3. Foliensatz Betriebssysteme

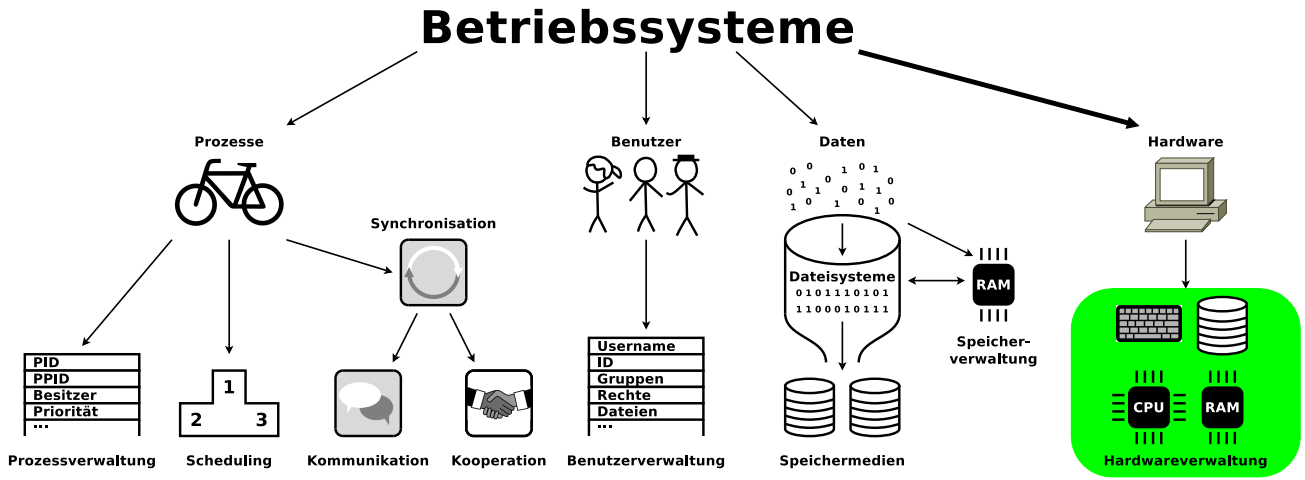
Prof. Dr. Christian Baun

Frankfurt University of Applied Sciences
(1971–2014: Fachhochschule Frankfurt am Main)
Fachbereich Informatik und Ingenieurwissenschaften
christianbaun@fb2.fra-uas.de

Lernziele dieses Foliensatzes

- Am Ende dieses Foliensatzes kennen/verstehen Sie...
  - die **Rechnerarchitektur – Von-Neumann-Architektur**
  - die **Hardware-Komponenten** eines Computers
    - **Hauptprozessor** (CPU) und **Busleitungen**
  - den Unterschied zwischen **zeichen-** und **blockorientierten Geräten**
    - wie **Daten von Ein- und Ausgabegeräten gelesen** werden
  - was für **Speicher** in Computern existiert
    - die Architektur der **Speicherhierarchie** (Speicherpyramide)
    - was **Primär-/Sekundär-/Tertiärspeicher** ist
    - die Arbeitsweise der **Speicherhierarchie**
    - die **Cache-Schreibstrategien** (Write-Back und Write-Through)

Übungsblatt 3 wiederholt die für die Lernziele relevanten Inhalte dieses Foliensatzes



Warum das Ganze?

- Warum besprechen wir auch die Arbeitsweise der CPU, des Speichers und der Bussysteme in der Vorlesung Betriebssysteme?

Edsger W. Dijkstra

„In der Informatik geht es genau so wenig um Computer, wie in der Astronomie um Teleskope.“

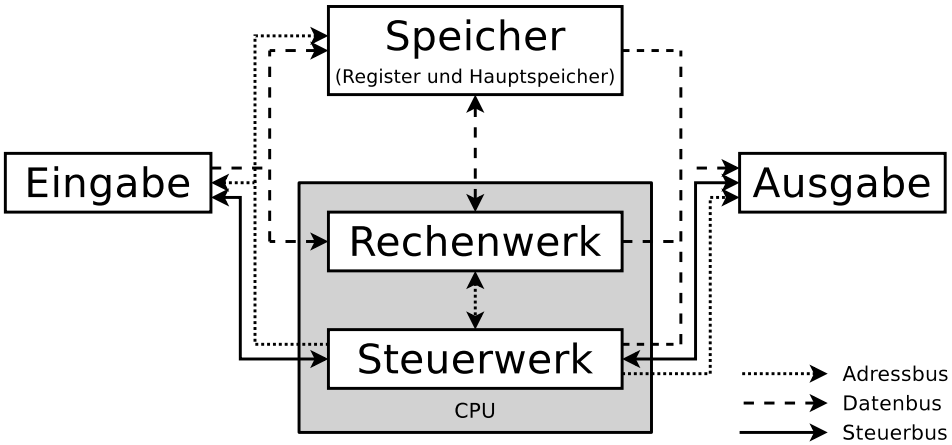
- Betriebssysteme erleichtern den Benutzern und deren Prozessen die Nutzung der Hardware
- Wer die Arbeitsweise der CPU, des Speichers und der Bussysteme nicht kennt und versteht, versteht auch nicht die Arbeitsweise der Betriebssysteme

Von-Neumann-Architektur

Bildquelle: United States Department of Energy (Public Domain)

- Idee und Aufbau des Universalrechners, der nicht an ein festes Programm gebunden ist und über Ein-/Ausgabegeräte verfügt
  - Entwickelt 1946 von John von Neumann
  - Nach ihm benannt ist die **Von-Neumann-Architektur**, bzw. der **Von-Neumann-Rechner**
- Im Von-Neumann-Rechner werden Daten und Programme **binär kodiert** und liegen im **gleichen Speicher**
- Wesentliche Ideen der Von-Neumann-Architektur wurden bereits 1936 von Konrad Zuse ausgearbeitet und 1937 in der Zuse Z1 realisiert
- Von Neumanns Verdienste:
  - Er hat sich als erster wissenschaftlich, mathematisch mit der Konstruktion von Rechenmaschinen beschäftigt

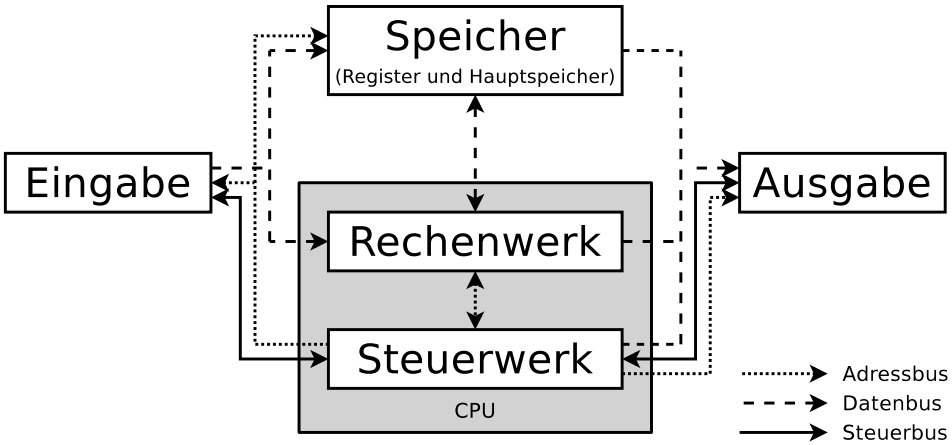
Der Hauptprozessor – Central Processing Unit (CPU)



- Gemäß der Von-Neumann-Architektur befindet sich der Speicher außerhalb der CPU
- In modernen Rechnersystemen befinden sich Teile des Speichers (z.B. Register und mehrere Cache-Ebenen) innerhalb der CPU

- Die meisten Komponenten eines Computers sind passiv und werden durch die CPU gesteuert
- Programme sind Folgen von Maschineninstruktionen, die in aufeinander folgenden Speicheradressen abgelegt sind
- Bei der Programmausführung setzt die CPU die Maschineninstruktionen Schritt für Schritt um
- Eine CPU besteht aus 2 Komponenten:
  - **Rechenwerk** und **Steuerwerk**
- Zudem sind **Ein-/Ausgabegeräte** ($\Longrightarrow$ Folie 15) und **Speicher** ($\Longrightarrow$ Folie 21) nötig

