Technische Grundlagen der Informatik 2

Rechnerorganisation

Kapitel 8:
Sekundärspeicher, Netzwerke und andere Peripheriegeräte

Prof. Dr. Ben Juurlink

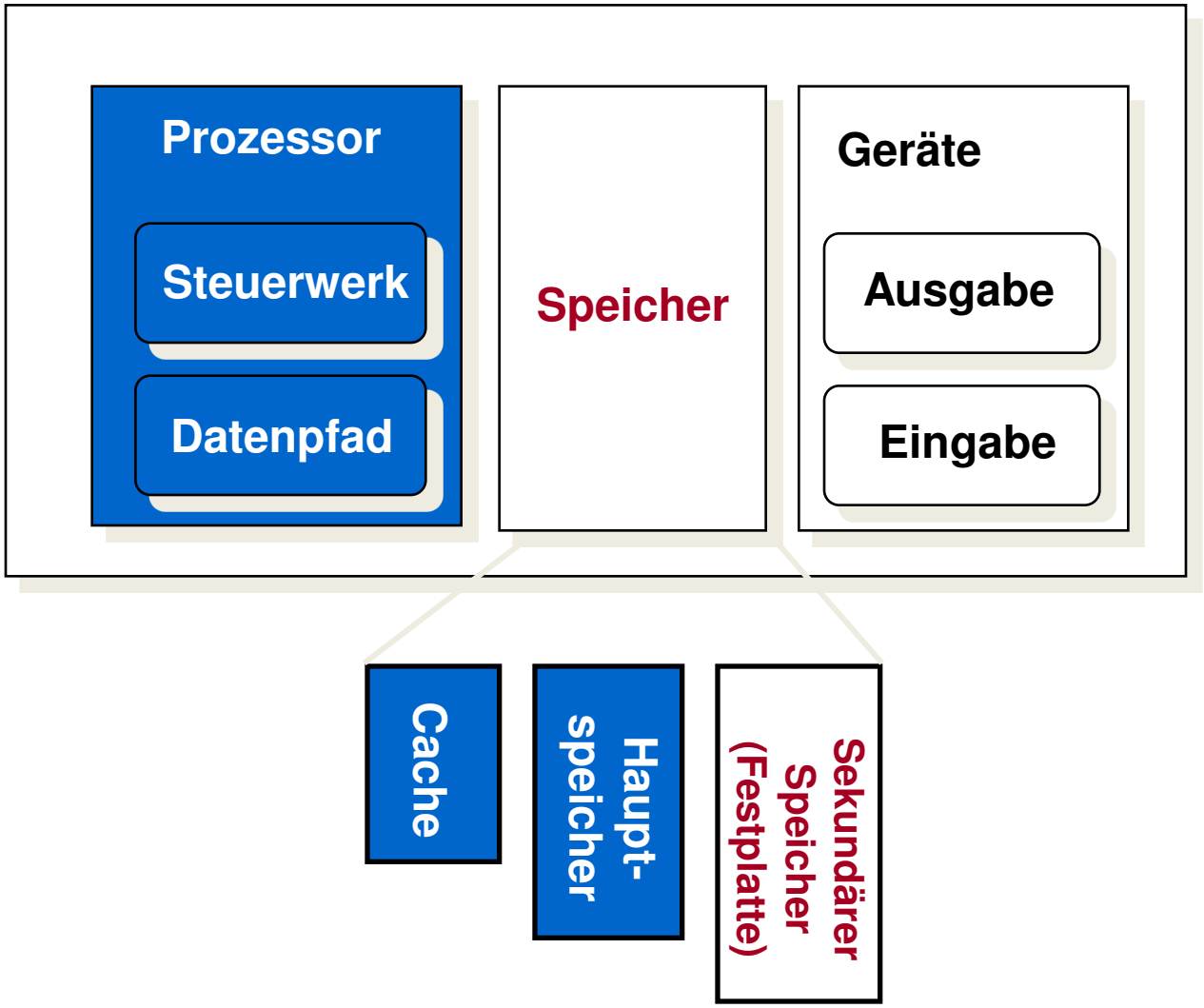
Fachgebiet: Architektur eingebetteter Systeme
Institut für Technische Informatik und Mikroelektronik
Fak. IV – Elektrotechnik und Informatik

SS 2012

Ziele

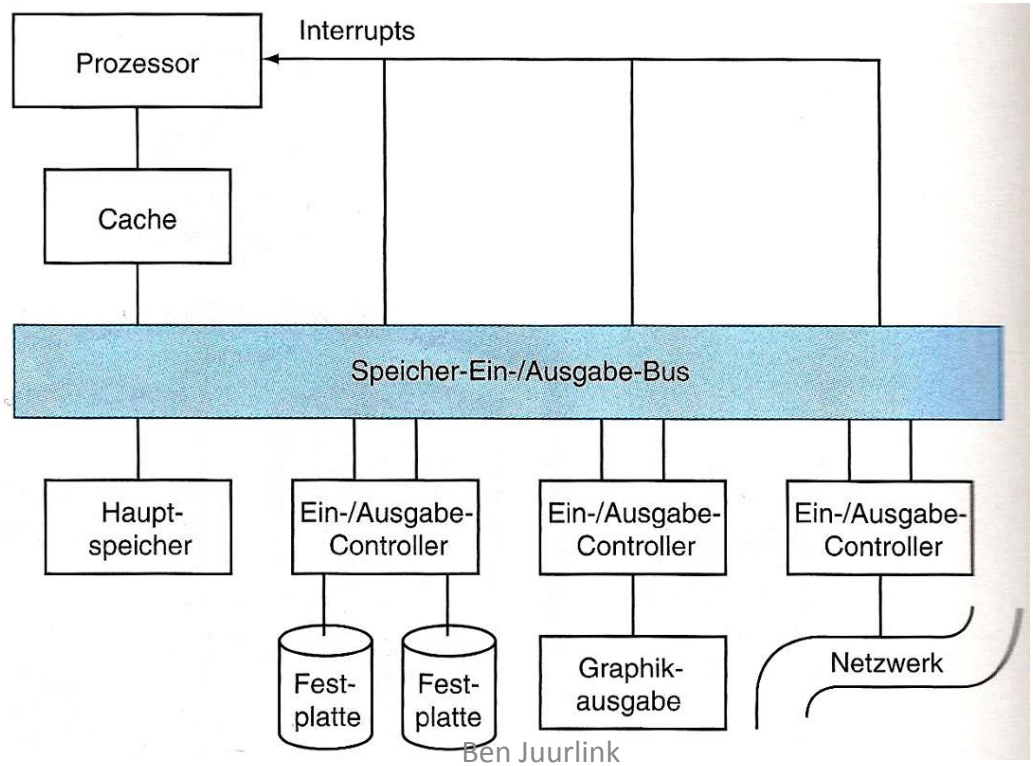
- Nach dieser Vorlesung sind Sie in der Lage,
  - den allgemeinen Aufbau von Festplattenspeicher zu beschreiben
  - Zugriffszeiten auf Festplattenspeicher zu berechnen
  - RAID-Phasen mitsamt ihre Vor- und Nachteile zu beschreiben
  - verschiedene Bussysteme zu beschreiben
  - das Handshake-Protokolls eines Bus-Systems zu beschreiben
  - Interrupt-gesteuerte Eingabe zu beschreiben
  - Ein-/Ausgabe-Leistung eines Systems zu berechnen

Hauptkomponenten eines Computers

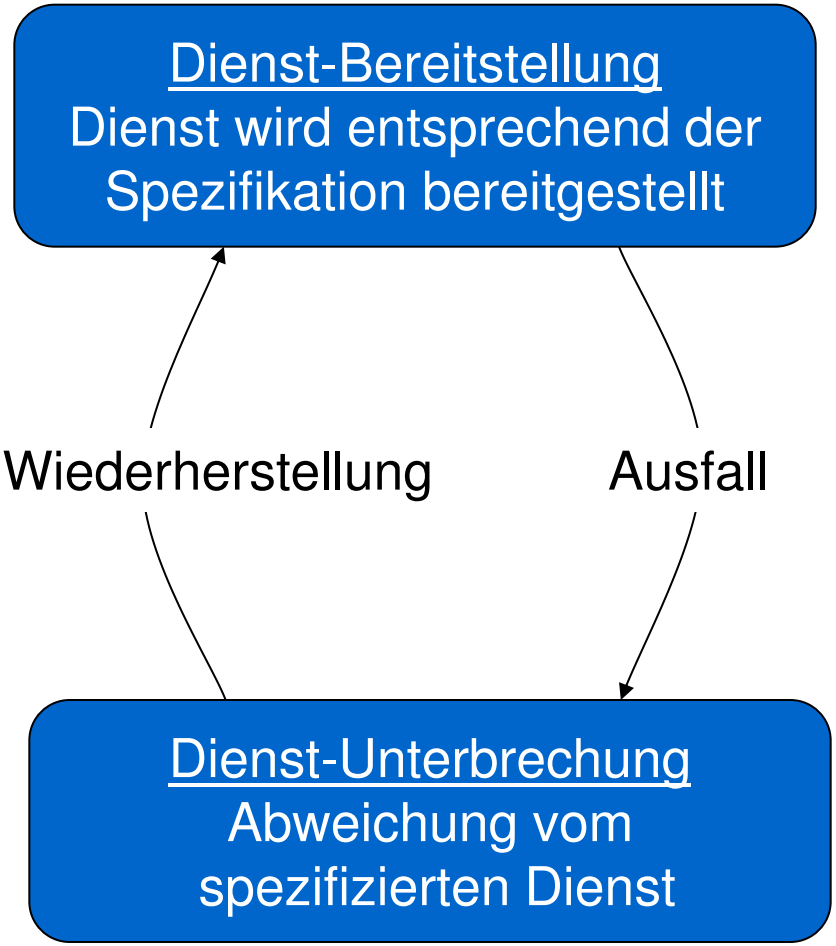


Ein-/Ausgabegeräte

- Ein-/Ausgabe-Geräte können unterteilt werden nach:
  - Verhalten: Eingabe, Ausgabe, Speicher
  - Partner: Mensch oder Maschine
  - Datenrate: Bytes/s, Übertragungen/s
- Tastatur: Verhalten = Eingabe, Partner = Mensch, Datenrate = ± 10B/s



