

Vorlesung: Rechnerstrukturen, Teil 2 (Modul IP7)

J. Zhang

zhang@informatik.uni-hamburg.de

Universität Hamburg

Fachbereich Informatik

AB Technische Aspekte Multimodaler Systeme

Inhaltsverzeichnis

8. Die Speicherhierarchie	341
RAID341
Optisches Speichermedium CD-ROM350
9. I/O: Ein- und Ausgabe	360
Busse361
Unterbrechungen369
DMA ("Direct Memory Access")386

RAID: Motivation

Amdahl's Gesetz:

langsamste Komponente behindert Leistungssteigerungen

- ausgewogenes Verhältnis CPU–Speicher–I/O nötig
- CPU und Speicher skalieren mit der Halbleitertechnologie
- aber wie kann die I/O-Leistung gesteigert werden?

RAID, “redundant array of inexpensive disks”:

- Grundidee: viele kleine PC-Festplatten statt einer großen
- bedingt in damaliger (1985) Festplattentechnologie:
Großrechner-Festplatten vs. PC-Festplatten
- Zuverlässigkeit durch redundante Platten
- Wiederherstellung der Daten nach Plattenausfall
- ursprünglich: “independent disks”

Disks: RAID

“redundant array of inexpensive disks”

- bahnbrechende Untersuchung von Festplatten-Performance
- ursprüngliche Analyse von Großrechner- und PC-Festplatten
- Ersetzen wenige große durch viele kleine Festplatten
- Zuverlässigkeit des Gesamtsystems?

- diverse RAID-Varianten (Level)
- unterschiedliche Anzahl von Platten
- Strategien zur Verwendung von Nutz- und Reserveplatten
- Ausfallsicherheit, Hot-Plugging
- Optimierung auf Schreib- und/oder Leseperformance
- vielfache Anwendungen

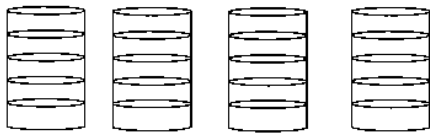
- möglichst das Original lesen!

(Patterson, Gibson, Katz: UCB report CS-98-391)

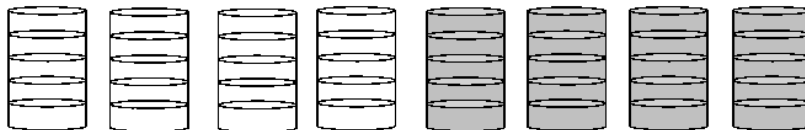
RAID: 7 Typen

Man unterscheidet mehrere Ebenen, die sieben bekanntesten sind:

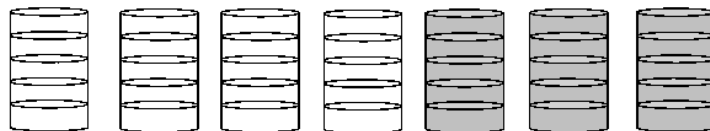
RAID 0: Keine Redundanz



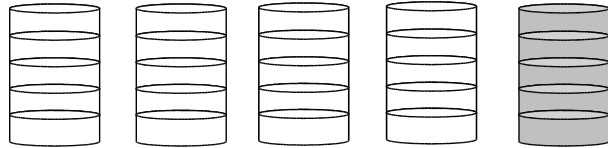
RAID 1: Spiegelung von Einzelplatten



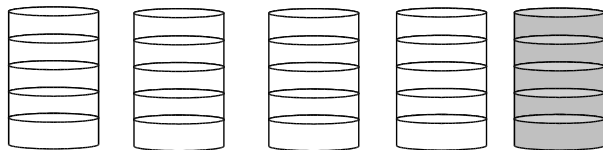
RAID 2: Einsatz fehlerkorrigierender Bitcodes



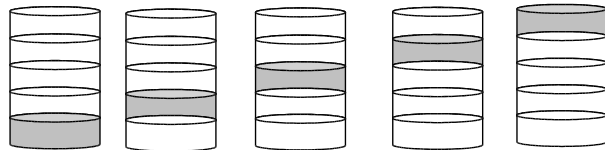
RAID 3: Bit-Parität



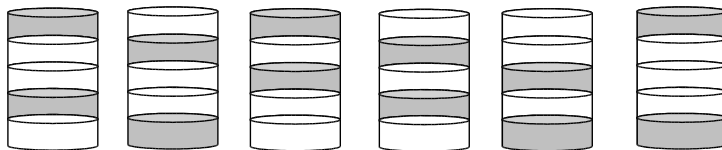
RAID 4: Block-Parität



RAID 5: Rotierende Block-Parität



RAID 6: Doppelte Redundanz



RAID 0

Die Datenblöcke werden gleichmäßig auf mehrere unabhängige Platten verteilt.

Hierdurch erreicht man Lastglättung.

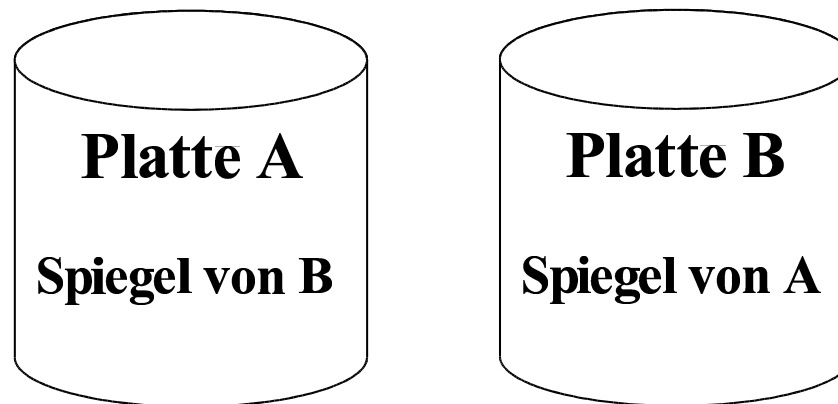
Block 0	Block 1	Block 2	Block 3
Block 4	Block 5	Block 6	Block 7
Block 8	Block 9	Block 10	Block 11
Block 12	Block 13	Block 14	Block 15

RAID 1

Jeder Platteninhalt ist zweimal vorhanden.

Bei Ausfall einer Platte nutzt man das Duplikat und erstellt eine neue Kopie auf einer Reserve-platte.

Hierbei muß jeder Schreibzugriff doppelt ausgeführt werden. Lesezugriffe werden nur einmal ausgeführt.



RAID 2

Man verwendet einen fehlerkorrigierenden Bitcode.

Ein einfacher ist der (7, 4) Hamming-Code. Die Originalbit und die Redundanzbit werden auf verschiedene Platten verteilt. Ein Datenzugriff erfordert viele synchronisierte Plattenoperationen.

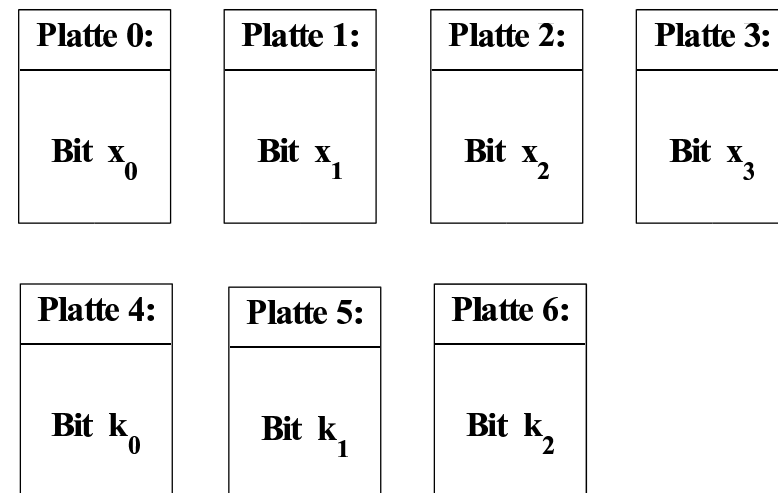
Im Fall des (7, 4)-Codes gehören zu einem Original-Quadrupel drei Korrekturbit, man benötigt also sieben Platten. Insgesamt erscheint der Aufwand nicht gerechtfertigt.

Gleichungssystem:

$$k_2 = x_2 + x_1 + x_0 \pmod{2}$$

$$k_1 = x_3 + x_1 + x_0 \pmod{2}$$

$$k_0 = x_3 + x_2 + x_0 \pmod{2}$$

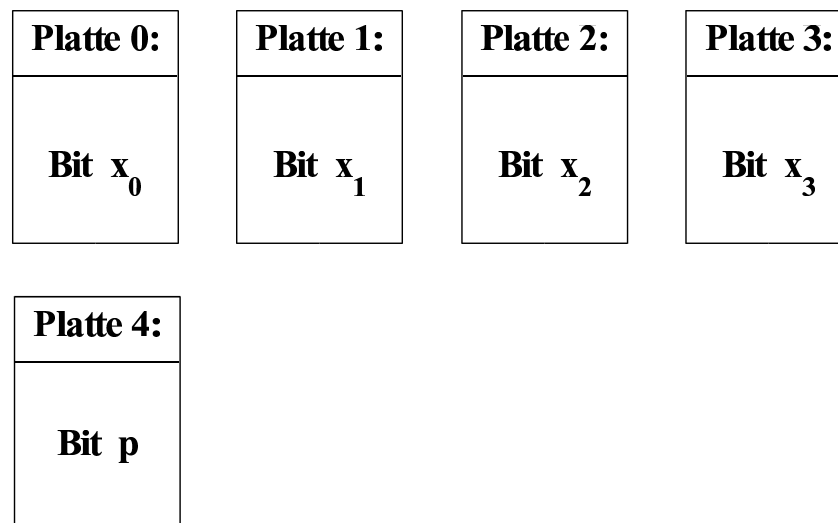


RAID 3

Um gegen den Totalausfall einer Platte gewappnet zu sein, genügt der Einsatz einer Paritätsplatte. Hat man die Originaldaten auf vier Platten verteilt, dann bildet man die Modulo-2-Summe über entsprechende Bits der Originaldaten.

