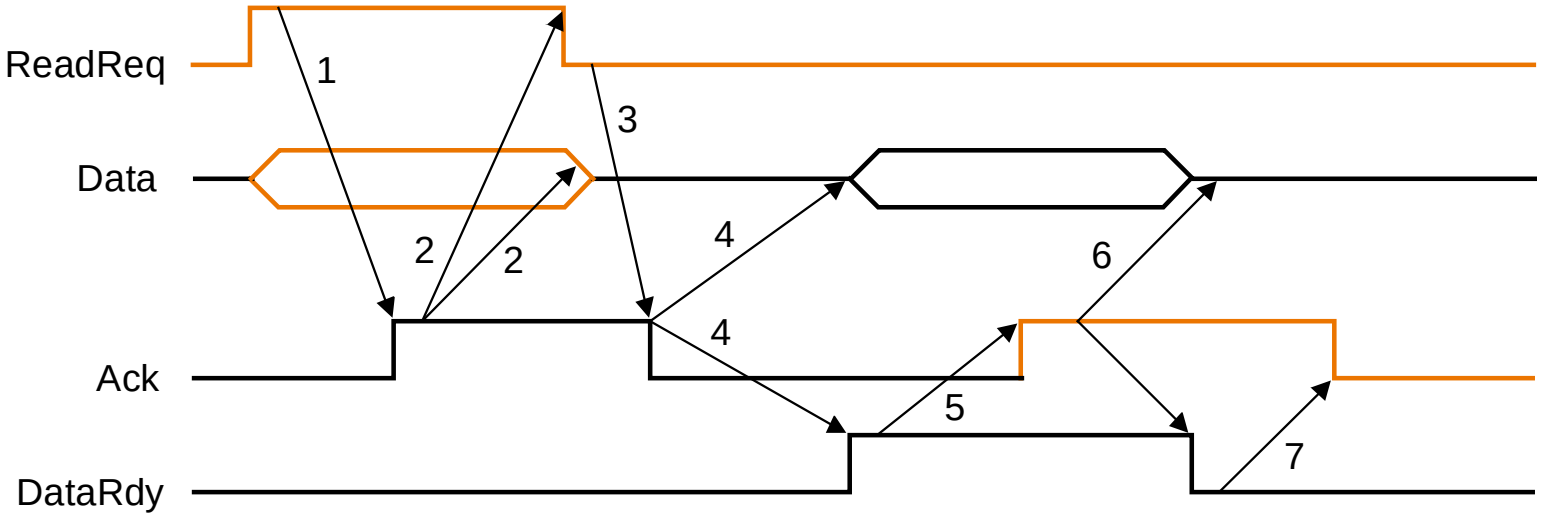
- Processor – Memory buses (synchronous)
  - Short, high speed
  - Design is matched to memory organization
- I/O buses (asynchronous)
  - Longer, allowing multiple connections
  - Specified by standards for interoperability
  - Connected to the processor – memory bus through a bridge

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 handshaking

Handshaking (Example)