- Zuverlässigkeit ist wichtig
  - Besonders für Speichergeräte
- Leistungsmerkmale
  - Latenz (Antwortzeit)
  - Durchsatz (Bandbreite)
- Desktops & Eingebette Systeme
  - Stark interessiert an Antwortzeit & Vielfältigkeit von Geräten
- Server
  - Stark interessiert an Durchsatz & Erweiterbarkeit von Geräten



- Fehler: Ausfall einer Komponente
  - Kann zum Systemausfall führen, muss aber nicht

Maße für Zuverlässigkeit

- Zuverlässigkeit: mittlere Zeit bis Ausfall (MTTF, *mean time to failure*)
- Dienst-Unterbrechung: mittlere Zeit bis Reparatur (MTTR, *mean time to repair*)
- Zeit zwischen Ausfällen (MTBF, *mean time between failures*)
  - MTBF = MTTF + MTTR
- Verfügbarkeit = MTTF / (MTTF + MTTR)
- Verfügbarkeit verbessern:
  - Erhöhe MTTF: Fehlervermeidung, Fehlertoleranz, Fehlervorhersage
  - Reduziere MTTR: bessere Werkzeuge für Fehlererkennung, Diagnose und Reparatur

MTBF Beispiel

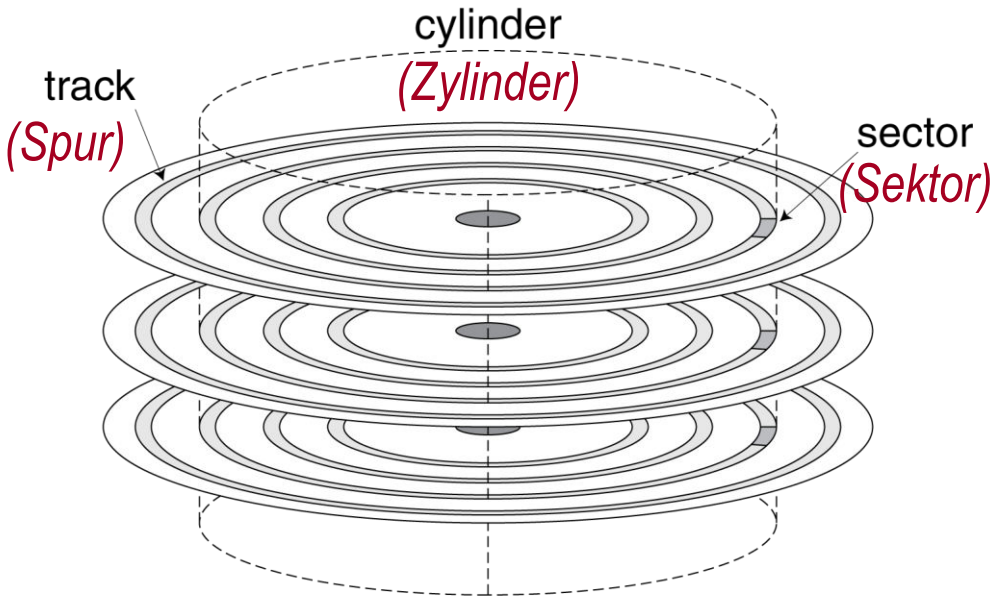
- MTBF wird oft genutzt, wenn MTTF angebrachter wäre

- MTBF(Festplatte A) = 1,000,000h
- MTBF(Festplatte B) = 500,000h
- Was ist die MTBF eines System mit den Festplatten A und B?

- Festplatte A: 1 Ausfälle in 1,000,000h
- Festplatte B: 2 Ausfälle in 1,000,000h
- Festplatten A + B: 3 Ausfälle in 1,000,000h
  - MTBF(Festplatten A + B) = 333,333h =

$$\frac{1}{\frac{1}{1{,}000{,}000} + \frac{1}{500{,}000}}$$

- nichtflüchtiger, rotierender Magnetspeicher



- 1 bis 4 Platten, 1 bis 3,5 Zoll Durchmesser
- jeweils 2 beschreibbare, magnetische Oberfläche
- unterteilt in konzentrische Kreise genannt Spuren (10K-50K)
- jede Spur 100-500 Sektoren
  - in der Regel 512 Byte groß
  - besteht aus
    - Sektornummer
    - Fehlerkorrekturcode
      (ECC, *Error correcting code*)
    - eigentliche Daten
    - Lücken zur Synchronisation

Festplattenzugriffszeit

- Komponente der Festplattenzugriffszeit:
  - Wartezeit, bis andere Zugriffe erledigt sind
  - Suchzeit (*seek time*) für Kopf-Positionierung (*Seek*): Lese-/ Schreibkopf über richtige Spur positionieren
  - Umdrehungslatenz (*rotational latency*): warten bis gesuchter Sektor unter Lese-/Schreibkopf rotiert
  - Transferzeit: Sektorgröße / Transferrate
  - (Festplatten)Controllerzeit

Beispiel Festplattenzugriff

- Angaben Festplattenherstellers:
  - 15000 Umdrehungen/Min
  - 4ms durchschnittl. Suchzeit
  - 100MB/s Transferrate
  - 0.2ms Controllerzeit
- Wie lange dauert das Lesen eines 512 Byte Sektors, wenn Festplatte nicht beschäftigt ist?
  - 4ms (Suchzeit)
    + 0.5 / (15000/60) (= 2ms Umdrehungslatenz)
    + 512 / 100MB/s (= 0.005ms Transferzeit)
    + 0.2ms (Controllerzeit)
    = 6.2ms
- Wenn gemessene durchschnittl. Suchzeit nur 1ms
  - Durchschnittl. Zugriffszeit = 3.2ms

Beispiele für Magnetfestplatten

Eigenschaft	Seagate ST37	Seagate ST32	Seagate ST94
Plattendurchm. (Zoll)	3.5	3.5	2.5
Kapazität (GB)	73.4	200	40
# Oberflächen (Köpfe)	8	4	2
Umdrehungsgeschw. (U/m)	15,000	7,200	5,400
Transferrate (MB/s)	57-86	32-58	34
Minimale Suchzeit (ms)	0.2r-0.4w	1.0r-1.2w	1.5r-2.0w
Durchschnittl. Suchzeit (ms)	3.6r-3.9w	8.5r-9.5w	12r-14w
MTTF (h@25°C)	1,200,000	600,000	330,000
Maße (cm)	2,5 x 10,2 x 14,7	2,5 x 10,2 x 14,7	1,0 x 6,9 x 9,9
GB/cm³	0,19	0,53	0,57
Leistung: op/idle/sb (W)	20?/12/-	12/8/1	2.4/1/0.4
GB/W	4	16	17
Gewicht (g)	862	635	90,7
Preis 2004 / $/GB	$400, $5/GB	$100, $0,5/GB	$100, $2,5/GB