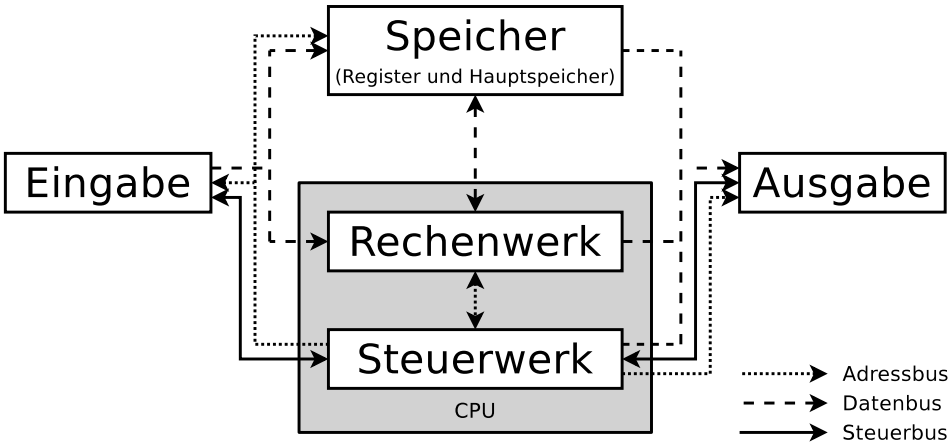
Komponenten der CPU



- Gemäß der Von-Neumann-Architektur befindet sich der Speicher außerhalb der CPU
- In modernen Rechnersystemen befinden sich Teile des Speichers (z.B. Register und mehrere Cache-Ebenen) innerhalb der CPU

- **Steuerwerk** bzw. **Leitwerk** bzw. **Befehlswerk** (Control Unit)
  - Interpretiert Befehle, koordiniert die anderen CPU-Komponenten, steuert die Ein-/Ausgabe-Einheiten und den Steuerbus
- **Rechenwerk** bzw. **Arithmetic Logic Unit** (ALU)
  - Manipulation von Daten und Adressen
  - Führt die logischen (NOT, AND, OR, XOR,...) und mathematischen (ADD, SUB,...) Operationen aus
- **Speicher**
  - Register zur kurzfristigen Speicherung von Operanden und Adressen
    - Arbeiten mit der selben Geschwindigkeit, wie der Rest der CPU
  - Cache und Hauptspeicher = Speicher für Programme und Daten

Von-Neumann-Zyklus (*Fetch-Decode-Execute Cycle*)

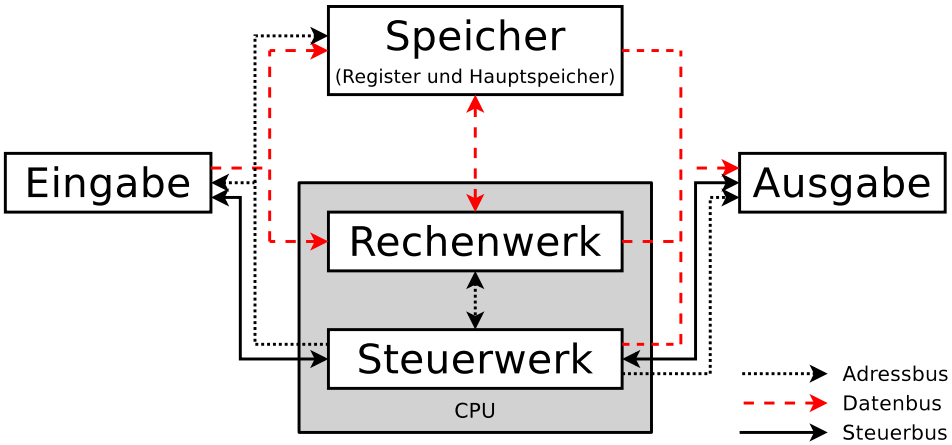


- Wiederholt die CPU vom Systemstart bis der Computer gestoppt wird
  - Jede Phase kann mehrere Takte in Anspruch nehmen

1. **FETCH**: Abzuarbeitenden Befehl aus dem Speicher in das Befehlsregister (*Instruction Register*) kopieren
2. **DECODE**: Steuerwerk löst den Befehl in Schaltinstruktionen für das Rechenwerk auf
3. **FETCH OPERANDS**: Eventuell verfügbare Parameter (Operanden) für den Befehl aus dem Speicher holen
4. **EXECUTE**: Rechenwerk führt den Befehl aus
5. **UPDATE PROGRAM COUNTER**: Befehlszähler (*Program Counter*) wird auf den nächsten Befehl gesetzt
6. **WRITE BACK**: Das Ergebnis des Befehls wird in einem Register oder im Hauptspeicher gespeichert oder zu einem Ausgabegerät gesendet

- Auch moderne CPUs und Rechnersysteme arbeiten nach dem Von-Neumann-Zyklus und Von-Neumann-Rechner
- Ausnahme: Ein einzelner Bus um Eingabe-/Ausgabe-Geräte direkt mit der CPU zu verbinden, ist nicht mehr möglich und Teile des Speichers (Register, L1/L2/L3-Cache) befinden sich innerhalb der CPU

Datenbus



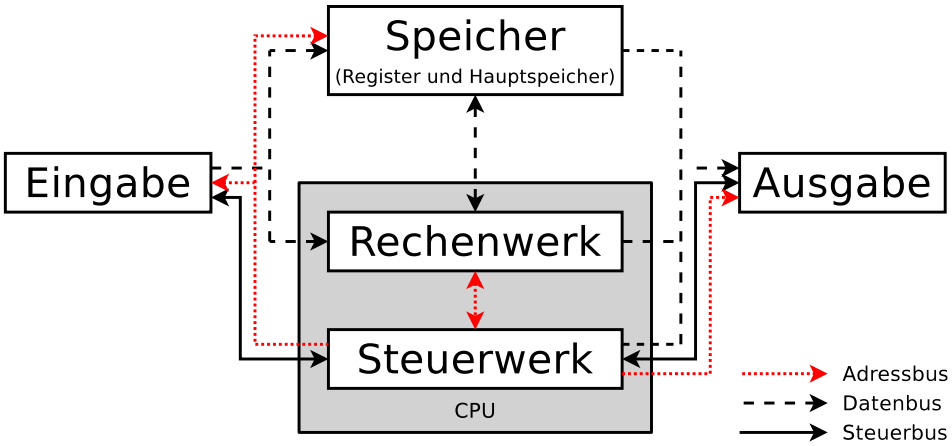
- Überträgt Daten zwischen CPU, Arbeitsspeicher und Peripherie
- Anzahl der Datenbusleitungen legt fest, wie viele Bytes pro Takt übertragen werden können

- Üblicherweise ist die Anzahl der Datenbusleitungen gleich der Größe der Arbeitsregister im Rechenwerk
- Datenbusbreite moderner CPUs: 64 Leitungen
  - Die CPU kann somit 64 Datenbits innerhalb eines Taktes zum und vom Arbeitsspeicher weg übertragen

Datenbusbreite einiger CPUs

CPU	Datenbus
4004, 4040	4 Bits
8008, 8080, 8085, 8088	8 Bits
8086 (XT), 80286 (AT), 80386SX	16 Bits
80386DX, 80486SX/DX/DX2/DX4	32 Bits
Pentium I/MMX/II/III/IV/D/M, Celeron, Core Solo/Duo, Core 2 Duo, Core 2 Extreme, Pentium Pro, Pentium Dual-Core, Core 2 Quad, Core i7, Itanium, AMD Phenom-II, Itanium 2, AMD64	64 Bits

Adressbus

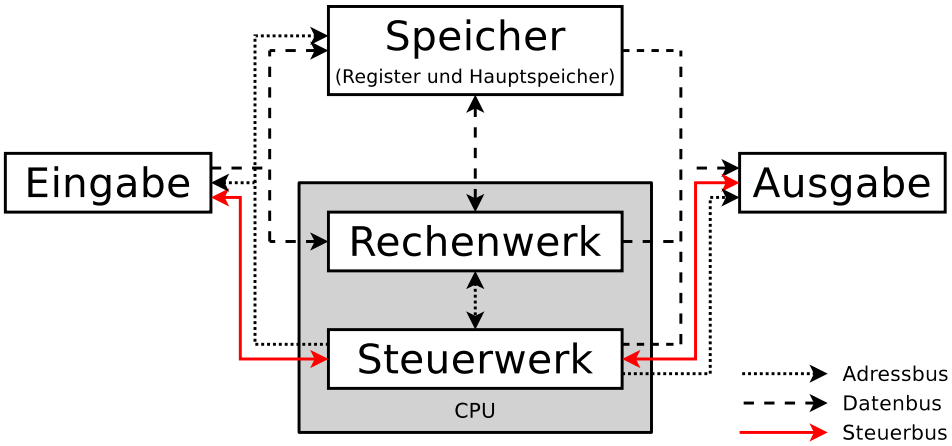


- Überträgt Speicheradressen
- Speicheradressen und E/A-Geräte werden über den Adressbus angesprochen (adressiert)
- Anzahl der Busleitungen legt die maximale Anzahl adressierbarer Speicheradressen fest