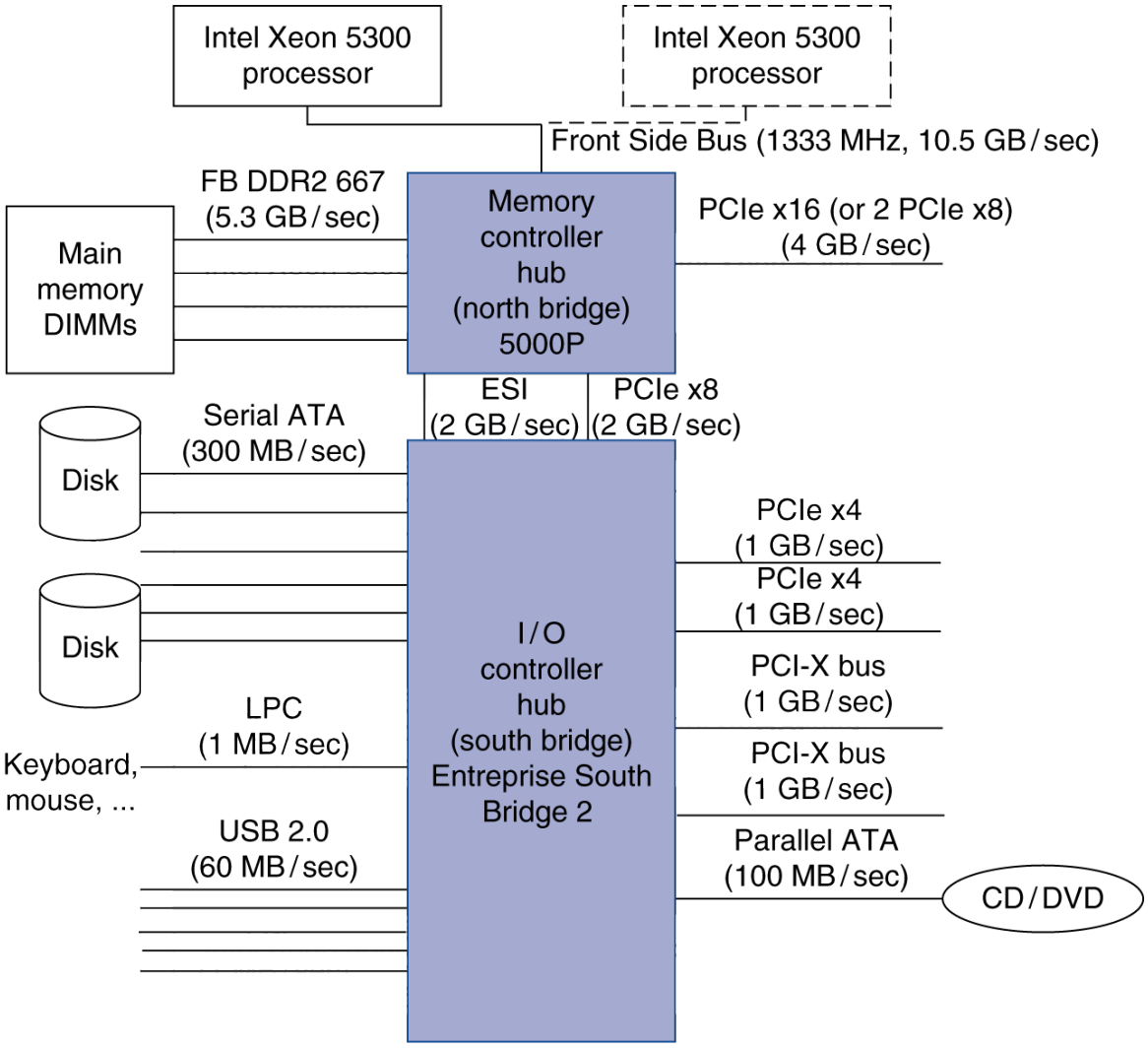
Data-output from memory to peripheral:

Read request with address (1) from peripheral, Acknowledge from memory (2), request is disabled (3), data ready from memory (4), acknowledge from peripheral (5), disable data ready (6), disable acknowledge (7)

	Firewire	USB 2.0	PCI Express	Serial ATA	Serial Attached SCSI
Intended use	External	External	Internal	Int./External	External
Devices per channel	63	127	1	1	4
Data width	4	2	2/lane	4	4
Peak bandwidth	50MB/s or 100MB/s	0.2MB/s, 1.5MB/s, or 60MB/s	250MB/s/lane 1×, 2×, 4×, 8×, 16×, 32×	300MB/s	300MB/s
Hot pluggable	Yes	Yes	Depends	Yes	Yes
Max length	4.5m	5m	0.5m	1m	8m
Standard	IEEE 1394	USB Implementers Forum	PCI-SIG	SATA-IO	INCITS TC T10