Formel: $p = x_0 \text{ XOR } x_1 \text{ XOR } x_2 \text{ XOR } x_3$

Bei Ausfall einer der Platten 0 bis 4 garantiert die Eigenschaft der Modulo-2-Addition die Restaurierbarkeit des Inhalts der ausgefallenen Platte.



RAID 4, 5, 6

RAID 4: Dies ist ähnlich zu RAID 3. Die Paritätsinformation nutzt man nur bei Totalausfall einer Platte. Daher führen nur Schreibzugriffe zu einer erhöhten Last. Jeder Schreibzugriff erfordert zwei Lese- und zwei Schreibzugriffe.

RAID 5: Um den Flaschenhals der Paritätsplatte zu mildern, verteilt man die Redundanzinformation zyklisch über alle Platten.

RAID 6: Hier erhöht man die Datensicherheit gegenüber RAID 5 durch Bildung mehrerer unabhängiger Schutzinformationen.

Optisches Speichermedium CD-ROM

(CD-ROM = Compact Disc Read Only Memory)

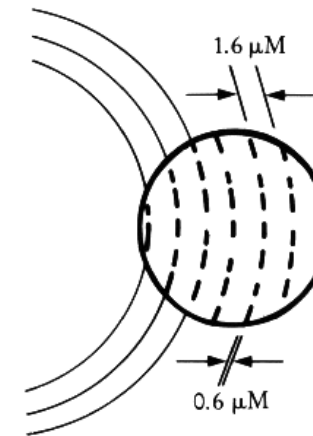
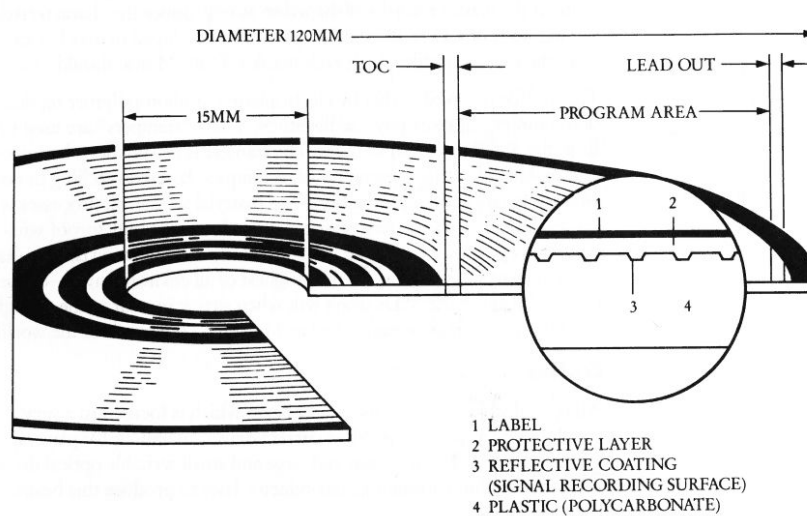
Die CD-ROM ist eine Weiterentwicklung der Audio-CD, die 1982 auf den Markt kam. Die Daten werden in einer Spirale von über 5 km Länge aufgezeichnet. Die Speicher-kapazität einer 74-Minuten-CD-DA beträgt etwa 783 Millionen Nutzbyte. Für die CD-ROM im engeren Sinne sinkt dieser Wert auf 650 MB. Die Standards für Audio-CD und CD-ROM sind IEC 908 ("red book") und ISO/IEC 10149 ("yellow book").

Aufzeichnungsverfahren für die Rohdaten

Jeweils 24 8-Bit-Datenbyte werden 4 Q-Paritäts-Byte und 4 P-Paritäts-Byte und ein Subcode-Byte hinzugefügt. Jedes Byte wird durch 17 Kanalbit codiert. Die teils technisch bedingte aufwendige Codierung, wobei noch eine geschickte Verschachtelung der Datenbyte hinzukommt, führt zu einer Wahrscheinlichkeit von 10^{-8} für nicht korrigierbare Lesefehler auf der Audio-Ebene.

Für Daten ist dies zu gering. Zu jeweils 2.048 Datenbyte berechnet man zusätzlich 276 Byte Fehlerkorrekturcode. Damit sinkt die Wahrscheinlichkeit eines nicht korrigierbaren Lesefehlers von 10^{-8} auf der Audio-Ebene auf 10^{-12} auf der Datenebene.

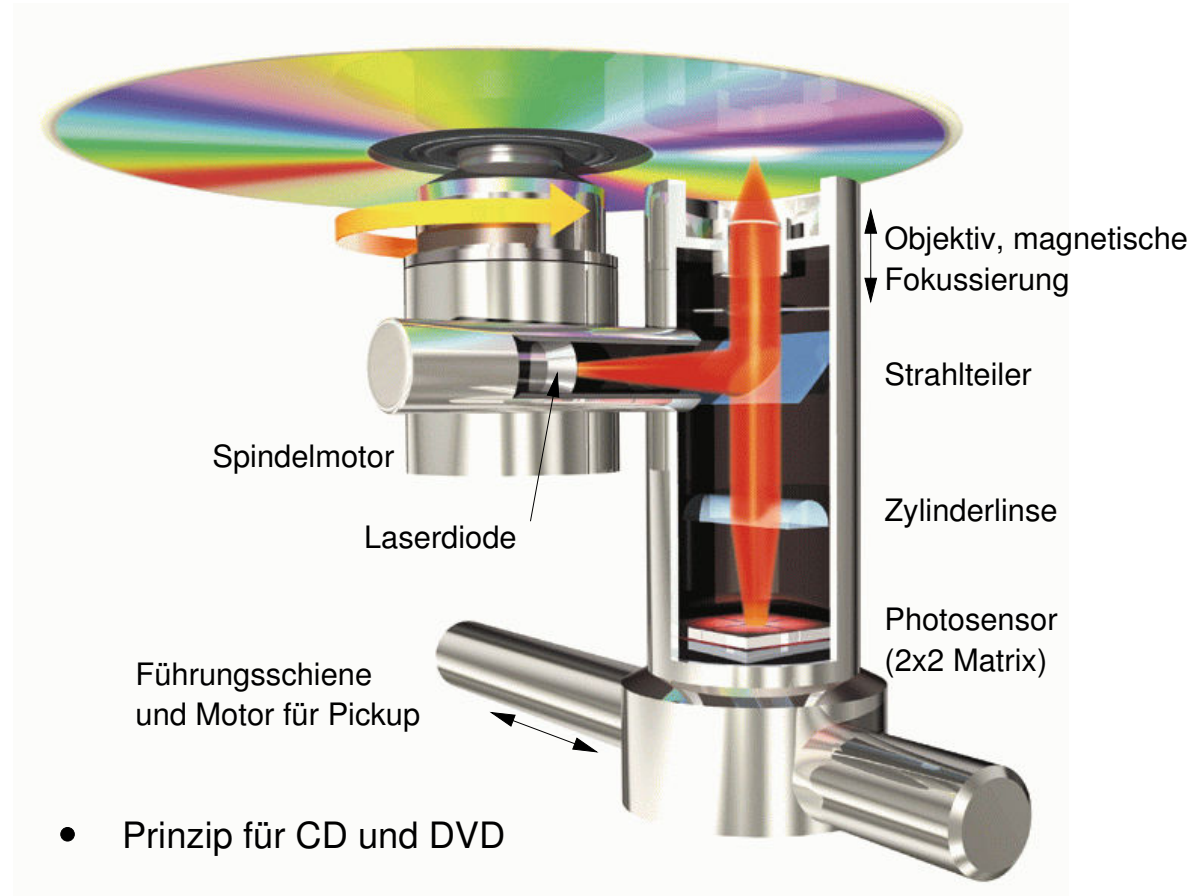
CD: Prinzip



- Polycarbonatträger, 12cm Durchmesser
- eingeprägte Vertiefungen ("pits") bilden die Daten
- spiralförmige Datenspur, 1.6μm Abstand, ca. 16000 Windungen
- Fertigungsmängel fest eingeplant => leistungsfähige Fehlerkorrektur

(CD-ROM – the new papyrus)