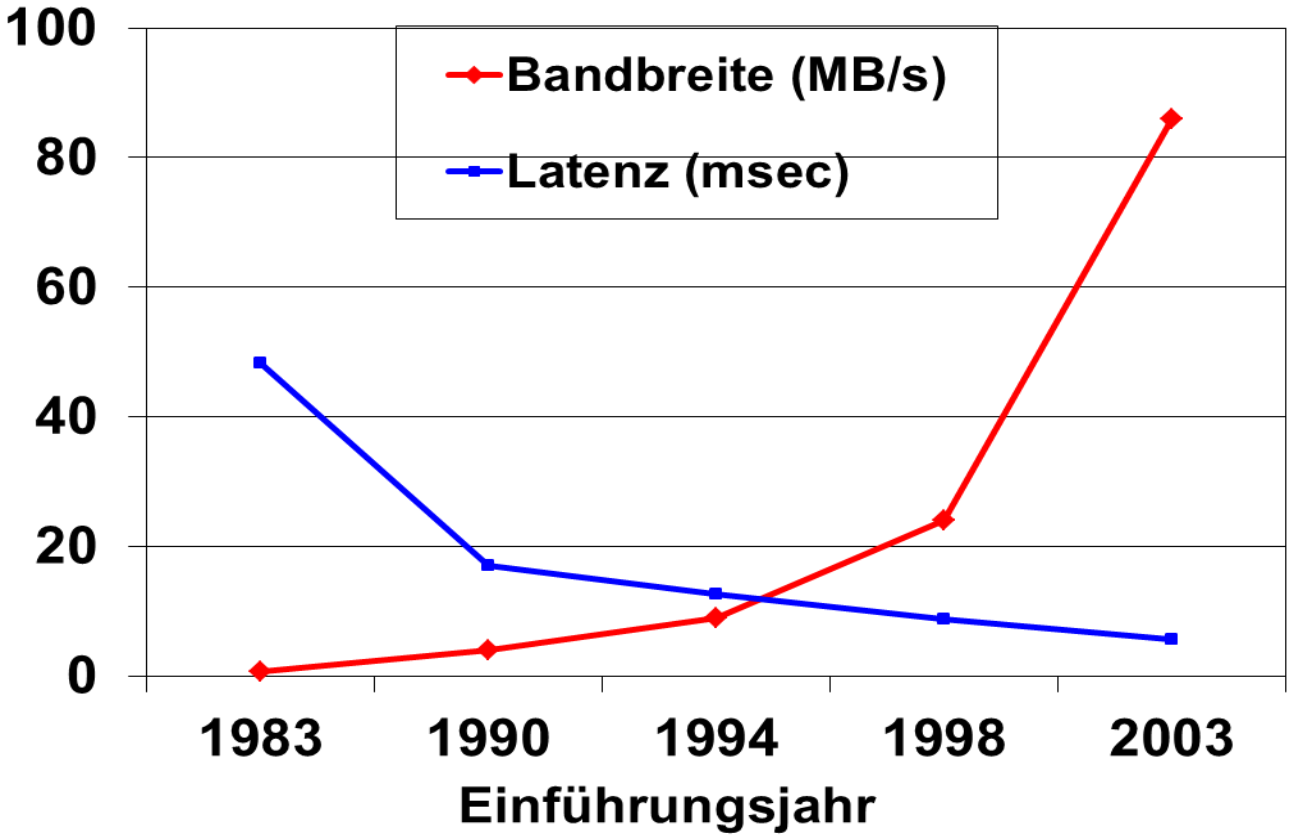
Festplattenlatenz- und Bandbreitenmeilensteine

	CDC Wren	SG ST41	SG ST15	SG ST39	SG ST37
Rot.-geschw. (rpm)	3600	5400	7200	10000	15000
Jahr	1983	1990	1994	1998	2003
Kapazität (Gbytes)	0.03	1.4	4.3	9.1	73.4
Durchm. (inches)	5.25	5.25	3.5	3.0	2.5
Anschluss	ST-412	SCSI	SCSI	SCSI	SCSI
Bandbreite (MB/s)	0.6	4	9	24	86
Latenz (msec)	48.3	17.1	12.7	8.8	5.7

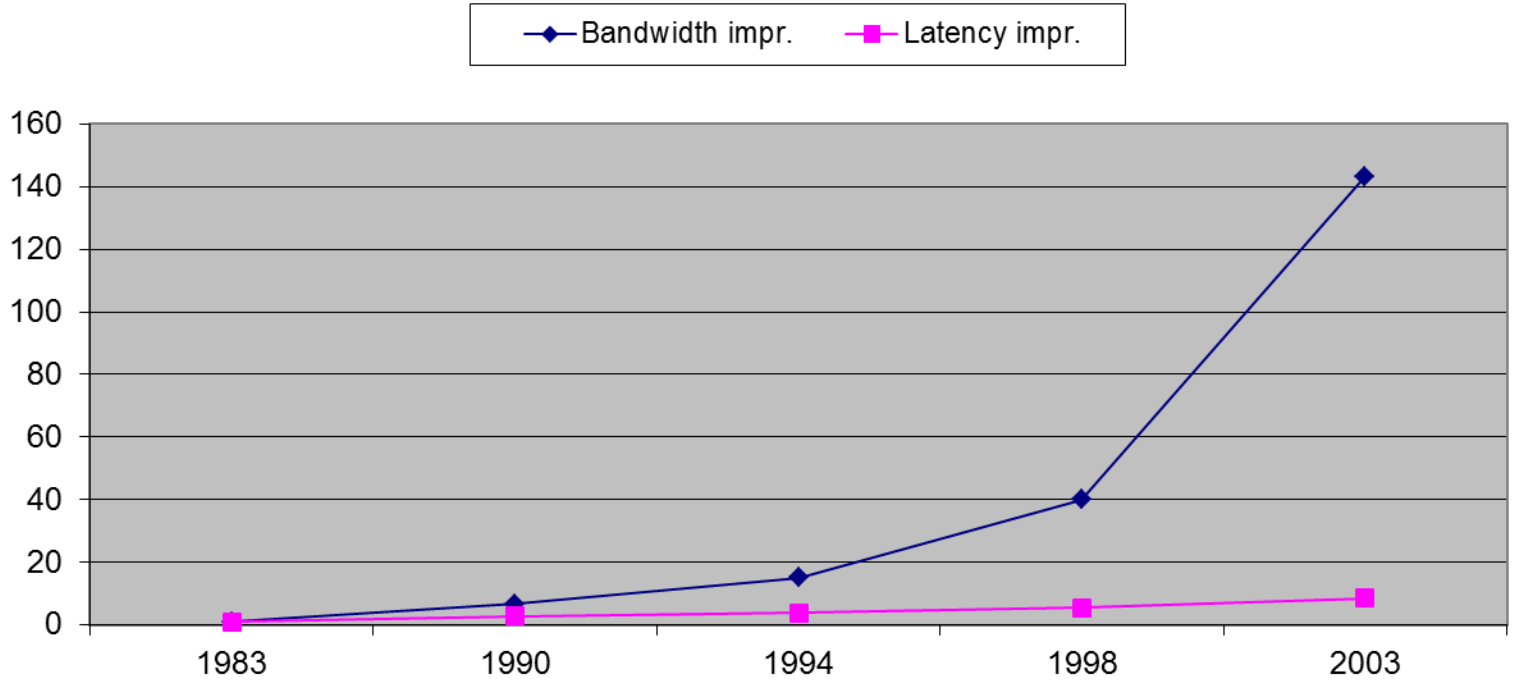
Patterson, CACM Vol 47, #10, 2004

- Festplattenlatenz = durchschnittl. Suchzeit + Umdrehungslatenz
- Festplattenbandbreite = höchste Transferrate von formatierten Daten vom Medium (nicht aus dem Cache).
- SCSI = small computer system interface = Standard für Ein-/ Ausgabegeräte

- In der Zeit in der sich die Bandbreite verdoppelt hat, verbesserte sich die Latenz nur um 1.2x bis 1.4x

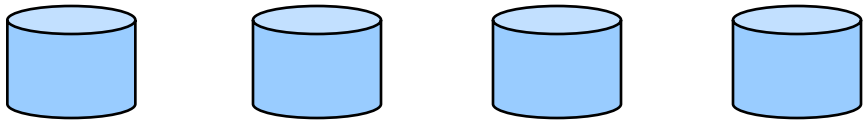


- In der Zeit in der sich die Bandbreite verdoppelt hat, verbesserte sich die Latenz nur um 1.2x bis 1.4x



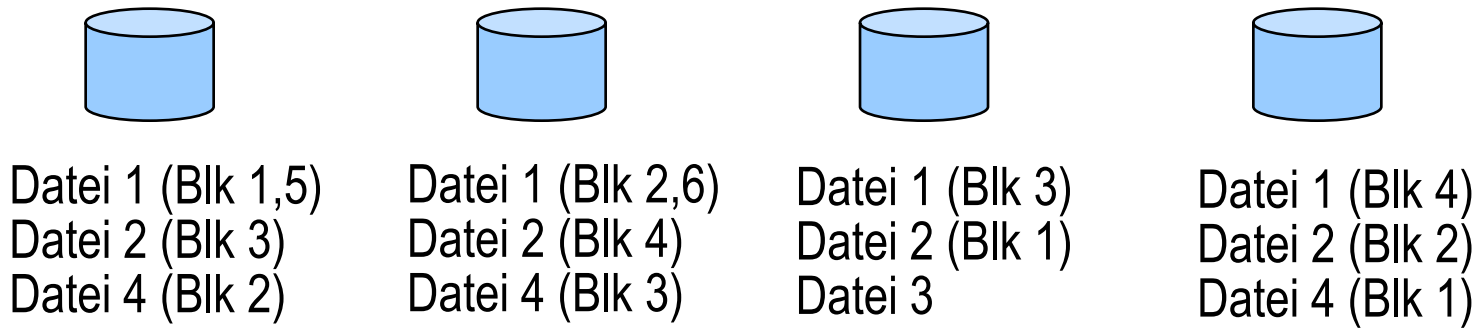
- Bandbreitenansprüche
  - Video in hoher Qualität
    - (30 frames/s) × (640 x 480 pixels) × (24-b color/pixel) = 221 Mb/s (27.625 MB/s)
    - Full High Definition: 1920×1080 = 7.75x größer
  - Audio in hoher Qualität
    - (44,100 Audio Spuren/s) × (16-b Audio Spuren) × (2 Audio Kanäle für Stereo) = 1.4 Mb/s (0.175 MB/s)
  - Kompression reduziert die Bandbreitenansprüche erheblich
- Latenzanforderungen
  - Wie empfindlich sind Augen/Ohren bei Variationen in Audio und Video Raten?
  - Wie stellt man konstante Bereitstellungsraten sicher?
  - Wie wichtig ist es, Audio und Video Übertragungen zu synchronisieren?
    - 15 - 20 ms früher bis 30 - 40 ms später tolerierbar