USB 3.0
> 4.8 Gb/s

Typical x86 PC I/O System



I/O Management

- I/O is mediated by the OS
  - Multiple programs share I/O resources
  - I/O causes asynchronous interrupts
    - Same mechanism as exceptions
  - I/O programming is fiddly
    - OS provides abstractions to programs

I/O Register Mapping

- Memory Mapped I/O
  - Registers are addressed in the same space as memory
  - Address decoder distinguishes between them
  - OS uses address translation to make them only accessible by the kernel
- I/O Instructions
  - Separate instructions to access I/O registers
  - Can only be executed in kernel mode

Polling

- Periodically check I/O status register
  - If device is ready, do operation
  - If error, handle it
- Common in small or low-performance real-time embedded systems
  - Predictable timing
  - Low hardware cost
- In other systems
  - Waste of CPU time

Interrupts

- When a device is ready or an error occurs
  - Controller notifies the CPU
- Interrupt comparable to exception
  - But not synchronized to instruction execution
  - Can invoke handler between instructions
  - Cause (register) information identifies the interrupting device
- Priority interrupts
  - Device needing more urgent attention get higher priority
  - Can preempt handler of lower priority

I/O Data Transfer

- Polling and interrupt-driven I/O
  - CPU transfers data between memory and I/O data registers
  - (CPU) time consuming for high-speed devices
- Direct memory access (DMA)
  - OS provides starting address in memory
  - I/O controller transfers to/from memory autonomously
  - Controller interrupts on completion or error
  - Keep cache coherence in mind!

Measuring I/O Performance

- I/O performance depends on
  - Hardware: CPU, memory, controllers, buses
  - Software: operating system, database, management system, application
  - Workload: request rates and patterns
- I/O system design can trade-off between response time and throughput
  - Measurements of throughput often done with constrained response time

I/O vs. CPU Performance (Example)

- Benchmark takes 90s CPU time, 10s I/O time
- Double the number of CPUs every two years
  - I/O unchanged
- How does the performance of the program changes over the years?

Year	CPU time	I/O time	Elapsed time	% I/O time
Now	90s	10s	100s	10%
+2	45s	10s	55s	18%
+4	23s	10s	33s	31%
+6	11s	10s	21s	47%
+8	6s	10s	16s	63%

RAID

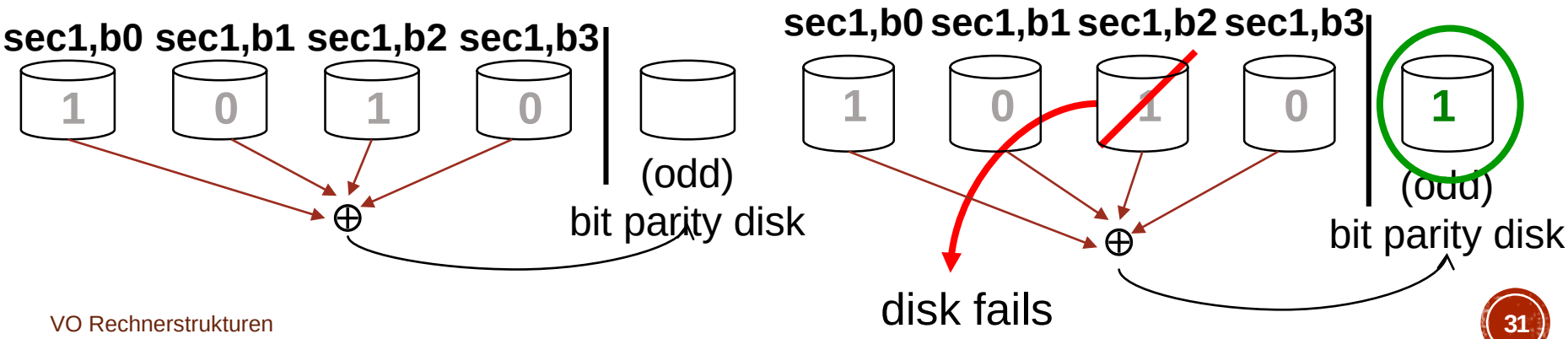
- Redundant Array of Inexpensive (Independent) Disks
  - Use multiple smaller disk (instead of one large disk)
  - Parallelism improves performance
  - Plus extra disk(s) for redundant data storage
- Provides fault tolerant storage system
  - Especially if failed disks can be hot swapped