CD: Aufbau eines Players



Einige Merkmale der Audio-CD:

maximale Spieldauer:	74 Minuten, 33 Sekunden
konstante Lesegeschwindigkeit:	1,2 m/s - 1,4 m/s
Spurabstand:	1,6 μm
Durchmesser:	120 mm
Dicke:	1,2 mm
Zentrumsdurchmesser:	15 mm
Aufzeichnungsbereich:	46 mm - 117 mm
Datenbereich:	50 mm - 116 mm
Minimale PitLänge:	0,833 μm (1,2 m/s) bis 0,972 μm (1,4 m/s)
Maximale PitLänge:	3,05 μm (1,2 m/s) bis 3,56 μm (1,4 m/s)
PitTiefe:	0,11 μm
PitBreite:	0,5 μm

Einige Merkmale der Audio-CD (Forts.):

Standard Wellenlänge:	780 nm
Brechungsindex des Materials:	1,55
Zahl der Kanäle:	2 (4 zulässig)
Quantelung:	16 Bit linear
Abtastfrequenz:	44.100 Hz
Kanal-Bit-Rate:	4,3218 Mbit/s
Daten-Bit-Rate:	2,0338 Mbit/s
Verhältnis Datenbit zu Kanalbit:	8 : 17
Fehlerkorrekturcode:	Verschachtelter Reed-Solomon Code (25% Redundanz)
Modulationssystem:	Eight to fourteen Modulation (EFM)

Ein Audio-Rahmen:

Synchronisation	24 Bit
Subcode	14 Bit
$6 * 2 * 2 * 14$ Datenbit	336 Bit
$8 * 14$ Paritätsbit	112 Bit
$34 * 3$ Pufferbit	102 Bit
<hr/>	
Summe:	588 Bit

Ein Audio-Rahmen: Bemerkungen

1. 192 Datenbit werden in 588 Kanalbit codiert.
2. Das Subcode-Byte enthält 8 Bit, die den Subkanälen P, Q, R, S, T, U, V, W zugeordnet sind. Nur die Subkanäle P und Q werden von der Audio-disc genutzt.
3. Die Fehlerbehandlung erfolgt in drei Schritten:
 - (a) Zunächst wird versucht, die fehlerhaften Byte zu korrigieren.
 - (b) Unkorrigierbare Bytewerte versucht man durch Interpolation zu gewinnen.
 - (c) Läßt sich auch die Interpolation nicht durchführen, dann wird der entsprechende Musikeil durch Stille ersetzt.

Beispiel zur Codeverschränkung:

Länge der Nachricht = 103

Auslesen der teilweise zerstörten Nachricht:

Ein Buch mit dem Titel "Das Geheimnis meiner XXXXXXXXXX" kann nichts anderes als leere Seiten enthalten.

Speicherung der gleichen Nachricht mit Verschränkungszahl 11:

E ltnsmi aeialet etSnhnlilhai ene so ann di iBXXXXXXXXXXmuleDtermeincenaer ncihe" sneMT hs r .siiet mekG

Rücktransformation:

Ein Buch mit dem Titel "Das Geheimnis meiner Millionen" kann nichts anderes als leere Seiten enthalten.

Nun ist die ursprüngliche Nachricht ohne großen Aufwand rekonstruierbar.

Bemerkung: Mittels Codeverschränkung werden Bündelfehler in Einzelfehler transformiert.

Konversion von 8-Bit-Byte zu “pits and lands”

Daten: 11101000 11100010 10111010 11101011

Umsetzung in 14-Bit-Darstellung:

10000100000010 10000100010010 10010000001001 00001001000010

Einfügen von Pufferbits:

10000100000010 000 10000100010010 000 10010000001001 001

00001001000010

Bitfolge:

10000100000010000100001000100100001001000000100100100001001000010

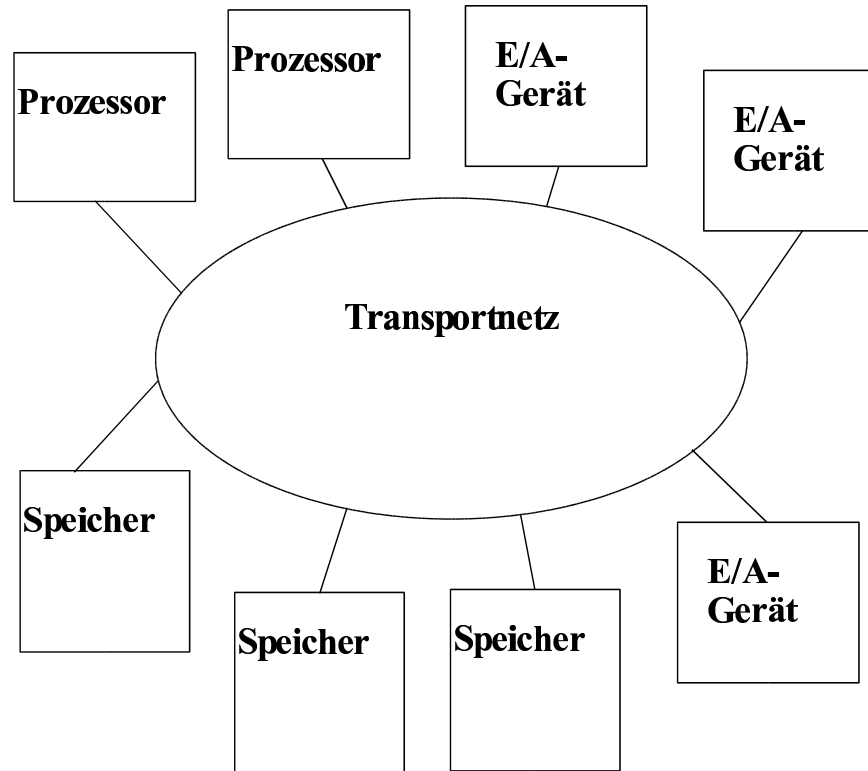
“Pits and Lands”:



Bemerkung: Der Wechsel zwischen “pit” und “land” wird durch eine 1 codiert.

I/O: Ein- und Ausgabe

Rechenanlage als Ansammlung von Einzelgeräten:

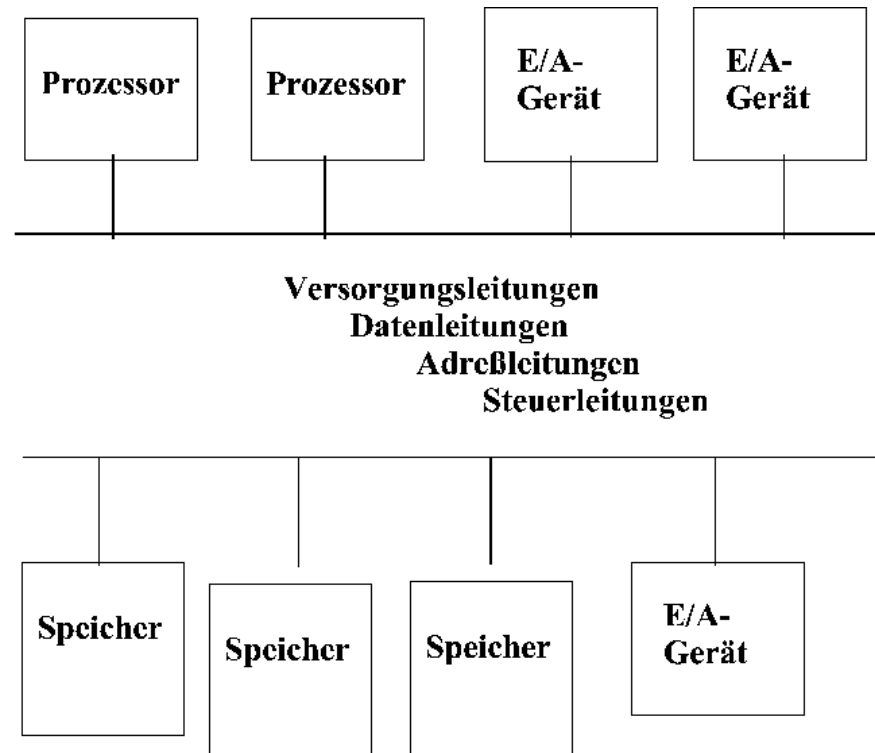


Bemerkung:

Der Datenaustausch zwischen zwei Geräten ist durch Protokolle zu regeln.

Busse

Ein einfaches Transportsystem ist der Bus.



Bemerkung:

Das Wort Bus ist eine Verkürzung des lateinischen Wortes omnibus. Es existieren viele Busstandards, zwei bekannte sind der PCI-Bus und der SCSI-Bus.

Kenngrößen eines Busses

- Verbindungsvielfalt:
 - ◆ Verbindung zwischen zwei Geräten,
 - ◆ Verbindung zwischen mehreren (≥ 3) Geräten.
- Nutzungsart:
 - ◆ Mehrfachnutzung von Leitungen,
 - ◆ dedizierte Leitungen.
- Bestimmung des temporären Busherrschers:
 - ◆ zentralisiertes Verfahren,
 - ◆ verteiltes Verfahren.
- Zeitsteuerung:
 - ◆ Taktsteuerung,
 - ◆ kein Zentraltakt (“asynchron”).