- RAID = Redundant Array of Inexpensive (Independent) Disks
  - Anordnung von Festplatten um Leistung und Zuverlässigkeit zu erhöhen
  - Verwendet mehrere kleine Festplatten anstatt einer großen
  - Durch mehrere Lese-/Schreibköpfe gleichzeitig lesen/schreiben
  - Zusätzliche Festplatte(n) für Redundanz
- Es gibt mehrere RAID-Phasen (RAID levels)

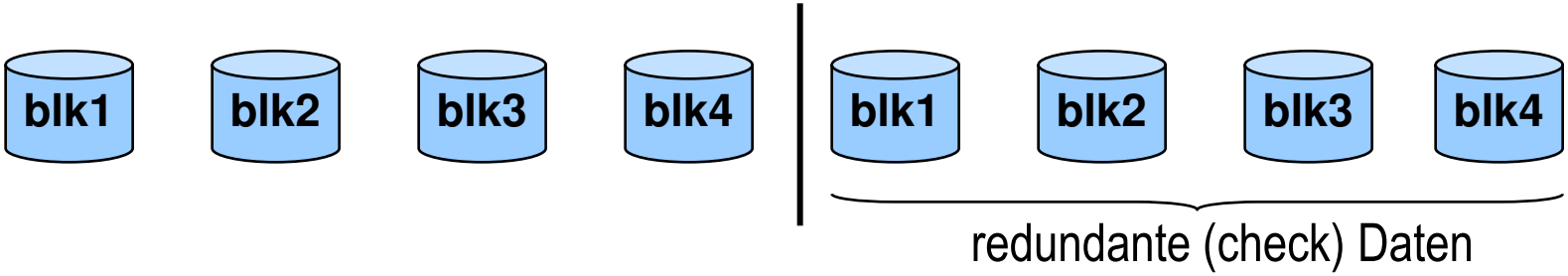


RAID 0

- Keine Redundanz (“AID”?)
  - Mit steigender Festplattenanzahl höhere Ausfallwahrscheinlichkeit
- Leistungssteigerung durch Verteilen von Daten auf mehrere Festplatten (Striping)
  - Paralleler Zugriff auf alle Festplatten
- Beispiel:
  - Datei 1: 6 Blöcke; Datei 2: 4 Blöcke; Datei 3: 1 Block; Datei 4: 3 Blöcke



- Spiegeln (*Mirroring*)
  - N + N Festplatten, Daten werden repliziert
    - Daten werden sowohl auf Festplatte als „Spiegelfestplatte“ geschrieben
    - Fällt eine aus wird von der anderen gelesen



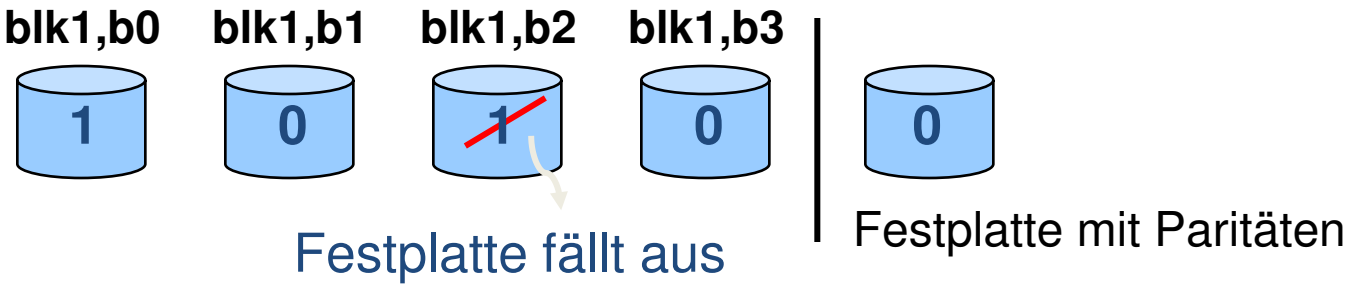
RAID 2

- Fehler erkennender und korrigierender Code (*Error correcting Code, ECC*)
  - N + E Festplatten (z.B., 10 + 4)
  - Für jede N-Bit Bitfolge wird E-Bit ECC erzeugt
  - Einzelne Bits einer (N+E)-Bit Bitfolge werden über einzelne Platten aufgeteilt
  - Zu komplex, wird nicht länger verwendet

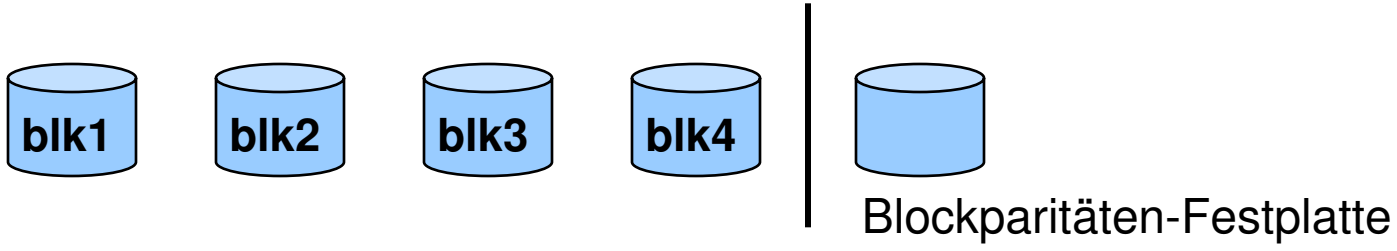
- Bitweise verschränkte Parität
- Für jede Schutzgruppe von N Platten eine redundante Platte
  - Redundante Festplatte speichert Paritäten
    - 0 wenn Bitsumme gerade (0101 0)
    - 1 wenn Bitsumme ungerade (1011 1)
    - Parität ist XOR der Bits

} Summe + Parität immer gerade

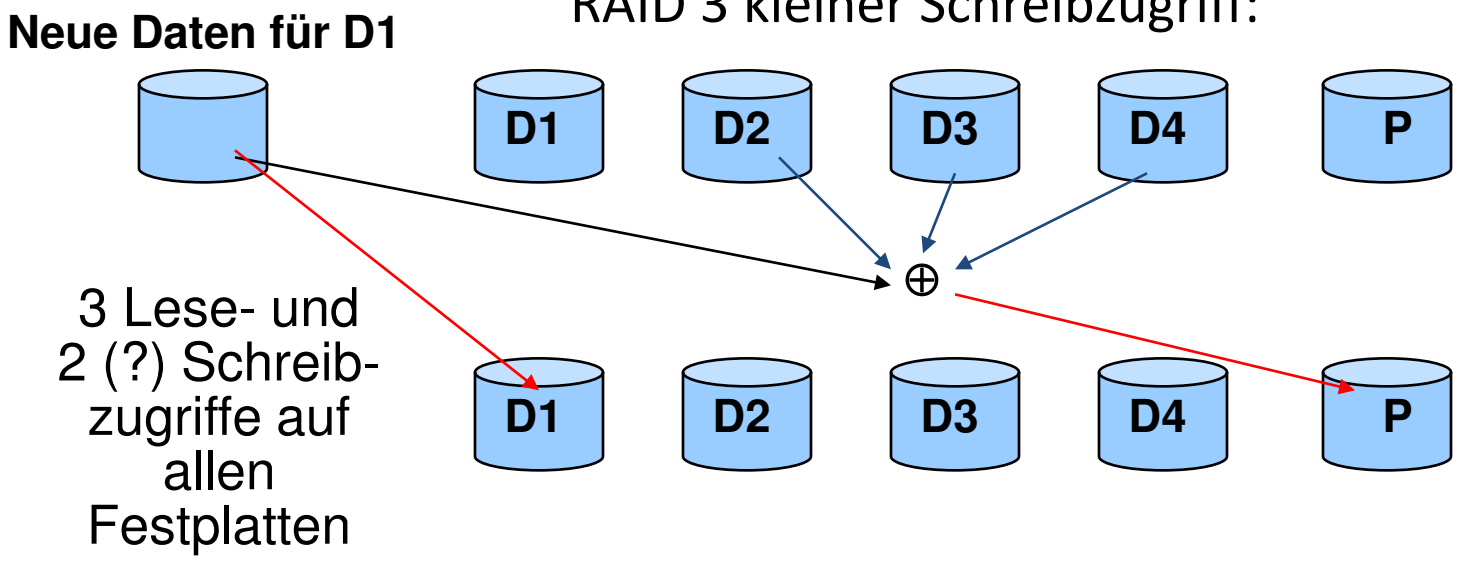
  - Block lesen: Lese von allen Festplatten
  - Block schreiben: Lese von allen Festplatten um neue Parität zu erstellen und aktualisiere alle Festplatten
  - Bei Ausfall: rekonstruiere fehlende Daten anhand der Parität