RAID Levels

- RAID 0: Striping
  - No redundancy! Just stripe data over multiple disks
  - Improves performance
- RAID 1: Mirroring
  - N+N disks, replicate data
  - Write data to both disks (mirror disks)
  - On disk failure read from mirror
- RAID 2: 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 Levels

- RAID 3: Bit-Interleaved Parity
  - N + 1 disks
  - Data striped across N disks at byte level
  - Redundant disk stores parity
  - Read from all disks
  - Write updates all disks and writes parity
  - On failure use parity to reconstruct missing data
  - Not widely used

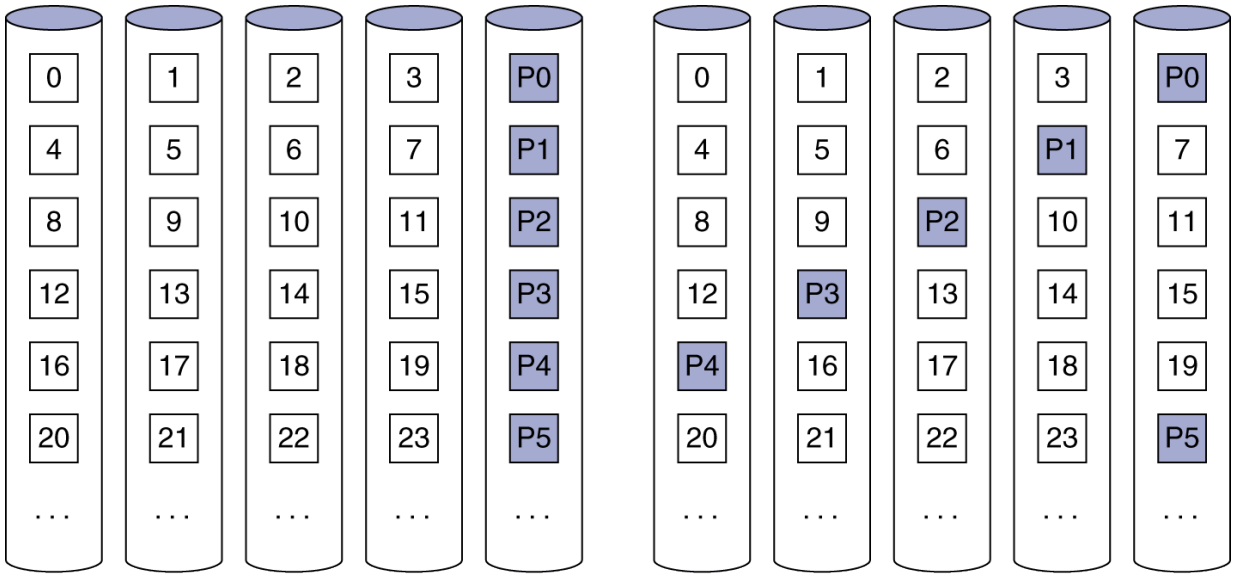


RAID Levels

- 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 only disk holding the required block
  - Write just disk containing modified block and parity disk
  - On failure use parity to reconstruct missing data
  - Not widely used

RAID Levels

- RAID 5: Distributed Parity
  - N + 1 disks
  - Like RAID 4 but parity blocks distributed across disks
  - Avoids parity disk being a bottle neck
  - Widely used!