Adressbusbreite einiger CPUs

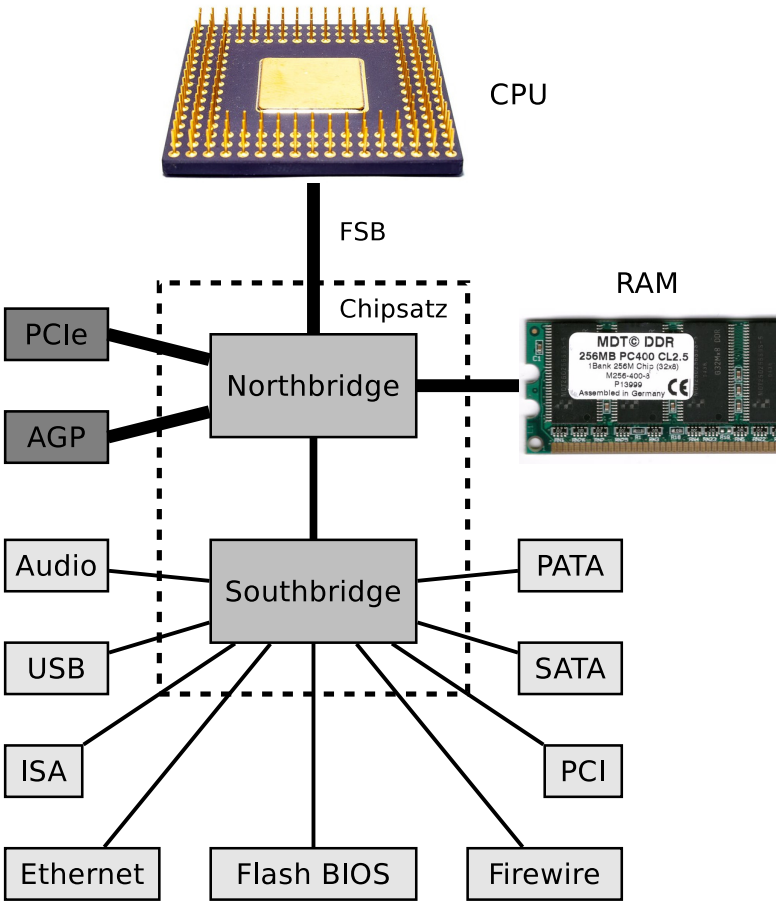
CPU	Adressbus	max. adressierbar
4004, 4040	4 Bits	2^4 = 16 Bytes
8008, 8080	8 Bits	2^8 = 256 Bytes
8085	16 Bits	2^{16} = 65 kB
8088, 8086 (XT)	20 Bits	2^{20} = 1 MB
80286 (AT)	24 Bits	2^{24} = 16 MB
80386SX/DX, 80486SX/DX/DX2/DX4, Pentium I/MMX/II/III/IV/D/M, Celeron,	32 Bits	2^{32} = 4 GB
Core Solo/Duo, Core 2 Duo/Extreme/Quad, Pentium Pro, Pentium Dual-Core, Core i7	36 Bits	2^{36} = 64 GB
Itanium	44 Bits	2^{44} = 16 TB
AMD Phenom-II, Itanium 2, AMD64	48 Bits	2^{48} = 256 TB

Steuerbus



- Überträgt Kommandos (z.B. Lese- und Schreibanweisungen) von der CPU und Statusmeldungen von den Peripheriegeräten
- Unterschied zwischen Adressbus und Steuerbus:
  - Komponenten des Computers werden über den Adressbus angesprochen und über den Steuerbus angewiesen, was sie zu tun haben
- Enthält auch Leitungen, über die E/A-Geräte der CPU Unterbrechungsanforderungen (Interrupts) signalisieren
- Typische Busbreite: $\leq$ 10 Leitungen

Busse in modernen Rechnersystemen



- Verbindendes Element: **Chipsatz**
- Der Chipsatz besteht aus. . .
  - **Northbridge**
    - Liegt dicht an der CPU, um Daten schnell übertragen zu können
    - Zuständig für Anbindung des Hauptspeicher und der Grafikkarte(n) an die CPU
  - **Southbridge**
    - Für „langsamere“ Verbindungen
- **Front-Side-Bus** (FSB) heißt der Bus zwischen CPU und Chipsatz
  - Er enthält den Adressbus, Datenbus und Steuerbus

Ausgewählte Bussysteme

- **Aus Geschwindigkeits- und Kostengründen werden zunehmend Teile des Chipsatzes in die CPU verlagert**
  - Anders als in der Von-Neumann-Architektur werden Geräte nicht direkt mit der CPU verbunden
  - Rechnersysteme enthalten heute verschiedenen seriellen und parallele Bussysteme, die für die jeweilige Erfordernisse ausgelegt sind
  - Immer häufiger werden Punkt-zu-Punkt-Verbindungen eingesetzt
  - Eingabe-/Ausgabecontroller arbeiten als Vermittler zwischen den Geräten und der CPU
- Einige Bussysteme:

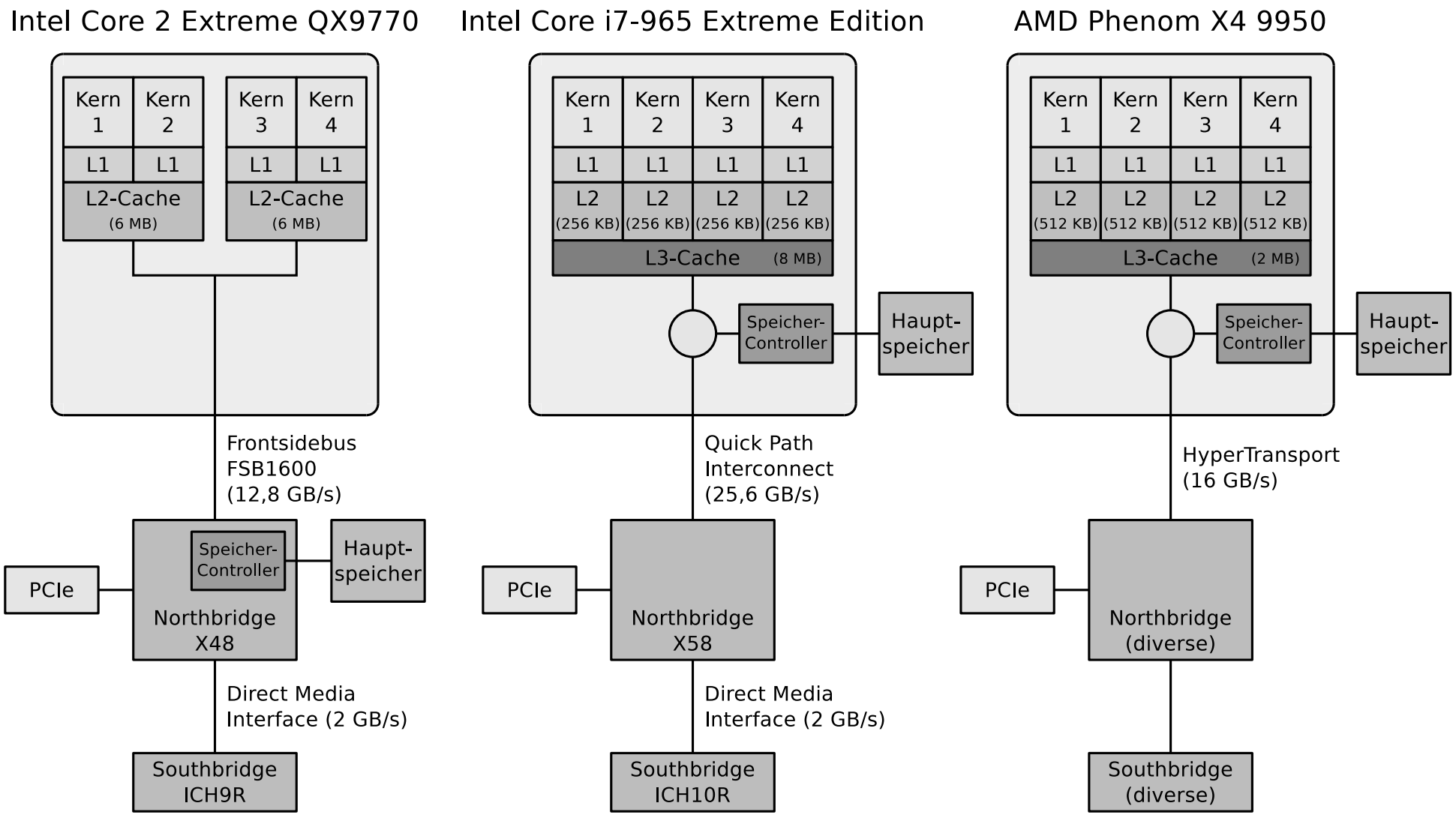
	Rechner-interne Busse	Rechner-externe Busse
Parallele Busse	PATA (IDE), PCI, ISA, SCSI	PCMCIA, SCSI
Serielle Busse	SATA, PCI-Express	Ethernet, FireWire, USB, eSATA