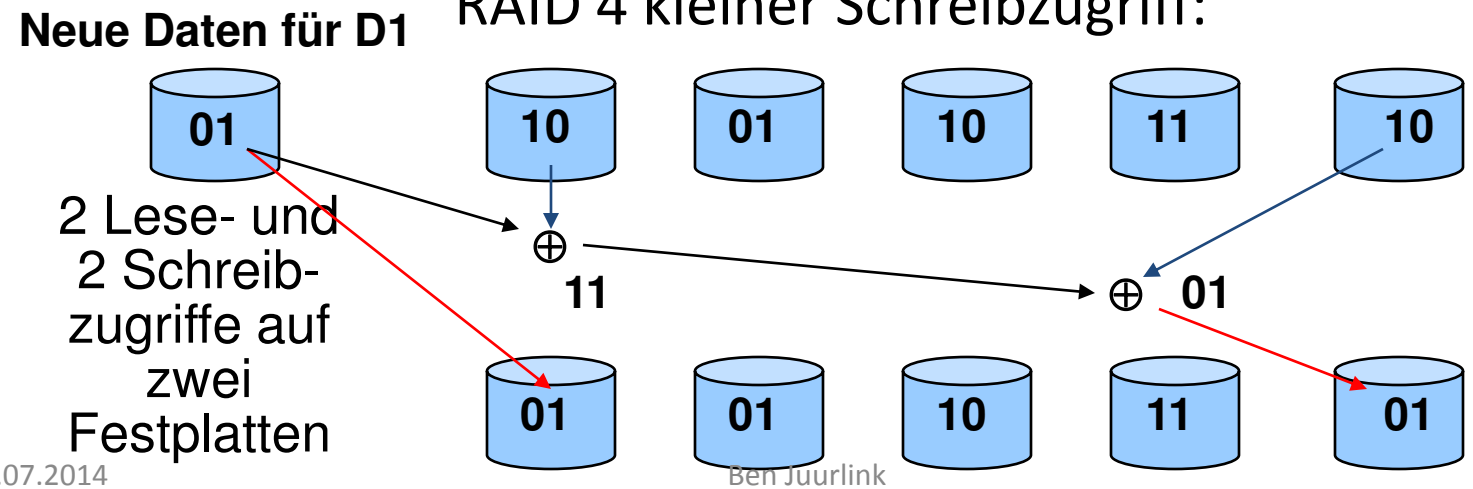
- Blockweise verschränkte Parität anstatt Bitweise verschränkt
  - Redundante Festplatte speichert Parität für eine Gruppe von Blöcken
  - Block lesen: Lese von der Festplatte des benötigten Blocks
  - Block schreiben:
    - Lese von Festplatte mit modifiziertem Block und Paritäten-Festplatte
    - Berechne neue Parität, aktualisiere Daten- und Paritäten-Festplatte
  - Bei Ausfall: benutze Parität zur Rekonstruktion der Daten
  - Schneller als RAID 3 bei kleinen Schreibzugriffen (schreibe einzelnen Block)



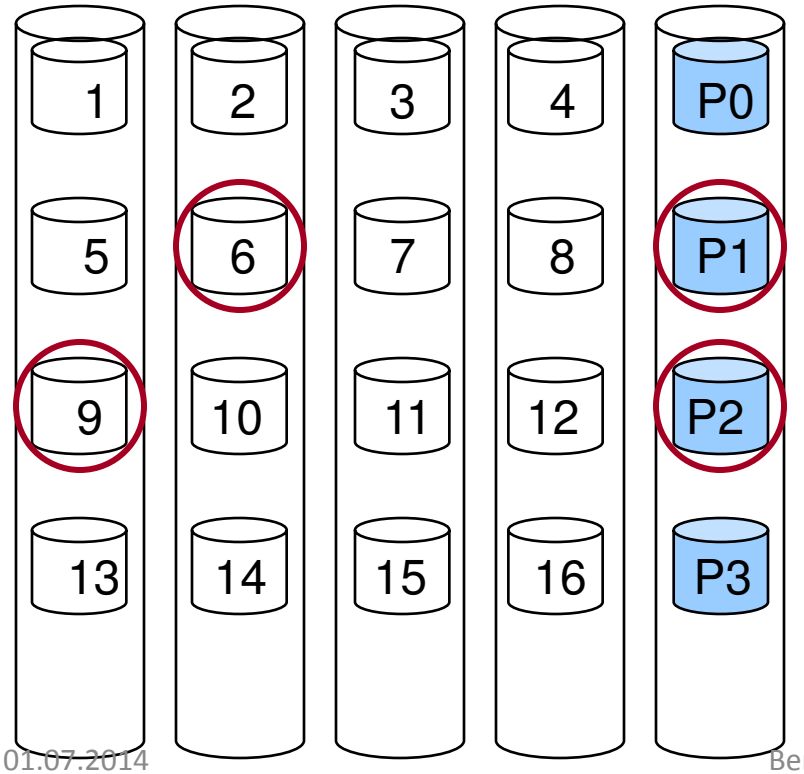
RAID 3 kleiner Schreibzugriff:



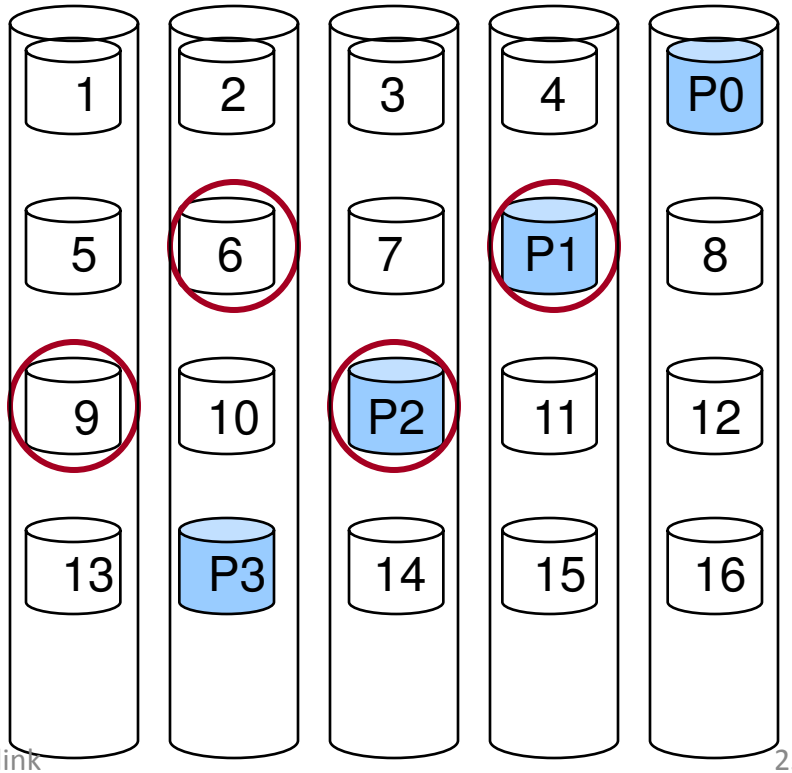
RAID 4 kleiner Schreibzugriff:



- Verteilte blockweise verschränkte Parität
  - Wie RAID 4, aber Paritätenblöcke sind über Festplatten verteilt
    - Verhindert, dass Paritätenfestplatte Engpass wird
    - Einige kleine Schreibzugriffe können parallel ausgeführt werden



RAID 4



RAID 5

Fazit Festplatten

- Vier Komponente der Festplattenzugriffszeit:
  - Suchzeit: angegeben mit 3 - 14 ms aber in Wirklichkeit schneller
  - Umdrehungslatenz: 5.6 ms bei 5400 U/min und 2.0 ms bei 15000 U/min
  - Datenrate: 30 - 80 MB/s
  - Controllerzeit: typischerweise weniger als 0.2 ms
- RAIDS zur Leistungs- und Verfügbarkeitssteigerung
  - RAID 0: keine Redundanz, nur Verteilung
  - RAID 1: Spiegeln
  - RAID 2: Fehler erkennender und korrigierender Code
  - RAID 3: Bitweise verschränkte Parität
  - RAID 4: Blockweise verschränkte Parität
  - RAID 5: Verteilte Blockweise verschränkte Parität ← oft genutzt

