4.Vorlesung Grundlagen der Informatik

Christian Baun

Hochschule Darmstadt
Fachbereich Informatik
christian.baun@h-da.de

3.11.2011

Wiederholung vom letzten Mal

- Generationen von Computersystemen
- Boolesche Algebra

Heute

- Rechnerarchitektur – Von-Neumann-Architektur
- Hardware-Komponenten eines Computers
- Speicher

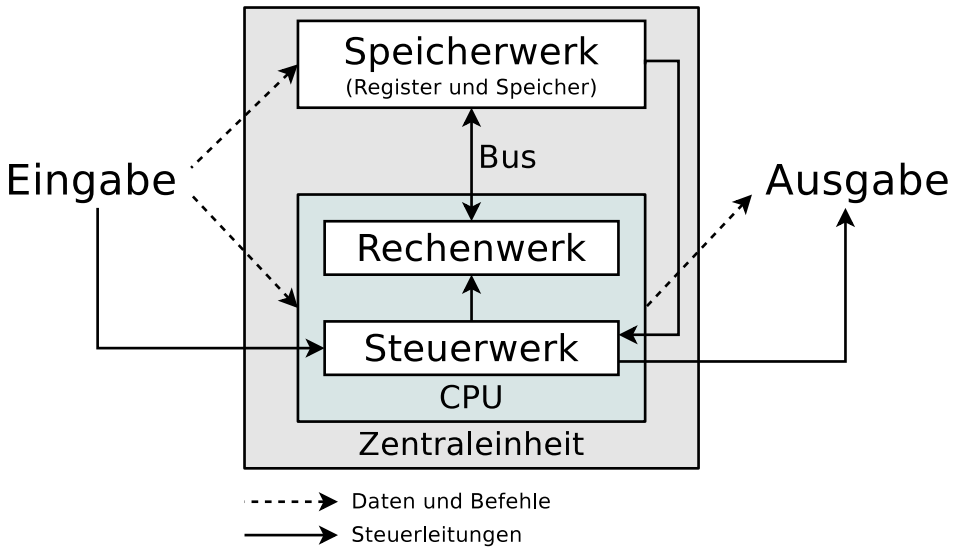
Von-Neumann-Architektur

- 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 sind:
  - Er hat sich als erster wissenschaftlich, mathematisch mit der Konstruktion von Rechenmaschinen beschäftigt
  - Urheberschaft am sequentiellen Prinzip (**Von-Neumann-Zyklus**)

Der Hauptprozessor – Die CPU

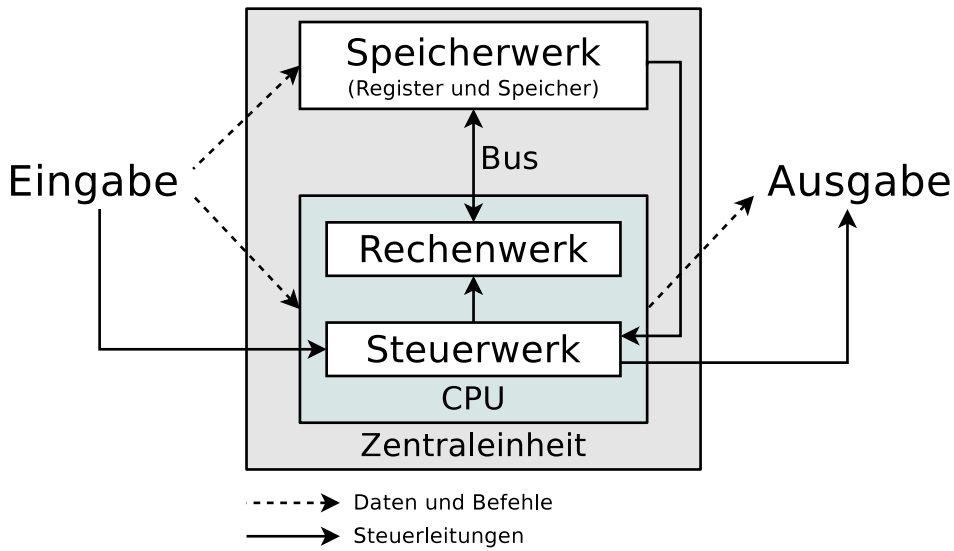
- Der Hauptprozessor ist das Herzstück moderner Computersysteme
  - Hauptprozessor = Central Processing Unit (CPU)
- Die meisten Komponenten eines Computers sind passiv und werden durch die CPU gesteuert
- Programme sind Folgen von Maschineninstruktionen, die in aufeinander folgenden Speicheradressen abgelegt sind

- Bei der Programmausführung setzt die CPU die Maschineninstruktionen Schritt für Schritt um
- Eine CPU besteht aus mindestens 3 Komponenten:
  - **Rechenwerk**
  - **Steuerwerk**
  - **Registersatz**