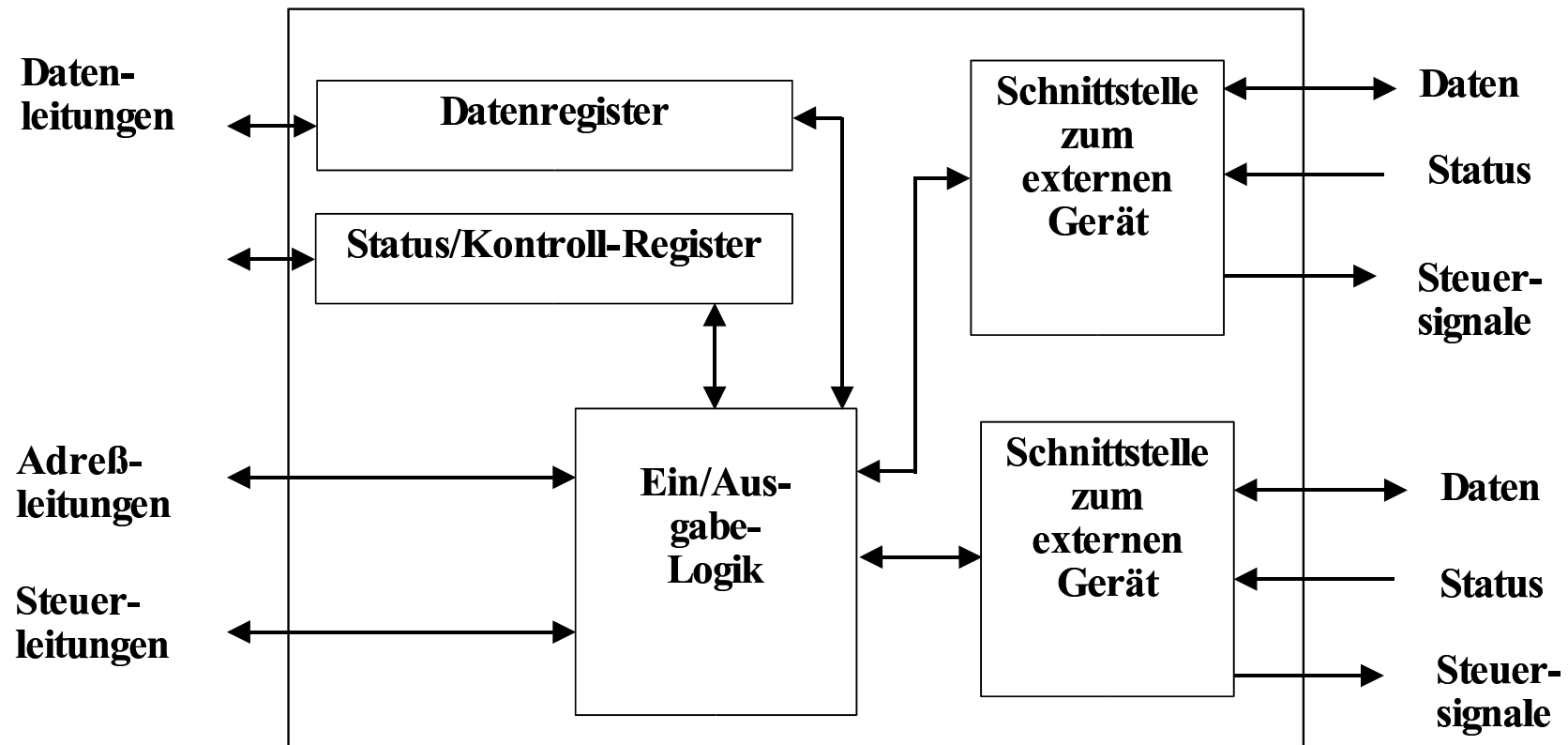
Kenngrößen eines Busses (Forts.)

- Busbreite (Zahl der Leitungen für Teilbusse):
 - ◆ Daten,
 - ◆ Adresse.
- Daten-Transfer-Art:
 - ◆ “read” ,
 - ◆ “write” ,
 - ◆ “read-modify-write” ,
 - ◆ “read-after-write” ,
 - ◆ “block-transfer” .

Bemerkung: Ein Bus, der viele Einheiten verbindet, stellt einen Systemengpass dar. Daher finden sich in einer Rechenanlage häufig mehrere Bussysteme.

Generischer E/A-Modul

Schnittstelle zum Systembus



Einige typische Datentransferraten:

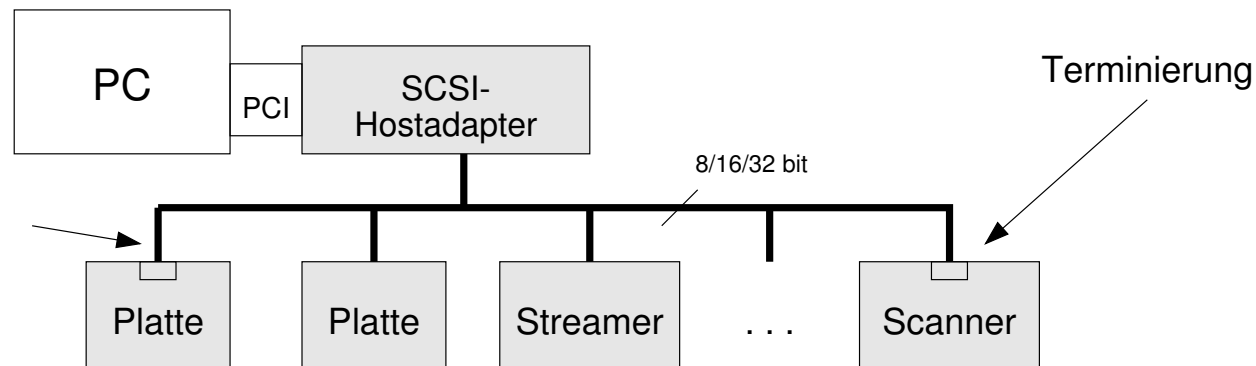
Gigabit Ethernet	10^9 Bit/s
Bildschirm	10^9 Bit/s
640 x 480, 8 Bit Farbtiefe, 70 fps:	$1,7 * 10^8$ Bit/s
1024 x 768, 24 Bit Farbtiefe, 100 fps:	$1,9 * 10^9$ Bit/s
1600 x 1200, 24 Bit Farbtiefe, 100 fps:	$4,6 * 10^9$ Bit/s
Ultra DMA 133	10^9 Bit/s
Fast Ethernet	10^8 Bit/s
Festplatte	10^8 Bit/s
Ethernet	10^7 Bit/s
Laser Drucker	10^6 Bit/s
Modem	$5 * 10^4$ Bit/s
Maus	10^2 Bit/s
Tastatur	10^2 Bit/s

SCSI: Übersicht

SCSI := Small Computer Systems Interface

- hervorgegangen aus “Shugart Associates SI”
- standardisiert als SCSI-I, SCSI-II, SCSI-III
- Einsatz in PCs (Server), Mac, Workstations
- universeller Bus für Peripheriegeräte (“Targets”)
- Z. B. Bandlaufwerke, Scanner, Musiksynthesizer, ...
- 8-bit parallel (wide-SCSI mit 16-/32-bit)
- “Hostadapter” steuert Bus
- komplexe Befehle und Arbitrierung
- flexibler, aber auch teurer und komplexer als EIDE/ATAPI

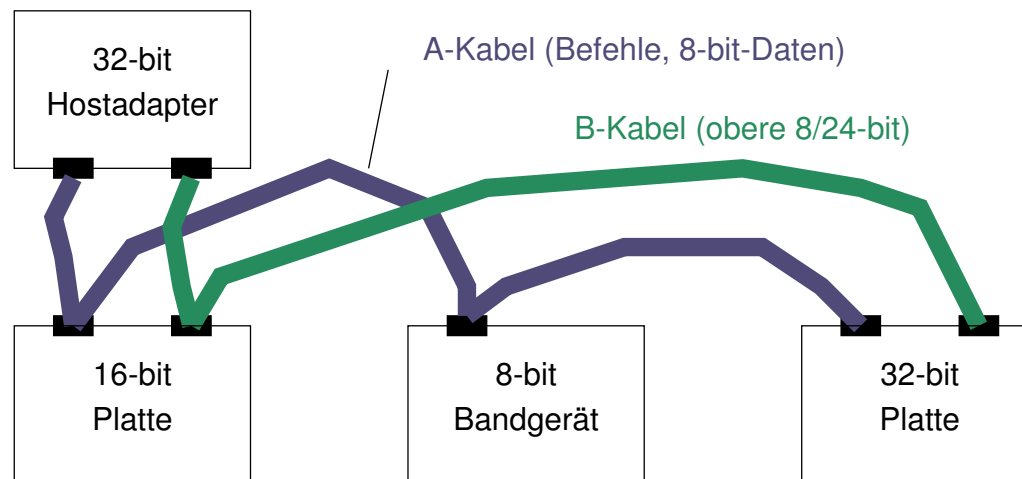
SCSI: Grundlagen



- Bus mit 8 Geräten (LUN 0..7), inklusive Controller
- Gerätenummer per Schalter eingestellt (nicht automatisch)
- komplexe Regeln zur Verkabelung (Abstände, Terminierung)
- parallele Datenübertragung, 8-bit oder (wide) 16/32-bit
- aufwendiges Busprotokoll mit Arbitrierung und split-transactions
- Geräte handeln die jeweils bestmögliche Geschwindigkeit aus
- langsame Geräte stören schnelle Geräte nicht

SCSI: Varianten

- Befehlssätze: SCSI-1, SCSI-2, SCSI-3
- Busbreite: normal 8-bit, wide-SCSI 16-bit und 32-bit
- Bustiming: SCSI-1 bis 5 MB/s, Fast 10 MB/s Ultra 20 MB/s
- alle Kombinationen, z. B. U2W = Ultra-Wide SCSI-2
- alle Gerätevarianten miteinander kombinierbar
- insbesondere auch normale und wide-SCSI Geräte



Unterbrechungen

Durch eine Unterbrechung wird der Prozessor dem laufenden Programm entzogen und einem anderen Programm zugeteilt. Auslöser für einen derartigen Programmwechsel ist im allgemeinen ein Hardware-Signal.