Komponenten verbinden

- Verbindungen nötig zwischen CPU, Speicher, Ein-/Ausgabegeräte
- Bus: gemeinsam genutzter Kommunikationskanal
  - Datenübertragung zwischen mehreren Teilnehmern
  - über gemeinsamen Kanal
- Vielseitig: Systeme, die denselben Bus verwenden können Peripheriegeräte austauschen
- Kosteneffektiv: einziger Leitungssatz gemeinsam genutzt
- Kann Engpass werden, Leistung durch physikalische Faktoren begrenzt
  - Leitungslänge, Anzahl der Verbindungen
- Alternative: Punkt-zu-Punkt Verbindungen mit Switches
  - Netzwerke

Bus Typen

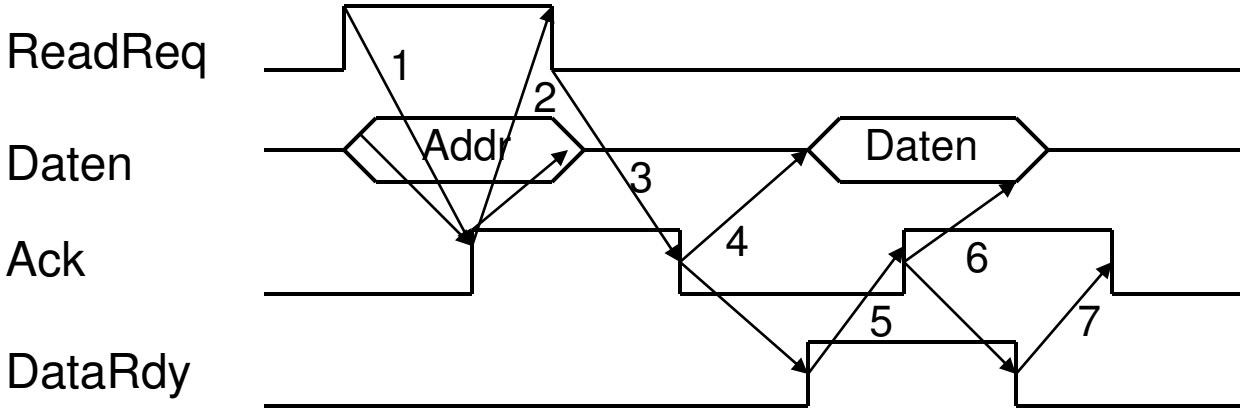
- Prozessor-Speicher Busse
  - kurz, hohe Geschwindigkeit
  - an das Speichersystem angepasst
- Ein-/Ausgabebussen
  - Länger, erlauben mehrere angeschlossene Geräte
  - Nach Standards zur Kompatibilitätssicherung entworfen
  - Mit Prozessor-Speicher-Bus über eine Brücke (*Bridge*) verbunden

Synchrone und asynchrone Busse

- Synchroner Bus
  - getaktet
  - Kommunikationsprotokoll einfach und schnell. Z. B. Lesebefehl und Adresse im 1. Taktzyklus, Speicher antwortet im 5.
  - jedes Gerät muss mit derselben Taktrate arbeiten
  - müssen kurz sein, Probleme mit Taktabweichung (*clock skew*)
- Asynchroner Bus
  - nicht getaktet
  - unterstutz große Vielfalt an Geräten, kann länger sein
  - braucht Handshake-Protokoll: Folge von Schritten um Übertragungen zu koordinieren

Bus Leitungen

- Datenleitung
  - Befördert Adressen und Daten
- Steuerleitungen:
  - *ReadReq* (*Leseanforderung*): signalisiert Anforderung zum Lesen **vom** Speicher
  - *DataRdy* (*Daten bereit*): zeigt an, dass Daten/Adresse auf Datenleitung verfügbar sind
    - Daten werden gleichzeitig auf die Datenleitung gelegt
  - *Ack* (*Bestätigung*): bestätigt Leseanforderung oder Daten-bereit-Meldung der anderen Seite



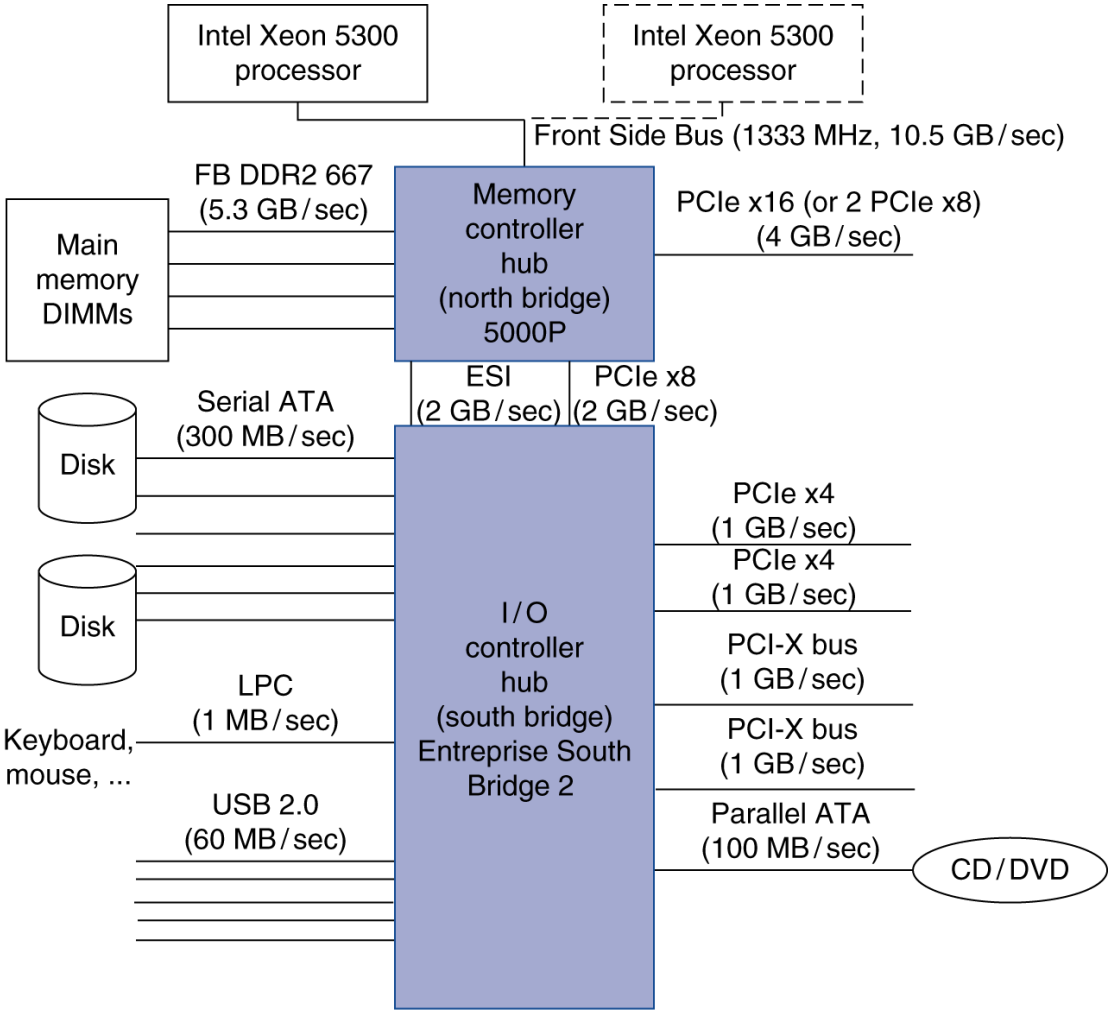
Ein-/Ausgabegerät signalisiert Anforderung durch erhöhen von ReadReq und Adresse auf Datenleitung zu legen

1. Speicher sieht ReadReq=high, liest Addr von Datenleitung und setzt Ack=high
2. EA-Gerät sieht Ack=high und gibt ReadReq und Datenleitung frei
3. Speicher sieht ReadReq=low und setzt Ack=low
4. Speicher platziert Daten auf Datenleitung und setzt DataRdy=high
5. EA-Gerät sieht DataRdy=high, liest Daten und setzt Ack=high
6. Speicher sieht Ack=high, gibt Datenleitungen frei und setzt DataRdy=low
7. EA-Gerät sieht DataRdy=low und setzt Ack=low