Zunehmende Verlagerung von Funktionen des Chipsatzes in die CPU

Die folgenden beiden Folien zeigen exemplarisch, wie um 2008 zunehmend der Speichercontroller von der Northbridge in die CPU verlagert wurde und um 2009 auch die PCIe-Anbindung

Verlagerung des Speichercontrollers in die CPU

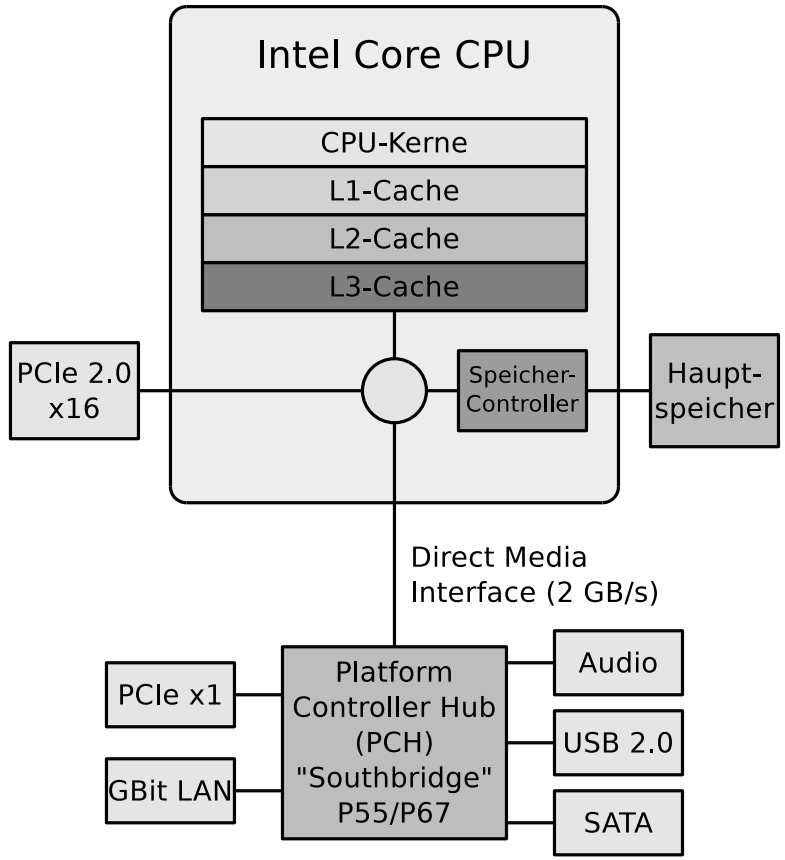
Quelle: c't 25/2008



- Ergebnis: Die Northbridge enthält nur noch den Controller für PCIe

Verlagerung der Northbridge in die CPU

- Bei einigen modernen Systemen ist die Northbridge in die CPU verlagert
- Vorteil: Geringere Kosten für das Gesamtsystem



- Die Abbildung zeigt die Verteilung der Funktionalitäten bei den Chipsatzgenerationen Intel P55 und P67 aus den Jahren 2009 und 2011
- Ab dieser Zeit heißt die Southbridge auch **Platform Controller Hub** (PCH)

Zeichenorientierte und Blockorientierte Geräte

Wir wissen bereits...

- Ein-/Ausgabegeräte sind ein wesentlicher Bestandteil der Von-Neumann-Architektur, aber einige wichtige Fragen sind noch nicht beantwortet
- Welche Gruppen von Ein-/Ausgabegeräten gibt es?
- Wie können Prozesse mit Ein-/Ausgabegeräten interagieren?

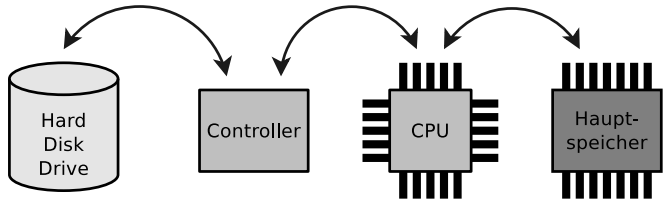
- Geräte an Computersystemen werden bezüglich der kleinsten Übertragungseinheit unterschieden:
- **Zeichenorientierte Geräte**
  - Bei Ankunft/Anforderung jedes einzelnes Zeichens wird immer mit der CPU kommuniziert
  - Beispiele: Maus, Tastatur, Drucker, Terminal und Magnetband
- **Blockorientierte Geräte**
  - Datenübertragung findet erst statt, wenn ein kompletter Block (z.B. 1-4 kB) vorliegt
  - Beispiele: Festplatte, SSD, CD-/DVD-Laufwerk und Disketten-Laufwerk

Daten einlesen

- Soll z.B. ein Datensatz von einer Festplatte gelesen werden, sind folgende Schritte nötig:
  1. Die CPU bekommt von einem Prozess die Anforderung, einen Datensatz von einer Festplatte zu lesen
  2. Die CPU schickt dem Controller mit Hilfe des Treibers einen I/O-Befehl
  3. Der Controller lokalisiert den Datensatz auf der Festplatte
  4. Der Prozess erhält die angeforderten Daten
- Es gibt 3 Konzepte, wie Prozesse im Computer Daten einlesen können:
  - **Busy Waiting** (geschäftiges bzw. aktives Warten)
  - **Interrupt-gesteuert**
  - **Direct Memory Access** (DMA)

Busy Waiting (geschäftiges bzw. aktives Warten)

- Der Treiber sendet die Anfrage an das Gerät und wartet in einer **Endlosschleife**, bis der Controller anzeigt, dass die Daten bereit stehen
  - Stehen die Daten bereit, werden sie in den Speicher geschrieben und die Ausführung des Prozesses geht weiter
- Beispiel: Zugriffsprotokoll **Programmed Input/Output** (PIO)
  - Die CPU greift via Lese- und Schreibbefehle auf die Speicherbereiche der Geräte zu und kopiert so Daten zwischen den Geräten und dem Hauptspeicher
- **Vorteil**:
  - Keine zusätzliche Hardware nötig
- **Nachteile**:
  - Belastet die CPU
  - Verlangsamt die gleichzeitige Abarbeitung mehrerer Prozesse
    - Grund: Regelmäßig muss die CPU überprüfen, ob die Daten bereit stehen



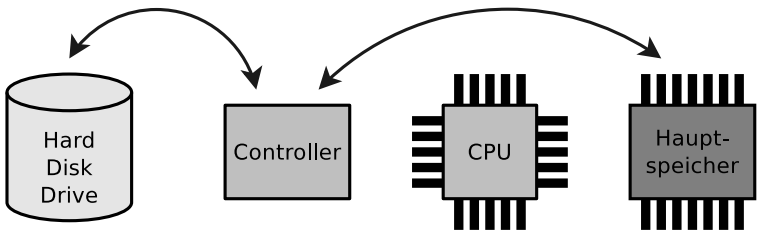
Beispiele: PATA-Festplatten im PIO-Modus, serielle Schnittstelle, parallele Schnittstelle, PS/2-Schnittstelle für Tastatur und Maus

Interrupt-gesteuert

- Voraussetzung: Ein **Interrupt-Controller** und **Leitungen** im Steuerbus für das Senden der **Interrupts**
- Der Treiber initialisiert die E/A-Aufgabe und wartet auf einen Interrupt (Unterbrechung) durch den Controller $\Longrightarrow$ Der Treiber *schläft*
  - Die CPU ist während des Wartens auf den Interrupt nicht blockiert und das Betriebssystem kann die CPU anderen Prozesse zuweisen
  - Kommt es zum Interrupt, wird der Treiber dadurch *geweckt* $\Longrightarrow$ bekommt Zugriff auf die CPU
    - Danach holt die CPU die Daten vom Controller und legt sie in den Speicher
    - Anschließend wird die CPU dem unterbrochenen Prozess zugewiesen, der seine Abarbeitung fortsetzen kann
- **Vorteile**:
  - Die CPU wird nicht blockiert
  - Gleichzeitige Abarbeitung mehrerer Prozesse wird nicht verlangsamt
- **Nachteile**:
  - Zusätzliche Hardware (Interrupt-Controller) ist nötig

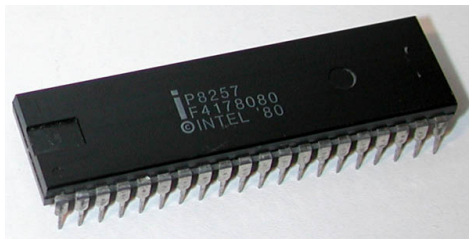
Direct Memory Access

- Voraussetzung: **DMA-Controller**
  - Kann Daten direkt zwischen Arbeitsspeicher und E/A-Gerät übertragen
    - Beispiele: HDD/SSD, Soundkarte, Netzwerkkarte, TV-/DVB-Karte
  - Löst nach der Datenübertragung einen Interrupt aus



- Beispiel: **Ultra-DMA** (UDMA)
  - Nachfolgeprotokoll des PIO-Modus
  - Legt fest, wie Daten zwischen DMA-Controller und Arbeitsspeicher übertragen werden

- **Vorteile**:
  - Vollständige Entlastung der CPU
  - Gleichzeitige Abarbeitung mehrerer Prozesse wird nicht verlangsamt
- **Nachteile**:
  - Zusätzliche Hardware (DMA-Controller) ist nötig
    - Seit Ende der 1980er Jahre im Chipsatz integriert



Bildquelle: `http://www.cpu-world.com/Support/82/Intel-P8257.jpg` (Verwendung für nicht-kommerzielle Nutzung und Lehre erlaubt)

Speicher

Wir wissen bereits...