RAID 4 RAID 5

RAID Levels & Summary

- RAID 6: P + Q Redundancy
  - N + 2 disks
  - Like RAID 5, but two lots of parity
  - Greater fault tolerance through more redundancy

- Summary
  - RAID can improve performance and availability (high availability requires hot swapping)
  - Assumes independent disk failures

Design of an I/O System

- Satisfying latency requirements
  - For time-critical operations
- Maximizing throughput
  - Find weakest link (lowest bandwidth component)
  - Configure to operate at maximum bandwidth
  - Balance remaining components
- If system is loaded, simple analysis is insufficient
  - Requires model of queuing behavior
  - Simulations necessary

Fallacy: Disk Dependability

- If a disk manufacturer quotes a MTTF as 1200000hr (140yr)
- In average a disk will not work that long!!
  - What is the distribution of failures
  - What if you have 1000 disks, how many will fail per year?
- Annual failure rate:

  (1000 disk x 8760 h/disk)/(12000000 hrs/failure) =

  0.73%

Pitfall: Offloading to I/O Processors

- Offloading to I/O processors and expecting performance improvements
  - Overhead of managing I/O processor requests may dominate!
  - Quicker to do small operations on the CPU, but I/O design may prevent that
  - I/O processer may be slower (since it is of simpler design)
  - Making it faster converts it into a major system component and might need its own coprocessor!

Fallacy: Disk Scheduling

- OS scheduling of disk access
  - But modern drive deal with logical block addresses
  - Map to physical track, sector location and blocks may be cached by the drive
  - OS is therefore unaware of physical location and may reduce performance due to erroneous reordering of requests

Pitfall: Peak Performance

- Peak I/O rates are nearly impossible to achieve
  - Usually some other system components limits the performance
  - E.g., transfer to memory over a bus
    - Collision with DRAM refresh
    - Arbitration contention with other bus masters
  - E.g., PCI bus: peak bandwidth ~133 MB/sec
    - In practice: max. 80MB/sec sustainable

Conclusion

- I/O performance measures
  - Throughput, latency
  - Dependability and cost also important
- Busses used to connect CPU, memory, I/O controllers
  - Polling, interrupts, DMA
- RAID
  - Improves performance and dependability

EXAMPLES

I/O System Design

- Given a Sun Fire x4150 system with
  - Workload: 64 KB disk reads (idle disks, RAID no bottleneck), Assumption: Each I/O operation requires 200000 user-code instructions and 100000 OS instructions
- Each CPU: 1 Giga-instructions/sec
- FrontSideBus: 10.6 GB/sec peak
- DRAM DDR2 667MHz: 5.336 GB/sec
- PCI-E 8x bus: 8 x 250 MB/sec = 2 GB/sec
- Disks: 15000rpm, 2.9ms avg. seek time, 112 MB/se transfer rate
- What I/O Rate can be sustained for random and for sequential reads?

I/O System Design

- I/O rate for CPUs
  - Per core: 10^9/(100000 + 200000) = 3333 I/Os/sec
  - 8 cores: 8x3333IOPs → 26667 ops/sec
- Random reads, I/O rate for disks
  - Assume actual seek time is average/4
  - Time/op = seek + latency + transfer = 2.9ms/4 + 4ms/2 + 84KB/(112MB/2) = 3.3ms
  - 1000/3.3 → 303 ops/sec per disk = 2424 ops/sec for 8 disks
- Sequential reads
  - 112 MB/s / 64KB = 1750 ops/sec per disk
  - → 14000 ops/sec for 8 disks

I/O System Design

- PCI-E I/O rate
  - 2GB/sec / 64KB = 31250 ops/sec
- DRAM I/O rate
  - 5336 GB/sec / 64KB = 83375 ops/sec
- FSB I/O rate
  - Assume we can sustain half the peak rate
  - 5.3 GB/sec / 64KB = 81540 ops/sec per FSB
  - 163080 ops/sec for 2 FSBs
- Weakest link: disks
  - 2424 ops/sec random, 14000 ops/sec sequential
  - Other components have enough headroom to accommodate these rates!

182.690 RECHNERSTRUKTUREN – INPUT/OUTPUT

Thomas Polzer
tpolzer@ecs.tuwien.ac.at
Institut für Technische Informatik