Bemerkungen:

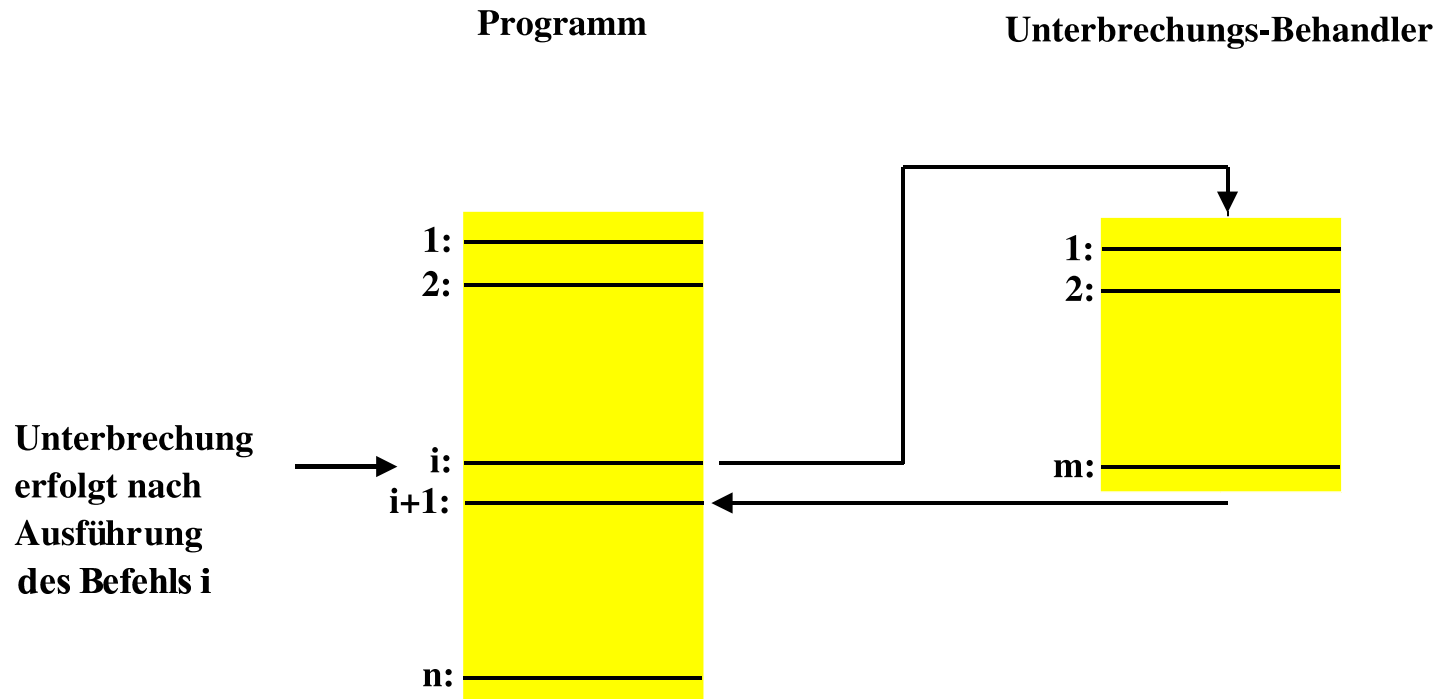
- Im allgemeinen wird eine Unterbrechung nur nach Ausführung eines Maschinenbefehls behandelt
- Außer den von der Hardware ausgelösten Unterbrechungen gibt es *synchrone Exceptions*, die von der Software selbst ausgelöst werden (*traps, faults, aborts*).

Unterbrechungen (Forts.)

Grober Ablauf einer Unterbrechung:

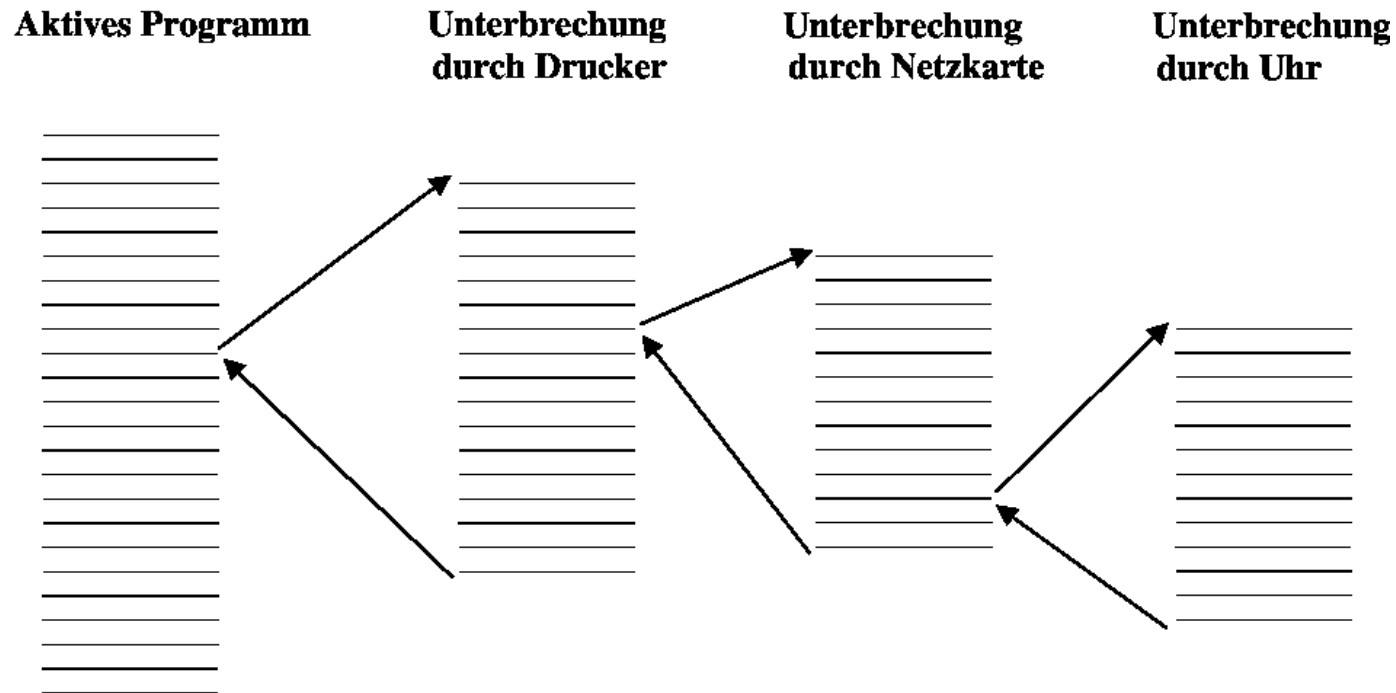
1. Ein Unterbrechungssignal S tritt auf und wird erkannt.
2. Es wird geprüft, ob das Signal S das aktive Programm AP unterbrechen darf.
3. Wenn nein, wird das Signal S zur späteren Behandlung in eine Warteschlange eingereiht.
4. Wenn ja, wird der Programmkontext von AP an vorbestimmten Orten gerettet.
5. Das dem Signal S zugeordnete Programm SP wird gestartet.
6. Hat das Programm SP seine Aufgabe erledigt, dann wird der Programmkontext von AP restauriert und das Programm AP setzt seine Arbeit fort. Im Idealfall hat eine Unterbrechung keine Auswirkungen auf das unterbrochene Programm.

Übergabe der Kontrolle mittels Unterbrechungen:



Wiederunterbrechung

Eine Unterbrechungsbehandlungsroutine kann auch wieder unterbrochen werden.



Bemerkung: Mittels Unterbrechungsmaskierung verhindert man die Unterbrechung einer Unterbrechungsbehandlungsroutine.