- Digitale Datenspeicher sind ein wesentlicher Bestandteil der Von-Neumann-Architektur, aber einige wichtige Fragen sind noch nicht beantwortet
- Was für Technologien zur digitalen Datenspeicherung gibt es?
- Welche digitalen Datenspeicherkomponenten sind mit einem Computer verbunden?
- Wie sind die unterschiedlichen digitalen Datenspeicher organisiert und wie werden Sie vom Betriebssystem angesprochen?

- Speichert die Daten und die ausführbare Programme
- Unterschiedliche Speicher sind durch Busse verbunden und bilden eine Hierarchie
  $\Longrightarrow$ **Speicherpyramide** (siehe Folie 26)
- Grund für die Speicher-Hierarchie: Preis/Leistungsverhältnis
  $\Longrightarrow$ Je schneller ein Speicher ist, desto teurer und knapper ist er

Digitale Datenspeicher

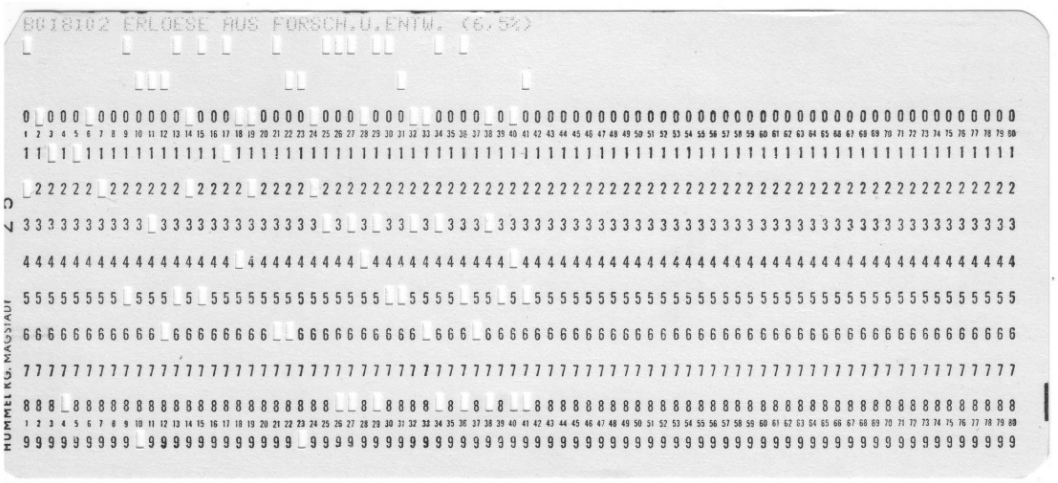
Speicher	Speicherung	Lesevorgang	Zugriffsart	Bewegliche Teile	Persistent
Lochstreifen	mechanisch		sequentiell	ja	ja
Lochkarte	mechanisch		sequentiell	ja	ja
Magnetband	magnetisch		sequentiell	ja	ja
Magnetkarte / Magnetstreifen	magnetisch		sequentiell	ja	ja
Trommelspeicher (Drum Memory)	magnetisch		wahlfrei	ja	ja
Kernspeicher	magnetisch		wahlfrei	nein	ja
Magnetblasenspeicher (Bubble Memory)	magnetisch		wahlfrei	nein	ja
Cache und Register (SRAM)	elektronisch		wahlfrei	nein	nein
Hauptspeicher (DRAM)	elektronisch		wahlfrei	nein	nein
Flashspeicher (USB-Stick, SSD, CF/SD-Karte)	elektronisch		wahlfrei	nein	ja
Compact Cassette (Datasette)	magnetisch		sequentiell	ja	ja
Diskette (Floppy Disk)	magnetisch		wahlfrei	ja	ja
Festplatte (Hard Disk)	magnetisch		wahlfrei	ja	ja
CD-ROM/DVD-ROM	mechanisch	optisch	wahlfrei	ja	ja
CD-R/CD-RW/DVD-R/DVD-RW	optisch		wahlfrei	ja	ja
MiniDisc	magneto-optisch	optisch	wahlfrei	ja	ja
Magneto Optical Disc (MO-Disk)	magneto-optisch	optisch	wahlfrei	ja	ja

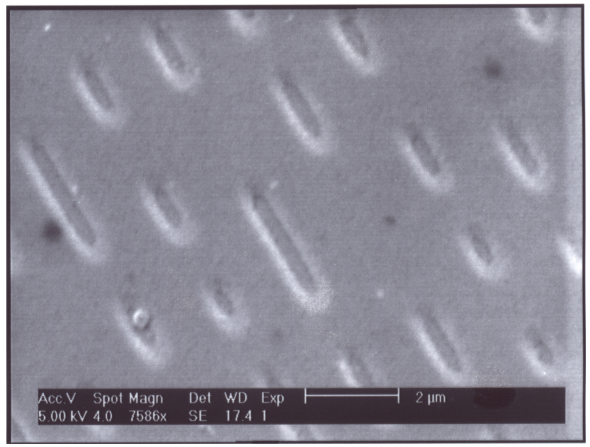
(graue Hintergrundfarbe bedeutet: veraltete bzw. überholte Technologie)

- **Wahlfreier Zugriff** heißt, dass das Medium nicht – wie z.B. bei Bandlaufwerken – von Beginn an sequentiell durchsucht werden muss, um eine bestimmte Stelle (Datei) zu finden
  - Die Köpfe von Magnetplatten oder ein Laser können in einer bekannten maximalen Zeit zu jedem Punkt des Mediums springen

Mechanische Datenspeicher

Bildquelle (Lochkarte): Eigenes Werk





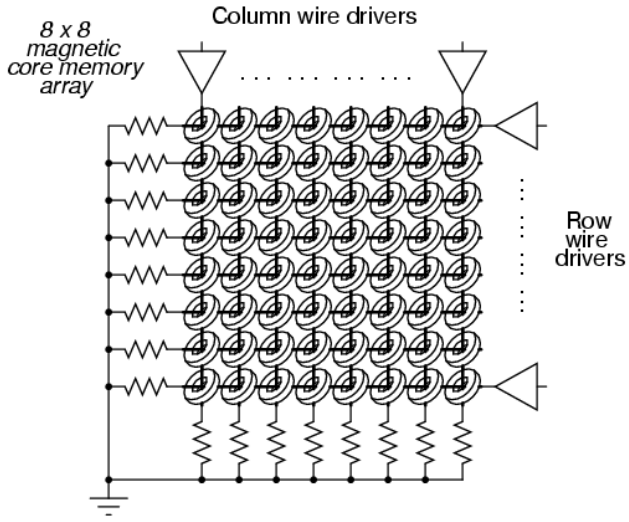
- Jede Lochkarte stellt üblicherweise eine Zeile Programmtext mit 80 Zeichen oder entsprechend viele binäre Daten dar
- Der dargestellte Lochstreifen hat 8 Löcher für Daten und eine kleinere Transportlochung
  - Man kann damit 1 Byte pro Zeile speichern
- Die Datenspeicherung auf CDs/DVDs erfolgt mit Pits (Gruben) und Lands (Flächen), die auf einem Kunststoff aufgebracht sind
  - Die Massenherstellung von CDs/DVDs heißt *pressen* und erfolgt via Spritzgussverfahren mit einem Negativ (*Stamper*)

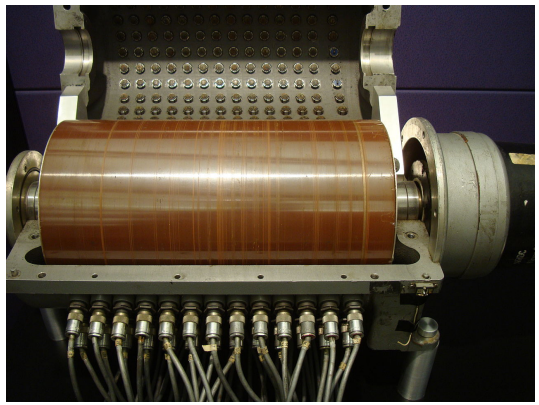
Bildquelle (Lochstreifen): TedColes. Wikimedia (CC0)
Bildquelle (Gepresste CD mit Pits und Lands): Stefan Kolb. Wikimedia (CC0)

Magnetische Datenspeicher

Bildquelle: http://sub.allaboutcircuits.com/images/04212.png

- Datenspeicherung auf magnetisierbarem Material
- Datenträger werden mit einem Lese-Schreib-Kopf gelesen und beschrieben
  - Ausnahme: Kernspeicher
- Der Lese-Schreib-Kopf kann beweglich (z.B. bei Festplatten) oder feststehend (z.B. bei Magnetbändern) sein
- **Rotierende Datenspeicher:**
  - Festplatte ($\Longrightarrow$ Foliensatz 4), Diskette, Trommelspeicher...
- **Nichtrotierende Datenspeicher:**
  - Kernspeicher, Magnetband, Magnetkarte/-streifen, Datasette, Magnetblasenspeicher...