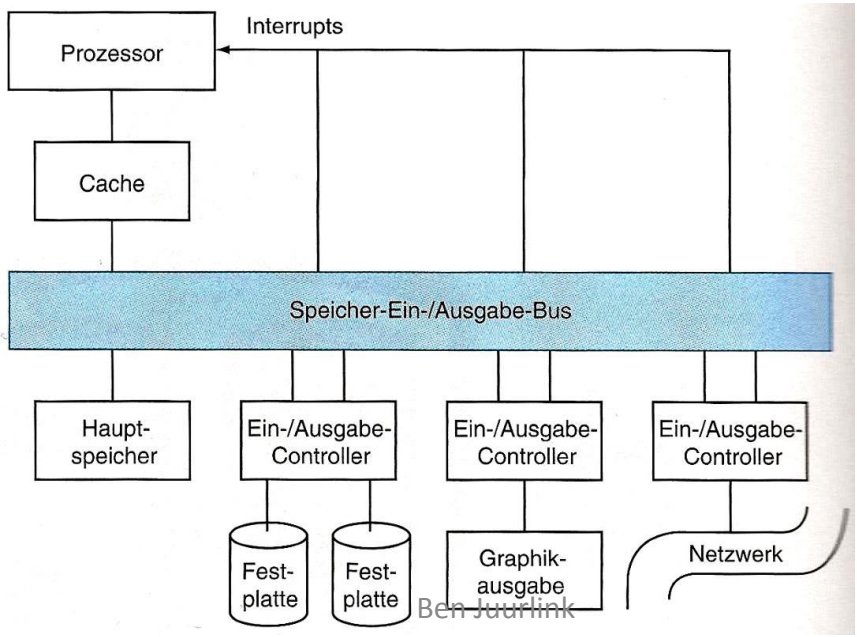
Beispiele für Ein-/Ausgabebusse

	Firewire	USB 2.0	PCI Express	Serial ATA	Serial Attached SCSI
Anwendung	Extern	Extern	Intern	Intern	Extern
Max. Geräteanzahl	63	127	1	1	4
Datenbusbreite	4	2	2/lane	4	4
Max. Bandbreite	50MB/s oder 100MB/s	0.2MB/s, 1.5MB/s, oder 60MB/s	250MB/s/lane 1×, 2×, 4×, 8×, 16×, 32×	300MB/s	300MB/s
Während Betrieb Anschließbar?	Ja	Ja	Abh. vom Gerät	Ja	Ja
Max. Buslänge	4.5m	5m	0.5m	1m	8m
Standardname	IEEE 1394	USB Implementers Forum	PCI-SIG	SATA-IO	INCITS TC T10

Typisches Ein-/Ausgabesystem eines x86 PCs



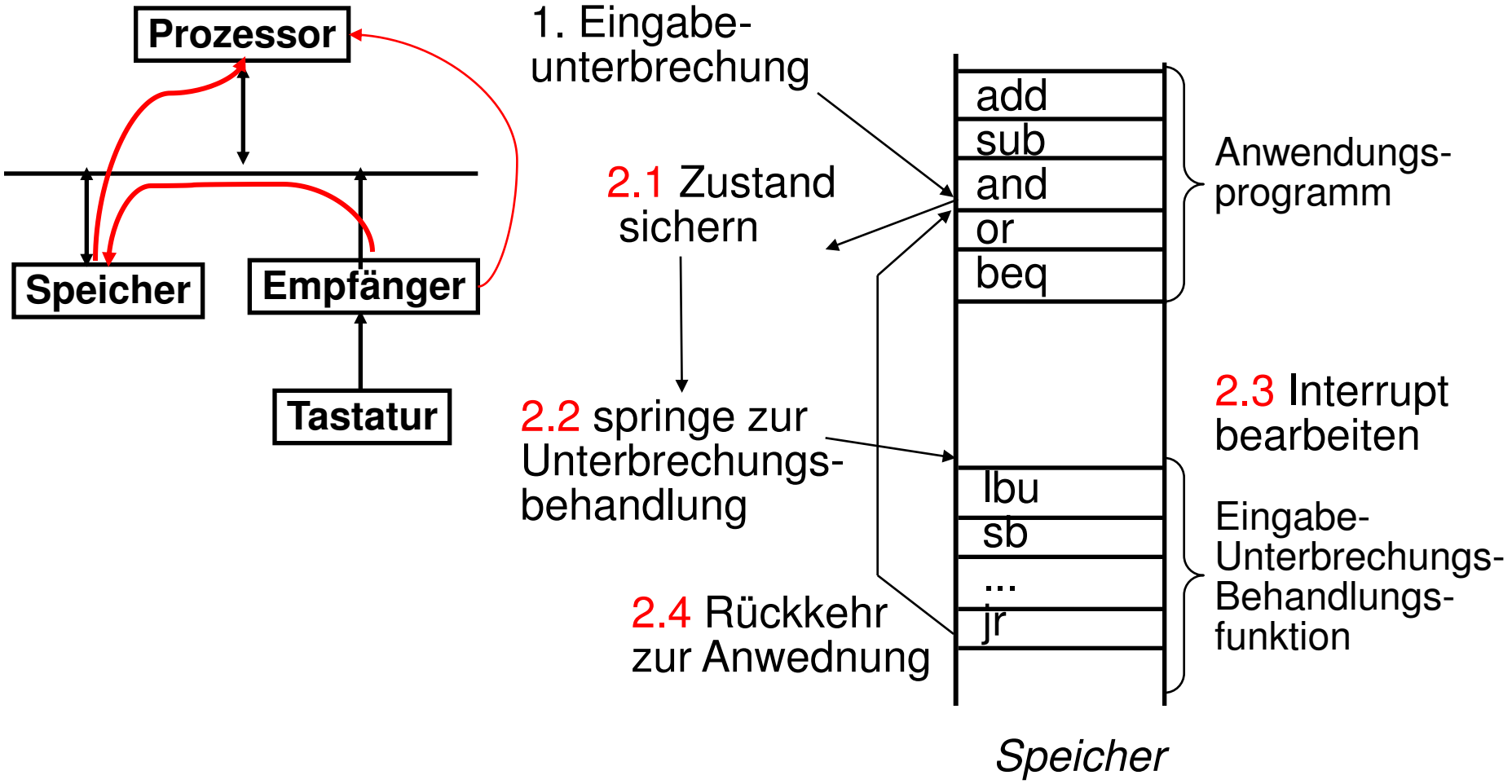
- Ein-/Ausgabe vom Betriebssystem koordiniert
  - Versch. Programmen teilen sich Ein-/Ausgaberessourcen
    - Benötigt Absicherung und Planung
  - Ein-/Ausgabe verursacht asynchrone Interrupts
    - Interrupthandler (Unterbrechungsbehandlung)
  - Ein-/Ausgabeprogrammierung ist kniffelig
    - Betriebssystem bietet Programmen Abstraktionen an



1. Befehle an Ein-/Ausgabegeräte übermitteln:
   - Speicherabgebildete Ein-/Ausgabe (*memory-mapped I/O*)
     - Teile des Adressraums werden bestimmten Ein-/Ausgabegeräten zugeordnet
     - Lese- und Schreibbefehle werden dort als Befehle für das Ein-/Ausgabegerät interpretiert
     - Nur Betriebssystem kann im Adressraum des Ein-/Ausgabegeräts lesen und schreiben
   - Spezielle Ein-/Ausgabe-Befehle
     - Separate Befehle zum Zugriff auf Register des Gerätes
     - Können nur im Kernel Mode ausgeführt werden
     - Beispiel: x86

2. Kommunikation mit dem Prozessor:

- Polling: regelmäßiges prüfen der Ein-/Ausgabe Status-Register
  - Wenn Gerät bereit, Operation ausführen. Bei einem Fehler Fehlerbehandlung starten
  - Häufig in kleinen Eingebetteten Systemen mit geringer Leistung(vorhersehbares Timing, geringe Hardwarekosten)
  - Aber: Verschwendet CPU-Zyklen
- Interrupt-gesteuerte Ein-/Ausgabe (*Interrupt-driven I/O*): Ein-/Ausgabe-Controller unterbricht Prozessor, wenn es Aufmerksamkeit benötigt
  - Prozessor stoppt aktuelle Prozesse und startet Interrupt Handler
  - Muss das unterbrechende Gerät identifizieren können
  - Es kann verschiedene Interrupt-Prioritäten geben
  - Prozessor benötigt kein Polling für Ein-/Ausgabeereignisse


Prozessor
Speicher
Empfänger
Tastatur
1. Eingabe-unterbrechung
2.1 Zustand sichern
2.2 springe zur Unterbrechungs-behandlung
2.4 Rückkehr zur Anwednung
add
sub
and
or
beq
lbu
sb
...
jr
Anwendungs-programm
2.3 Interrupt bearbeiten
Eingabe-Unterbrechungs-Behandlungs-funktion
Speicher

- Polling und Interrupt-gesteuerte Ein-/Ausgabe
  - CPU überträgt Daten zwischen Speicher und Datenregistern des Ein-/Ausgabegerätes
  - Für Geräte mit geringer Leistung akzeptabel (z.B. Tastatur) aber zu zeitaufwändig für Geräte mit hoher Bandbreite (z.B. Festplatten)
- Direkter Speicherzugriff (Direct memory access, DMA)
  - Prozessor stellt Startadresse der Daten im Speicher bereit
  - Ein-/Ausgabe-DMA-Controller überträgt Daten selbstständig von und zum Speicher
  - Controller sendet Interrupt bei Fertigstellung oder einem Fehler
  - Aber: CPU und DMA konkurrieren um Bus-/Speicher-Bandbreite