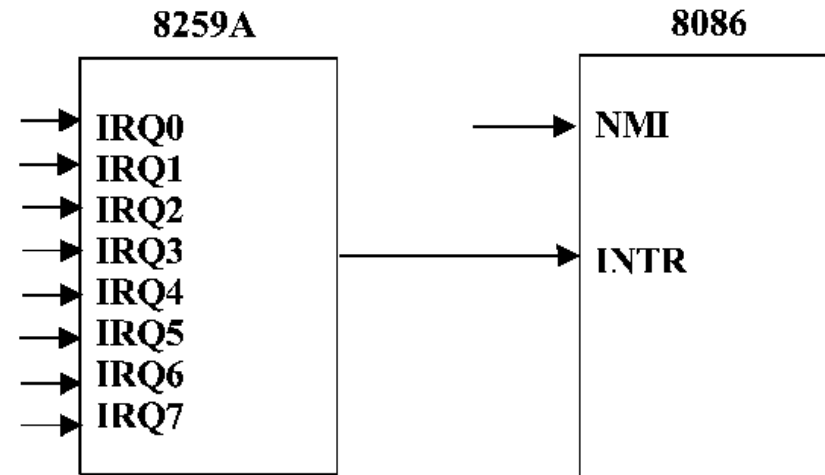
Bemerkungen zum Unterbrechungsbegriff

1. Die Unterbrechungsbehandlung wird integriert in eine allgemeine Behandlung von Ausnahmesituationen. Zu den Ausnahmen zählt man u. a. Division durch 0, Schutzverletzungen, Ausrichtungsverletzungen und Seitenalarme.
2. Damit der Kontextwechsel schnell durchgeführt werden kann, wird er weitgehend in Hardware realisiert.
3. Den einzelnen Ausnahme-Ursachen ordnet man verschiedene Prioritäten zu. Im Konfliktfall wird die Unterbrechung höherer Priorität behandelt.
4. Unterbrechungsanforderungen können durch Überschreiben verloren gehen.
5. Die Unterbrechungen nummeriert man gerne durch. Da die Startadresse einer Unterbrechungsbehandlungsroutine ohne weitere Umstände aus der Unterbrechungsnummer herleitbar sein sollte, nutzt man oft die Wörter mit niederen Hauptspeicheradressen als Unterbrechungsvektoren.

8086-Systeme

Bemerkungen:

1. 80x86 Prozessoren besitzen zwei Unterbrechungseingänge, einen normalen und einen NMI-Eingang (NMI = "Non-Maskable Interrupt").
2. Ein 8086-System besitzt acht Unterbrechungsleitungen, die von einem 8259A-Baustein verwaltet werden. Der Unterbrechungs-Controller 8259A ist über die Ports 20h und 21h programmierbar.
3. Die nicht maskierbare Unterbrechung kann nicht über den CPU-Befehl CLI unterdrückt werden, wohl aber über Port 70h. Der "NMI" dient dazu, schwerwiegende Hardware-Fehler, wie z. B. Paritätsfehler und Stromausfall, zu erkennen.

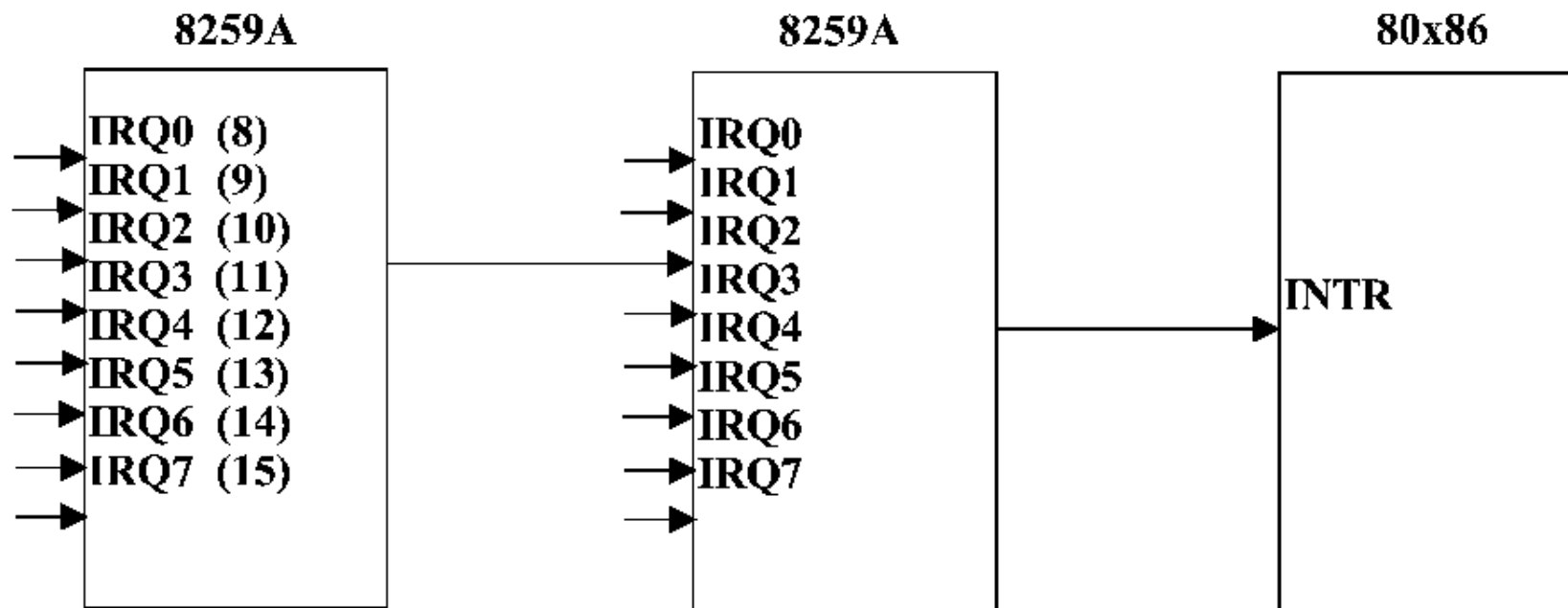


Struktur einer Unterbrechungs-Behandlungs-Routine

am Beispiel eines einzelnen 8259A PIC (PIC = Programmable Interrupt Controller)

1. Rette alle Register.
2. Führe nicht unterbrechbaren Code aus.
3. Ermögliche Unterbrechungen.
4. Führe unterbrechbaren Code aus.
5. Unterbinde Unterbrechungen
6. Sende nichtspezifischen End-Of-Interrupt an PIC.
7. Restauriere Register.
8. Rückkehr aus Unterbrechungsbehandler.

Kaskadierung von Unterbrechungen



Die normale Prioritätsanordnung der Unterbrechungen ist:
IRQ0, IRQ1, IRQ8 - IRQ15, IRQ3 - IRQ7

Struktur einer Unterbrechungsbehandlungsroutine

bei Kaskadierung von 8259A Bausteinen:

B = "Slave" und A = "Master":

1. Rette alle Register.
2. Führe nicht unterbrechbaren Code aus.
3. Ermögliche Unterbrechungen.
4. Führe unterbrechbaren Code aus.
5. Unterbinde Unterbrechungen.
6. Sende nichtspezifischen End-Of-Interrupt an PIC B.
7. Lese ISR des PIC B.

Struktur einer Unterbrechungsbehandlungsroutine

8. Ist niedriger priorisierte Unterbrechung von B in Bearbeitung?

Nein: Sende nichtspezifischen End-Of-Interrupt an A.
 Restauriere Register.
 Rückkehr von Interrupt.

Ja: Restauriere Register.
 Rückkehr von Interrupt.

Unterbrechungen im PC

und Zuordnung von Unterbrechungsleitungen und Unterbrechungsvektoren:

1. Der 80x86 erlaubt bis zu 256 Unterbrechungen. Die Adressen der Unterbrechungsrountinen sind 0:0 bis 0:03fch. Unterbrechung ist hier ein Oberbegriff für Ausnahme, Software-Interrupt und Hardware-Interrupt.
2. Beispiele für Ausnahmen:
 - INT 0: Divisionsfehler
 - INT 1: Einzelschrittmodus
 - INT 5: Grenzverletzungen bei Array-Zugriff
 - INT 6: Falscher Befehlscode

Geräte-Unterbrechungen im AT

Leitung	U-Vektor	Gerät
IRQ 0	8	Zeitunterbrechung
IRQ 1	9	Tastatur
IRQ 2	0ah	Kaskade für Controller 2
IRQ 3	0bh	Serielle Schnittstelle 2
IRQ 4	0ch	Serielle Schnittstelle 1
IRQ 5	0dh	Parallele Schnittstelle 2
IRQ 6	0eh	Diskettenlaufwerk
IRQ 7	0fh	Parallele Schnittstelle 1
IRQ 8	70h	Echtzeituhr
IRQ 9	71h	umgeleitet auf IRQ 2
IRQ 10–12	72h	reserviert
IRQ 13	75h	FPU Unterbrechung
IRQ 14	76h	Festplattenlaufwerk
IRQ 15	77h	reserviert

Unterbrechungsbehandlung für PC-Bus im Überblick

Schritte der Unterbrechungsbehandlung im System:

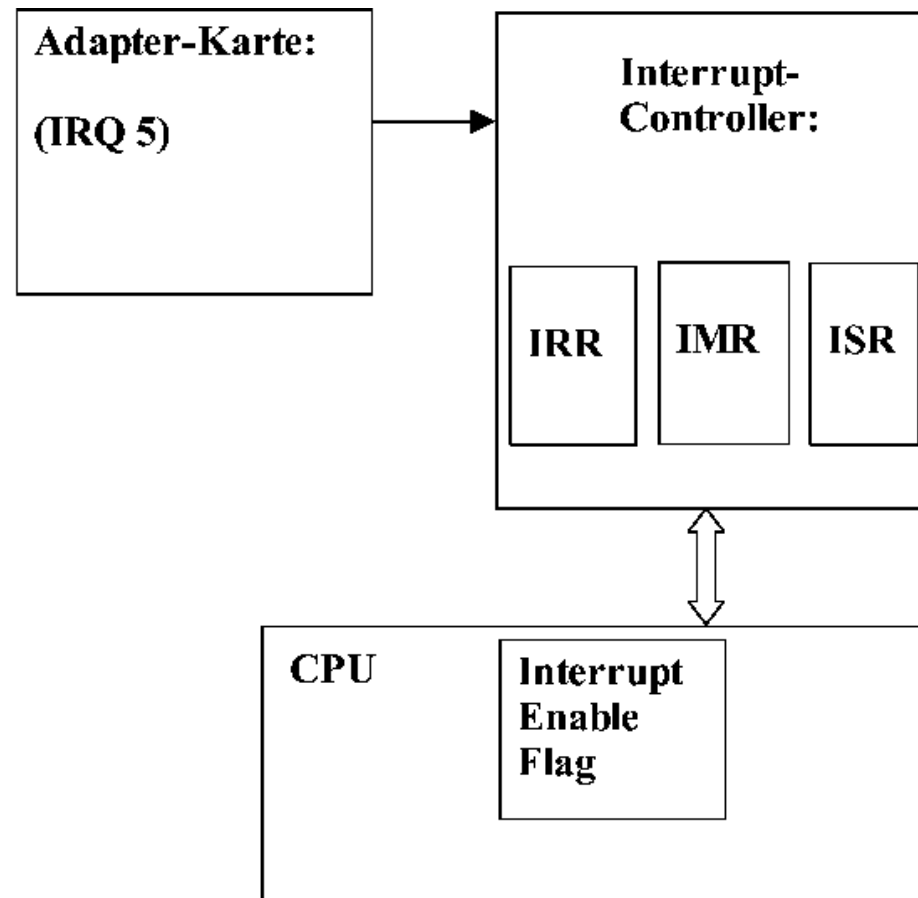
- 8259A akzeptiert Unterbrechung,
- 8259A bestimmt Priorität,
- 8259A signalisiert Unterbrechung,
- Prozessor bestätigt Unterbrechung,
- 8259A legt Unterbrechungsvektor auf Unterbrechung.

Unterbrechungsbehandlung für PC-Bus im Überblick

Schritte der Unterbrechungsbehandlung durch Prozessor:

- Prozessor erhält Unterbrechungsaufforderung,
- Prozessor beendet gegenwärtigen Befehl,
- Prozessor bestätigt Unterbrechungssignal,
- Prozessor rettet Statuswort und Befehlszähler,
- Prozessor lädt neues Statuswort und neue Befehlsadresse,
- der volle Maschinenzustand wird von Unterbrechungsroutine gerettet,
- Unterbrechung wird bearbeitet,
- Maschinenzustand wird restauriert,
- altes Statuswort und alter Befehlszähler werden restauriert.

Ein Beispiel zum Ablauf einer Unterbrechung im PC



Ein Beispiel zum Ablauf einer Unterbrechung im PC

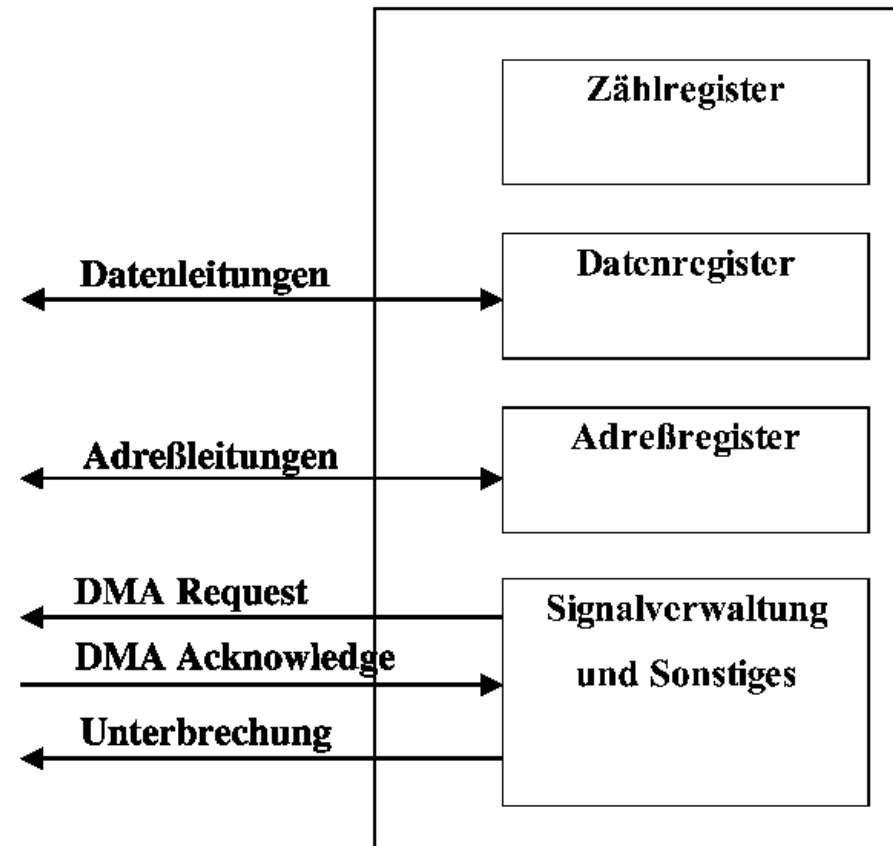
Ablauf einer Unterbrechung, hier als Beispiel mit Unterbrechung 5:

1. Ein Gerät meldet eine Unterbrechung über Eingang IR 5 des Interrupt-Controllers an.
2. Im IRR des Controllers wird das Bit 5 gesetzt.
3. Es wird geprüft, ob Unterbrechung 5 zulässig ist, bei Zulässigkeit wird zu Schritt 4 übergegangen.
4. Es wird überprüft, ob eine höher priorisierte Unterbrechung ansteht.
5. Falls nein, wird die CPU über Leitung INT benachrichtigt.
6. Falls Unterbrechungen zulässig sind, sendet die CPU in kurzem Abstand zweimal ein INTA-Signal.

7. Bei Empfang des ersten INTA-Signals wird das Bit 5 im ISR gesetzt und Bit 5 im IRR gelöscht. Bei Empfang des zweiten INTA-Signals sendet der Controller die Adresse der Unterbrechungs-Behandlungs-Routine an die CPU.
8. Die CPU rettet den Zustand des gegenwärtigen Prozesses auf den Prozeßkeller und beginnt die Ausführung der Unterbrechungs-Routine.
9. Die Unterbrechungsbehandlung wird ausgeführt, in ihrem Verlauf wird das Unterbrechungssignal zurückgenommen.
10. Die Unterbrechungs-Routine sendet End-of-Interrupt an den Controller, Datenwert 20h an Port 20h.
11. Der End-of-Interrupt Befehl löscht das Bit 5 im ISR, nun kann wieder eine neue Unterbrechung über IR 5 behandelt werden.
12. Mit dem Befehl IRET ("Return from Interrupt") wird die Unterbrechungsbehandlung beendet und der unterbrochene Prozeß wieder aufgenommen.

DMA ("Direct Memory Access")

Ein DMA-Modul ist ein sehr einfacher E/A-Modul:



Bemerkung:

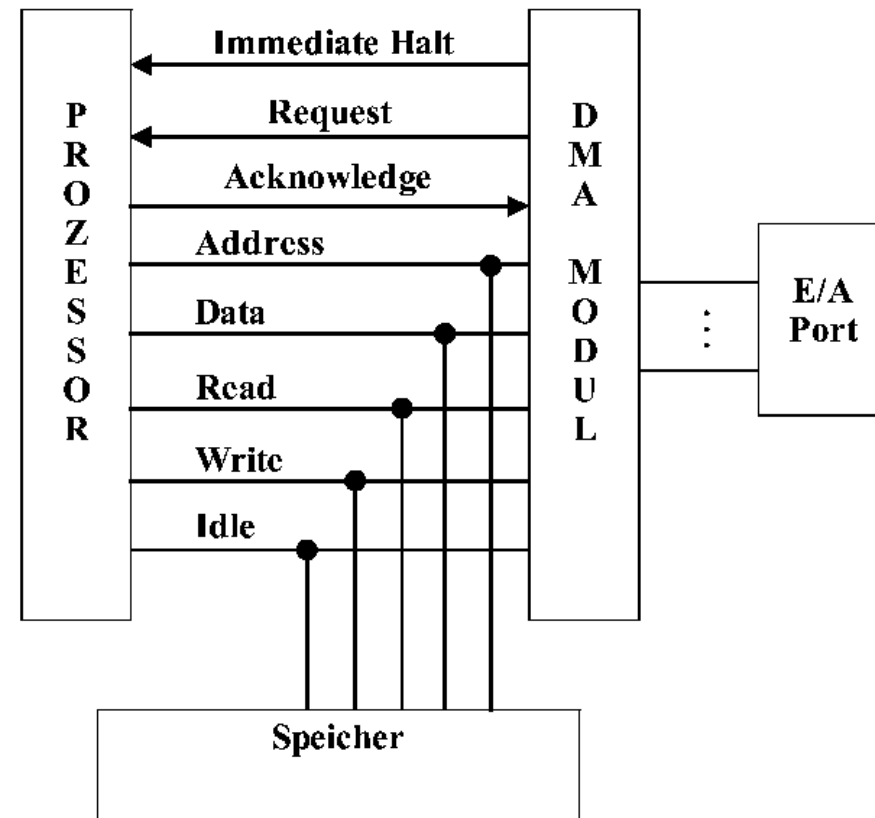
Die Anbindung des DMA-Moduls an die Ein- und Ausgabegeräte wird nicht gezeigt.