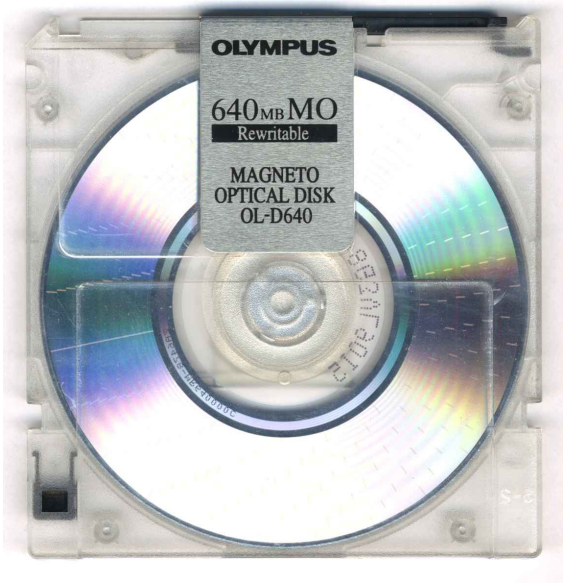


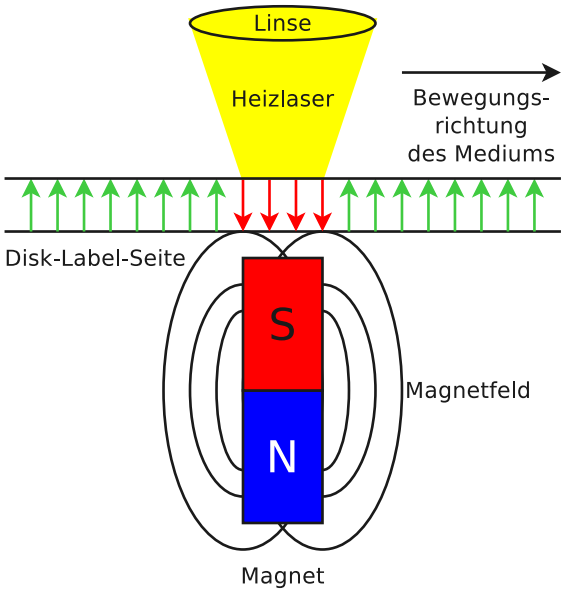
Bildquelle (Drum memory): Gregg Tavares (CC-BY-2.0)
Bildquelle (Floppy disks): George Chernilevsky (CC0)

Magneto-optische Datenspeicher

Bildquelle (Medium): ocrho. Wikimedia (CC0)

- Rotierendes Speichermedium
- Wird magnetisch beschrieben
- Der Datenträger muss zum Schreiben erhitzt werden
  - Erst oberhalb der *Curie-Temperatur* ist die magnetische Information änderbar
    - Vorteil: Unempfindlich gegen Magnetfelder
  - Das Erhitzen erfolgt via Laserstrahl
- Wird optisch ausgelesen
  - Unterschiedlich magnetisierte Bereiche reflektieren Licht unterschiedlich





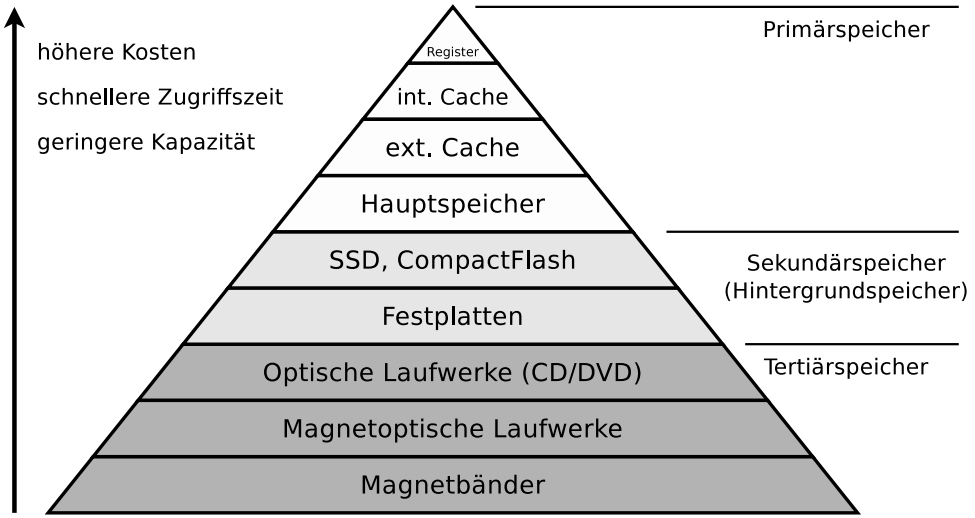
Magnet-optische Datenspeicherung kommt in der Praxis nicht mehr vor. Diese Technologie war etwa zwischen 1990 und 2005 vor allem in Japan einigermaßen populär. Der Grund, warum sie hier vorgestellt wird, ist nur um zu zeigen, das in den vergangen Jahrzehnten versucht wurde auf die den verschiedensten Arten von Speichermedien Daten abzulegen

Elektronischer Datenspeicher

- **Flüchtiger Speicher** (*volatile*) – Random-Access Memory (RAM)
  - Static Random-Access Memory (SRAM)
    - Informationen werden als Zustandsänderung einer bistabile Kippstufe (*Flipflop*) gespeichert
    - Informationen können beim Anliegen der Betriebsspannung beliebig lange gespeichert werden
    - Schneller und teurer als DRAM
    - Wird für **Cache** und CPU-interne **Register** verwendet
  - Dynamic Random-Access Memory (DRAM)
    - Informationen werden in Kondensatoren gespeichert
    - Benötigt ein periodisches Auffrischen der Informationen
    - Bei fehlender dauerhaft Betriebsspannung oder zu später Wiederauffrischung gehen die Information wegen der Leckströme verloren
    - Wird für **Hauptspeicher** verwendet
- **Nichtflüchtiger Speicher** (*non-volatile*)
  - Read-Only Memory (ROM)
    - z.B. Electrically Erasable Programmable ROM (EEPROM)
  - Flash-Speicher $\Longrightarrow$ Foliensatz 4

Speicherpyramide

- Primärspeicher und Sekundärspeicher sind permanent mit dem Computer verbunden
  - Vorteil: Geringe Zugriffszeiten auf gespeicherte Daten

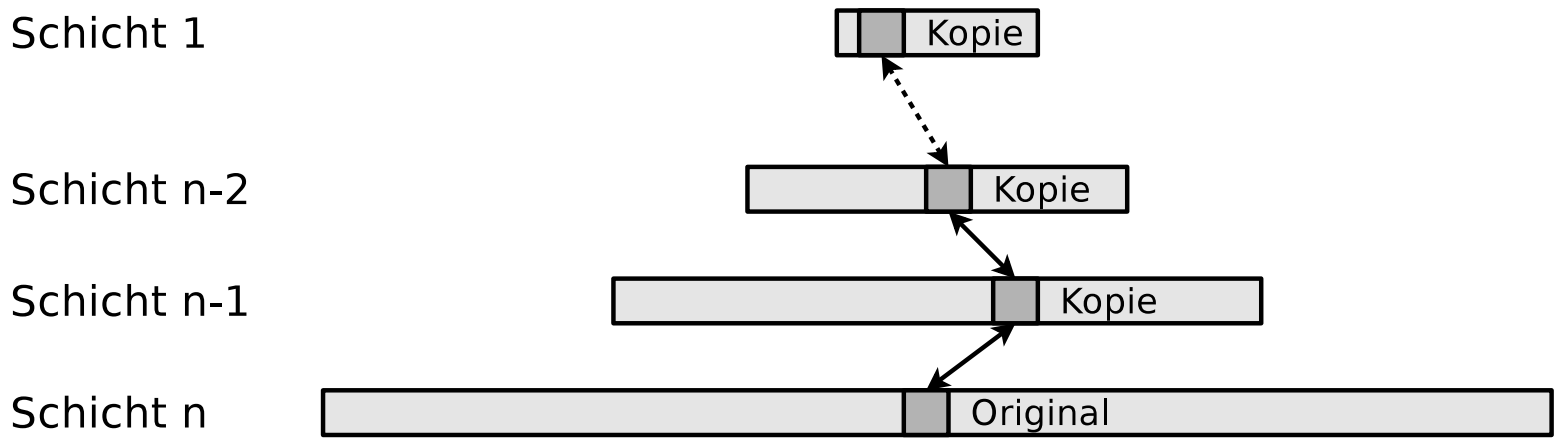


- **Primärspeicher**: Darauf greift die CPU direkt zu
- **Sekundärspeicher**: Wird über einen Controller angesprochen
- **Tertiärspeicher**: Nicht dauerhaft mit dem Rechner verbunden. Hauptaufgabe ist Archivierung