Zusammenspiel von Cache und DMA

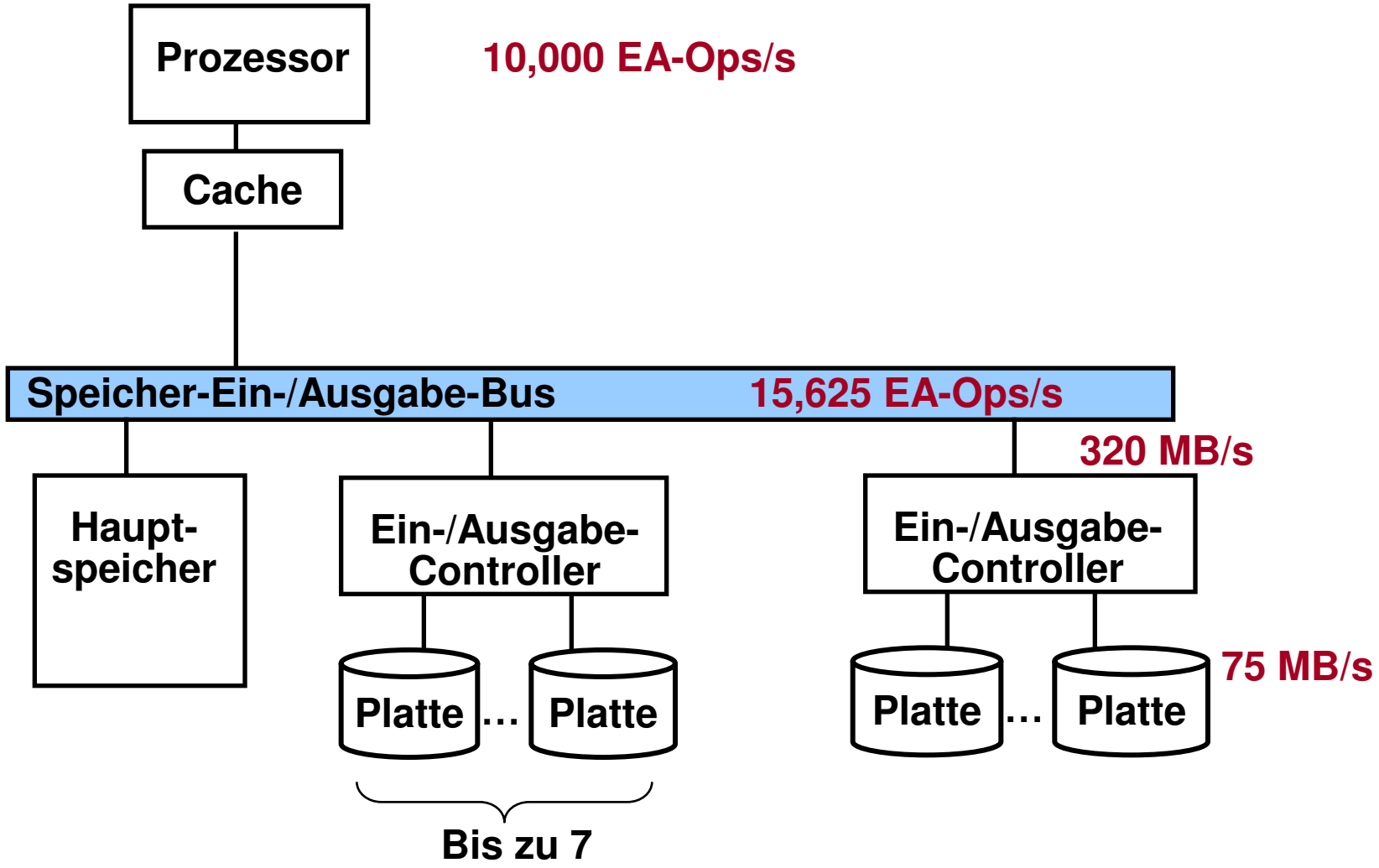
- DMA schreibt Datenblock in den Speicher, der auch im Cache ist
  - Kopie im Cache ist veraltet (stale)
- DMA liest Block aus Speicher, während neuerer Wert im Rückschreiben-Cache:
  - DMA liest veraltete Daten
- Kohärenz der Daten muss gesichert werden
  - Blöcke aus Cache werfen, wenn sie vom DMA benutzt werden
    - Cache snooping: auf dem Speicherbus lauschen, ob gecachte Daten gelesen oder geschrieben werden
      - Lesen: liefere Block aus dem Cache
      - Schreiben: Cache Block ungültig machen
    - Auch in Mehrkernprozessoren benutzt
  - Benutze Speicherbereiche für Ein-/Ausgabe, die nicht gecached werden

1. Finde die Schwachstelle im Ein-/Ausgabe-System – Komponente, die den Entwurf beschränkt
   - Prozessor- und Speicher-System?
   - Verbindungeinheit (z.B. Bus)?
   - Ein-/Ausgabe-Controller?
   - Ein-/Ausgabegeräte selbst?
2. Gestalte die Schwachstelle um, sodass Bandbreite- und Latenzanforderungen erfüllt werden
3. Bestimme Anforderungen für die anderen Komponenten und gestalte sie um, um Bandbreite- und Latenzanforderungen zu unterstützen

- Belastung: 64KB Festplattenleseoperationen
  - Jede EA-Operation benötigt 200 000 Anwendungs- und 100 000 Betriebssystembefehle
- Prozessor: 3 x 109 Befehle/s
- Speicher-Ein-/Ausgabe-Bus: 1000 MB/s
- SCSI-Controller mit:
  - DMA-Übertragungsrate von 320 MB/s
  - Bis zu 7 Festplatten pro Controller
- Festplatten:
  - Lese-/Schreib-Bandbreite: 75 MB/s
  - Durchschnittl. Suchzeit + Umdrehungslatenz: 6 ms

- Was ist maximal erreichbare Ein-/Ausgabe-Rate (EA-Ops/s) und welche Anzahl von Festplatten und SCSI-Controllern ist nötig um diese zu erreichen?

- Belastung: 64KB Festplattenleseoperationen
  - Jede EA-Op benötigt 200 000 Anwendungs- und 100 000 Betriebssystembefehle
- Prozessor: 3 x 10^9 Befehle/s
  - EA-Ops/s: $$\frac{3\times10^9}{(200+100)\times10^3}=10{,}000$$
- Speicher-Ein-/Ausgabe-Bus: 1000 MB/s
  - EA-Ops/s : $$\frac{1000\times10^6}{64\times10^3}=15{,}625$$
- Also Prozessor und nicht Bus ist Engpass.


Prozessor
10,000 EA-Ops/s
Cache
Speicher-Ein-/Ausgabe-Bus
15,625 EA-Ops/s
320 MB/s
Haupt-speicher
Ein-/Ausgabe-Controller
Ein-/Ausgabe-Controller
Platte
Platte
Platte
Platte
75 MB/s
Bis zu 7

- Festplatten:
  - Lese-/ Schreib-Bandbreite: 75 MB/s
  - Durchschnittl. Suchzeit + Umdrehungslatenz: 6 ms
- ➢ Zeit pro EA-Op auf Festplatte = Suchzeit + Umdrehungszeit + Transferzeit = 6ms + 64KB/(75MB/s) = 6.9ms
- ➢ Jede Festplatte kann 1 EA-Op/6.9ms durchführen = 146 EA-Ops/s
- ➢ Um CPU auszulasten sind 10 000 EA-Ops/s nötig → 10,000/146 = 69 Platten

- SCSI-Controller:
  - Bis zu 7 Festplatten pro Controller
  - DMA-Übertragungsrate von 320 MB/s
- ➢ Lasten wir den SCSI-Bus mit 7 Festplatten pro Controller aus?
- ➢ Nein, da Festplattenübertragungsrate = Datengröße/Übertragungszeit = 64KB/6.9ms = 9.56 MB/s
- ➢ Lasten wir den Speicher-Ein-/Ausg.-Bus mit 69/7 = 10 SCSI-Busse und -Controllern aus?
- ➢ Nein, da 69 x 9.56 MB/s = 660 MB/s < 1000 MB/s

Fazit

- Maße für Ein-/Ausgabe-Leistung
  - Durchsatz (Bandbreite), Reaktionszeit (Latenz)
  - Verlässlichkeit und Kosten
- RAID: verbessert Leistung und Verlässlichkeit
- Busse werden benutzt um CPU, Speicher und Ein-/Ausgabe-Geräte zu verbinden
  - Polling, Interrupts, DMA
- Vergiss Amdahl nicht:
  - Vernachlässige nicht die Ein-/Ausgabeleistung da Parallelisierung die Rechenleistung erhöht.
  - Beispiel:
    - Benchmark benötigt 90s CPUZeit, 10s Ein-/Ausgabe-Zeit
    - Doppelte CPU-Anzahl/2 Jahre, Ein-/Ausgabe unverändert
    - ➢ Nach 6 Jahren: 11s CPU-Zeit, 10s Ein-/Ausgabe-Zeit (47%)