Zur Kommunikation Prozessor - DMA-Modul

Bemerkung:

Im Bild sind drei Möglichkeiten der exklusiven Nutzung des Systembusses durch den DMA-Modul vorgesehenen:

1. über Immediate Halt,
2. über Abfragen der Idle-Leitung,
3. ein Request - Acknowledge - Protokoll.



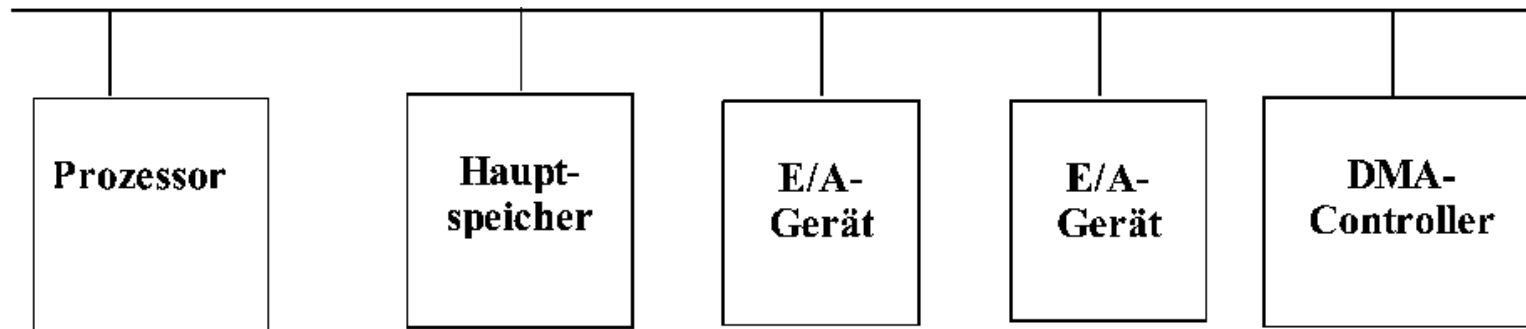
Grober Ablauf eines DMA-Transfers:

1. Der Prozessor lädt das Zählregister und das Adreßregister des DMA-Moduls mit ihren Anfangswerten. Im Adreßregister steht die Speicheradresse, im Zählregister die Wortzahl.
2. Ist der DMA-Modul bereit, einen Datentransfer durchzuführen, aktiviert er das DMA Request Signal. Der Prozessor honoriert die Busanforderung mit dem Signal DMA Acknowledge.
3. Ein Datenwort zwischen Speicher und DMA-Modul wird transferiert. Zähler und Adresse werden aktualisiert.
4. Der DMA-Modul gibt den Bus frei durch Deaktivierung des DMA Request Signals. Der Prozessor reagiert mit der Deaktivierung des Signals DMA Acknowledge.
5. Falls der Zähler ungleich 0 ist, erfolgt eine Wiederholung des Zyklus ab Schritt 2. Falls der Zähler gleich 0 ist, wird der Prozessor durch ein Unterbrechungssignal benachrichtigt.

Bemerkung: Die zeitweilige, kurzfristige Benutzung des Systembusses zur Datenübertragung durch den DMA-Modul ist unter dem Namen “cycle stealing” bekannt.

DMA-Konfiguration (1)

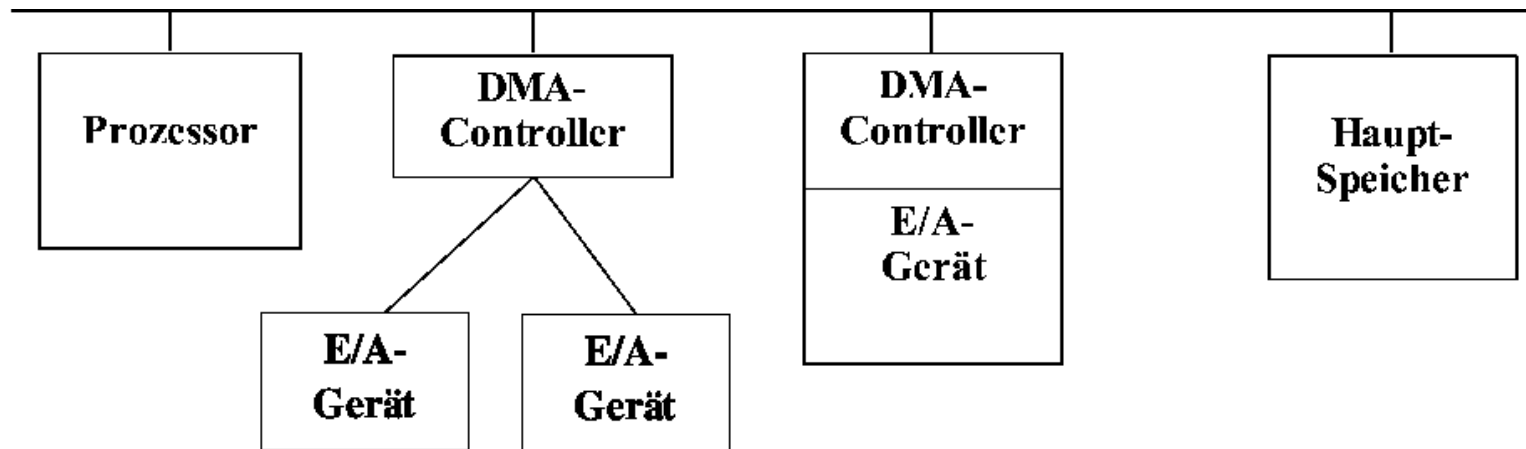
Ein-Bus-System:



Bemerkung: In dieser Konfiguration ist es denkbar, daß jeder Datentransfer den Bus zweimal beansprucht, ein Datum wird zwischen E/A-Gerät und DMA-Controller und zwischen DMA-Controller und Hauptspeicher übertragen; der Prozessor wird zweimal gestört.

DMA-Konfiguration (2)

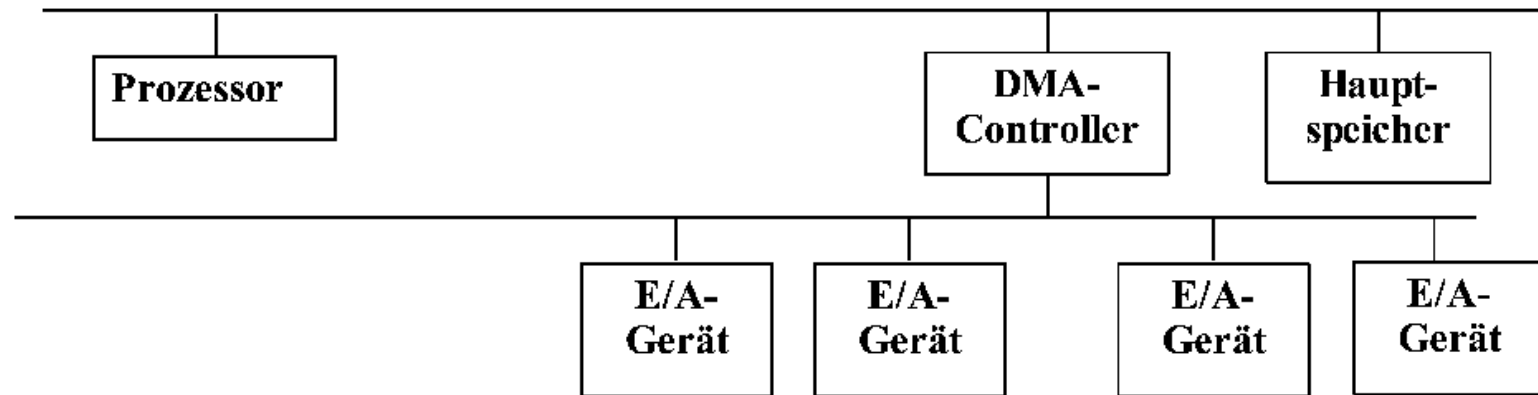
Ein-Bus-System mit integrierten DMA-Controllern:



Bemerkung: Jeder Datentransfer beansprucht den Bus nur einmal.

DMA-Konfiguration (3)

Separater E/A-Bus:



Bemerkung: Jeder Datentransfer beansprucht den Bus nur einmal, daher wird der Prozessor nur einmal pro Datentransfer gestört.

Literatur

- [1] Randal E. Bryant and David O'Hallaron. *Computer Systems*. Pearson Education, Inc., New Jersey, 2003.
- [2] N. Hendrich. Pc technologie: Cd-rom und dvd.
<http://tams-www.informatik.uni-hamburg.de/lehre/ss2002/pc-technologie/09-cdrom.pdf>.
- [3] M. Lehmann. T3: Kapitel 4 (externspeicher).
<http://www.informatik.uni-hamburg.de/TKRN/world/abro/T3.WS0102/t3k4ws02.pdf>.
- [4] Andrew S. Tanenbaum and James Goodman. *Computerarchitektur*. Pearson Studium München, 2001.