- Tertiärspeicher wird unterschieden in:
  - **Nearlinespeicher**: Werden automatisch und ohne menschliches Zutun dem System bereitgestellt (z.B. Band-Library)
  - **Offlinespeicher**: Medien werden in Schränken oder Lagerräumen aufbewahrt und müssen von Hand in das System integriert werden
    - Streng genommen sind Wechselfestplatten auch Offlinespeicher

Arbeitsweise der Speicherhierarchie

- Beim ersten Zugriff auf ein Datenelement, wird eine Kopie erzeugt, die entlang der Speicherhierarchie nach oben wandert

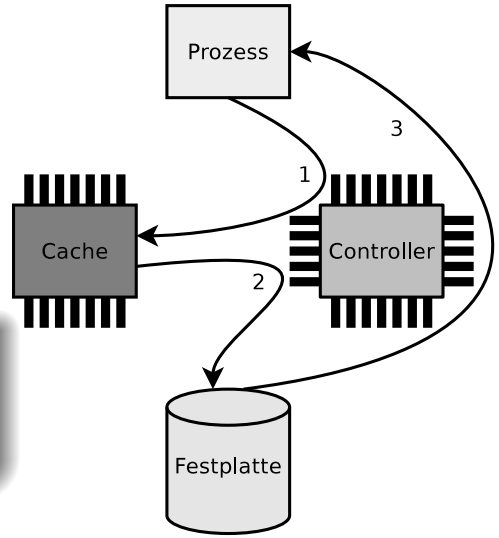


- Wird das Datenelement verändert, müssen die Änderungen irgendwann nach unten durchgereicht (zurückgeschrieben) werden
  - Beim Zurückschreiben, müssen die Kopien des Datenblocks auf allen Ebenen aktualisiert werden, um Inkonsistenzen zu vermeiden
  - Änderungen können nicht direkt auf die unterste Ebene (zum Original) durchgereicht werden!

Cache-Schreibstrategien

- **Write-Through**: Änderungen werden sofort an tiefere Speicherebenen weitergegeben
  - **Vorteil**: Konsistenz ist gesichert
  - **Nachteil**: Geringere Geschwindigkeit

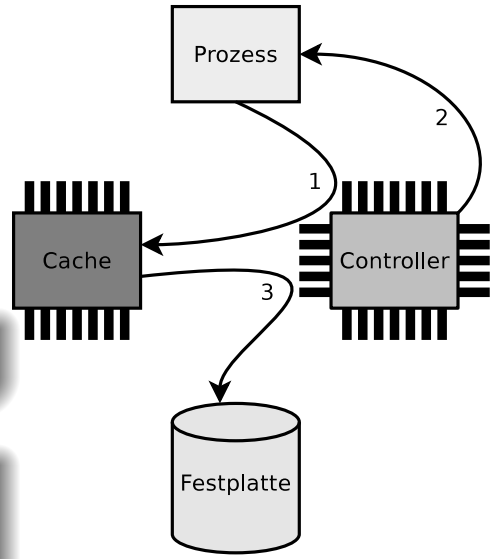
Bild: Ein Prozess möchte eine Schreibanweisung abarbeiten. Er schreibt (1) die Daten in den Cache und übergibt dem Controller die Schreibanweisung. Der Controller weist (2) das Schreiben der Daten auf dem Datenspeicher an. Wurden die Daten erfolgreich geschrieben, meldet (3) der Controller das erfolgreiche Schreiben der Daten an den Prozess



- **Write-Back**: Änderungen werden beim Verdrängen der Seite aus dem Cache weitergegeben
  - **Vorteil**: Höhere Geschwindigkeit
  - **Nachteil**: Änderungen sind beim Systemausfall verloren

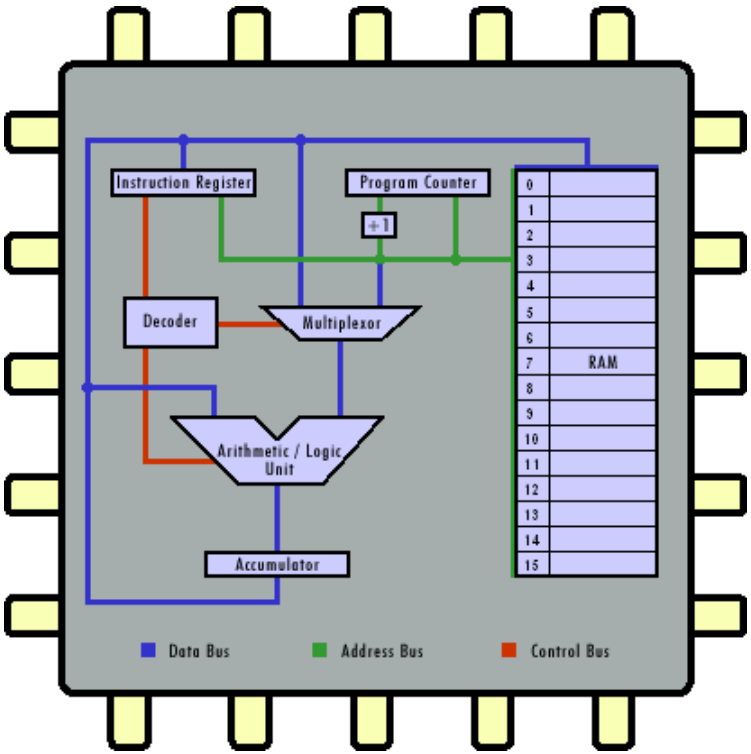
Für jede Seite im Cache wird ein **Dirty Bit** im Cache gespeichert, das angibt, ob die Seite geändert wurde

Bild: Ein Prozess möchte eine Schreibanweisung abarbeiten. Er schreibt (1) die Daten in den Cache und übergibt dem Controller die Schreibanweisung. Der Controller meldet (2) **sofort** das erfolgreiche Schreiben der Daten an den Prozess. Das Schreiben (3) der Daten auf dem Datenspeicher erfolgt asynchron zur Schreibanweisung im Prozess



Register, Cache und Hauptspeicher (1/4)

- **Register** enthalten die Daten, auf die die CPU sofort zugreifen kann
- Die Register sind genauso schnell getaktet wie die CPU selbst

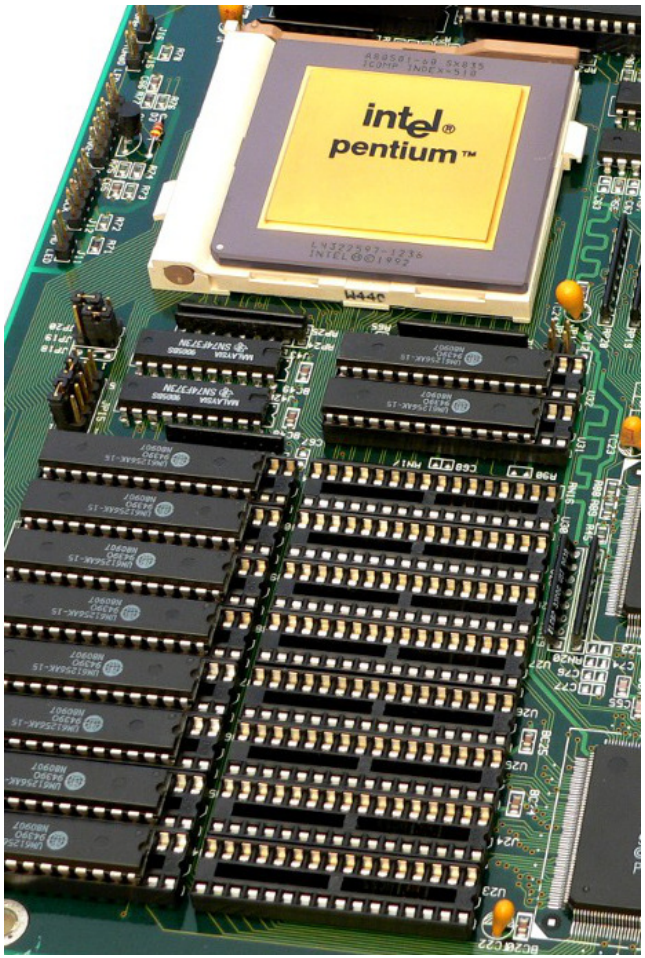


Bildquelle: `http://courses.cs.vt.edu/~csonline/MachineArchitecture/Lessons/CPU/cpu_circuit.gif`

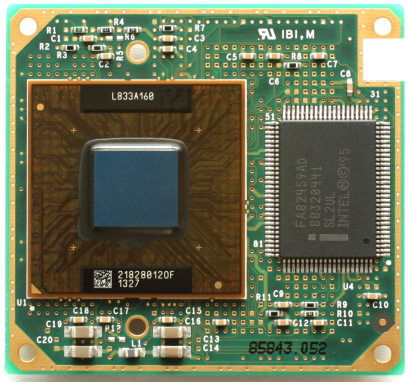
- **Datenregister** (= *Accumulatoren*) speichern Operanden für die ALU und deren Resultate
  - z.B. EAX, ECX, EDX, EBX (32-Bit)
    RAX, RBX, RCX, RDX (64-Bit)
    $\Longrightarrow$ Foliensatz 7
- **Adressregister** für Speicheradressen von Operanden und Befehlen
  - z.B. **Basisadressregister** (= *Segmentregister*) und **Indexregister** (für den Offset)
    $\Longrightarrow$ Foliensatz 5
- **Befehlszähler** (= *Program Counter*) (= *Instruction Pointer*) enthält die Speicheradresse des nächsten Befehls
- **Befehlsregister** (= *Instruction Register*) speichert den aktuellen Befehl
- **Stapelregister** (= *Stack Pointer*) enthält die Speicheradresse am Ende des Stack $\Longrightarrow$ Foliensatz 7

Register, Cache und Hauptspeicher (2/4)

- **Cache** (Pufferspeicher) enthält Kopien von Teilen des Arbeitsspeichers um den Zugriff auf diese Daten zu beschleunigen



- **First Level Cache** (L1-Cache)
  - In die CPU integriert
- **Second Level Cache** (L2-Cache)
  - Langsamer und größer
  - Ursprünglich außerhalb der CPU

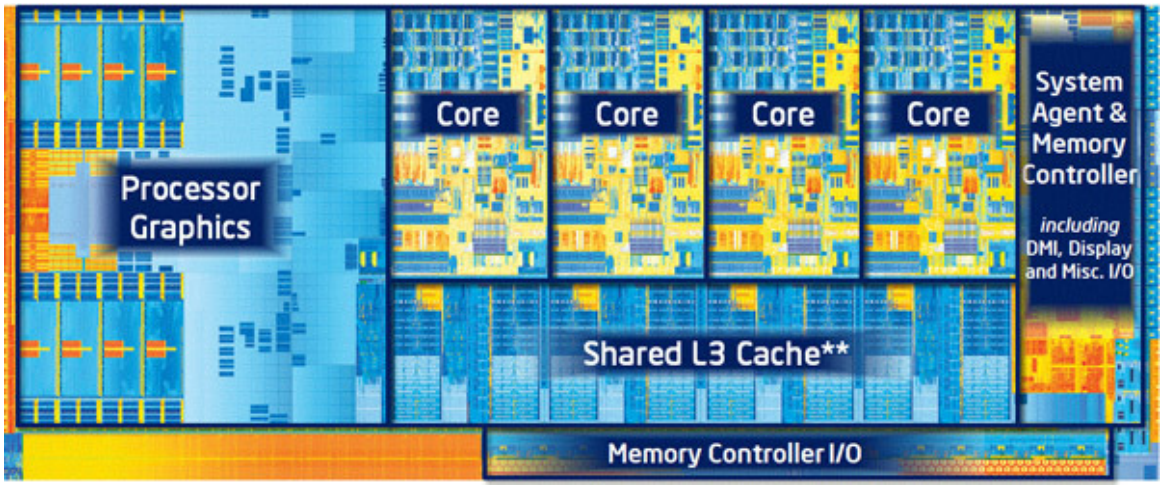


Bildquelle:
Wikipedia (Konstantin Lanzet CC-BY-SA-3.0)
Das Bild zeigt eine Intel Mobile Pentium II „Tongae“ 233 MHz CPU mit externem 512 kB L2-Cache. Der L2-Cache läuft mit halber Taktfrequenz

Bildquelle: `http://www.amoretro.de/2012/03/si5pi-aio-rev-1-1-socket-4-motherboard.html`
Das Bild zeigt ein Elitegroup SI5PI AIO mit einem Pentium 60.
Das Mainboard verfügt über 16 Sockel für Speichermodule für L2-Cache

Register, Cache und Hauptspeicher (3/4)

- Seit 1999/2000 integrieren die CPU-Hersteller zunehmend den L2-Cache in die CPUs
  - Das führte zur Etablierung das **Third Level Cache** (L3-Cache) als CPU-externen Cache
- Bei modernen CPUs (z.B. Intel Core-i-Serie und AMD Phenom II) ist auch der L3-Cache in die CPU integriert
  - Bei Multicore-CPUs mit integriertem L3-Cache teilen sich die Kerne den L3-Cache, während jeder Kern einen eigenen L1-Cache und L2-Cache hat



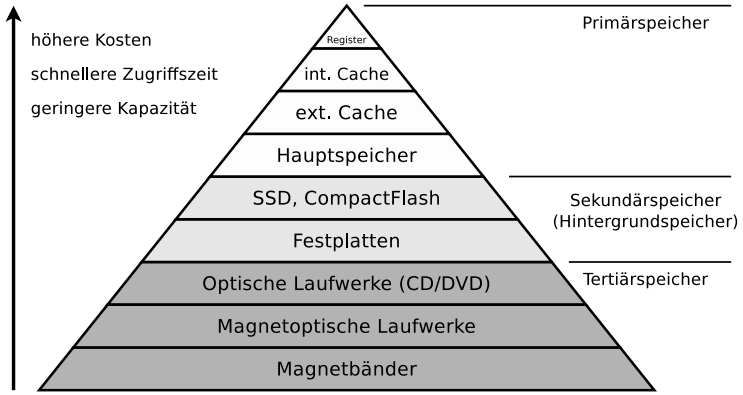
Bildquelle: Intel
Das Bild zeigt eine Intel Core i7-3770K „Ivy Bridge“ CPU mit 4 Kernen und integriertem L3-Cache

Einige Prozessor-Architekturen haben einen L4-Cache

- Intel Itanium 2 (2003): 64 MB
- Einige Intel Haswell CPUs (2013): 128 MB

Register, Cache und Hauptspeicher (4/4)

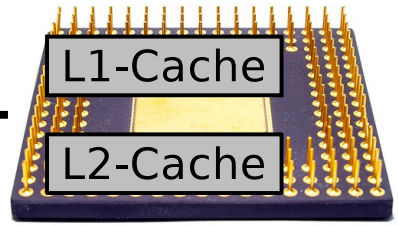
- Typische Kapazitäten der Cache-Ebenen:
  - L1-Cache: 4 kB bis 256 kB
  - L2-Cache: 256 kB bis 4 MB
  - L3-Cache: 1 MB bis 16 MB



- **Hauptspeicher**, auch **Arbeitsspeicher** oder **RAM** (*Random Access Memory* = Speicher mit wahlfreiem Zugriff) genannt
  - Kapazität: Wenige hundert MB bis mehrere GB
  - Alle Anfragen der CPU, die nicht vom Cache beantwortet werden können, werden an den Hauptspeicher weitergeleitet

Hauptspeicher (RAM)

L3-Cache



CPU

RAM und ROM

Im Gegensatz zum flüchtigen Hauptspeicher **RAM** ist **ROM** (*Read Only Memory*) ein nicht-flüchtiger Lesespeicher

Grundlagen zum Speicher und Speicherverwaltung

Wiederholung

- Bislang haben wir bzgl. Speicher geklärt:
  - Er nimmt Daten und die auszuführenden Programme auf
  - Er bildet eine Hierarchie ($\Longrightarrow$ Speicherpyramide)
    - Grund: Preis/Leistungsverhältnis
    - Je schneller ein Speicher ist, desto teurer und knapper ist er
  - Beim ersten Zugriff auf ein Datenelement, wird eine Kopie erzeugt, die entlang der Speicherhierarchie nach oben wandert
  - Da die obersten Speicherebenen praktisch immer voll belegt sind, müssen Daten verdrängt/ersetzt werden
  - Änderungen müssen durchgereicht (zurückgeschrieben) werden

- Es ist zu klären:
  - **Speicherverwaltung** $\Longrightarrow$ Foliensatz 5
  - **Speicheradressierung** $\Longrightarrow$ Foliensatz 5