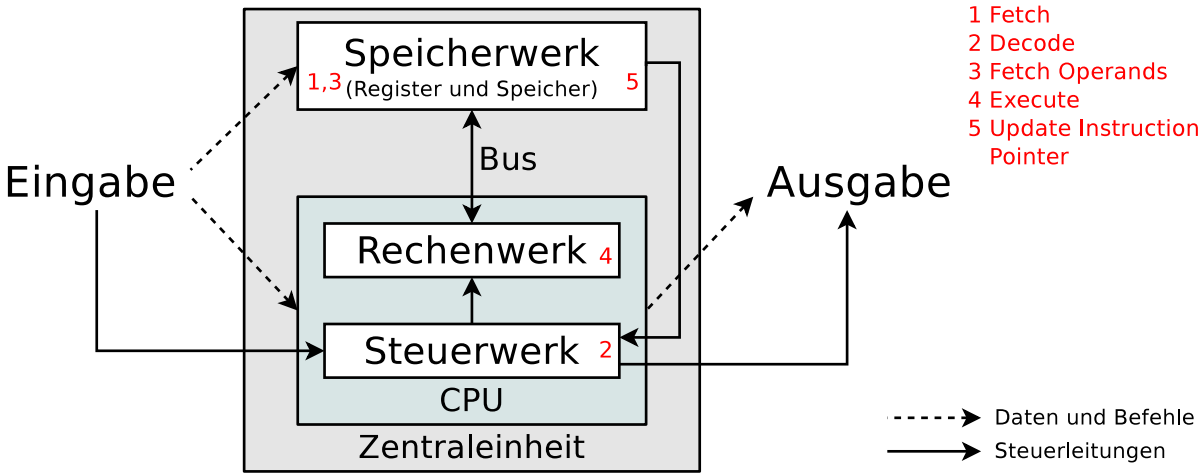
Komponenten der CPU

- **Rechenwerk** bzw. **Arithmetic Logic Unit** (ALU)
  - Manipulation von Daten und Adressen
  - Führt alle logischen und mathematischen Operationen aus
- **Steuerwerk** bzw. **Leitwerk** bzw. **Befehlswerk** (Control Unit)
  - Interpretiert Befehle, koordiniert der anderen CPU-Komponenten, steuert die Ein-/Ausgabe-Einheiten und den Steuerbus
  - Enthält das Befehlsregister (Instruction Table), das alle Befehle enthält, die die CPU ausführen kann
- **Registersatz** (Daten- und Spezialregister)
  - Speicherzellen (Register) für die kurzfristige Speicherung von Operanden und Adressen
  - Register arbeiten mir der selben Geschwindigkeit, wie der Rest der CPU



Von-Neumann-Zyklus des Von-Neumann-Rechners

- Sequentielle Arbeitsweise mit 5 Phasen (**Von-Neumann-Zyklus**)
  - Jede Phase kann mehrere Takte in Anspruch nehmen



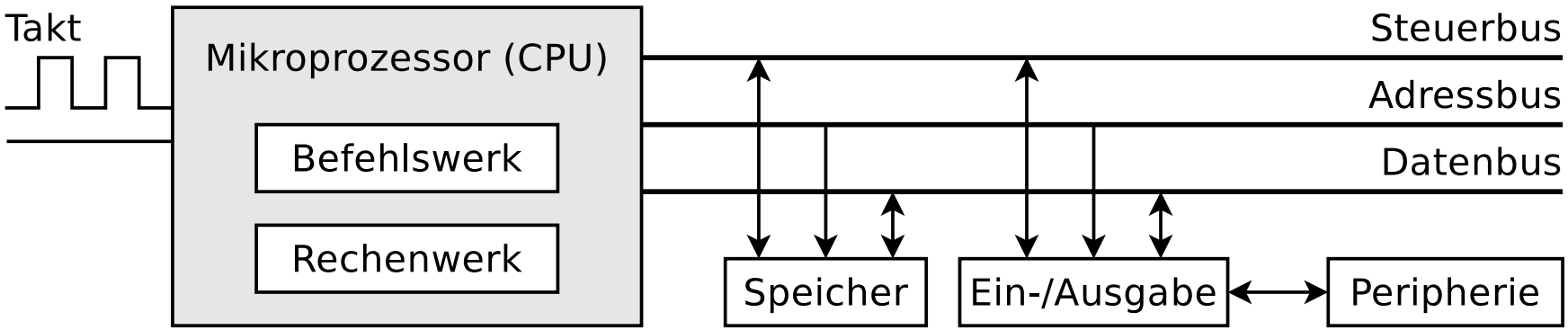
- Prozessor und Speicher kommunizieren über einen Bus direkt
- Befehle und Daten werden über diesen Bus transportiert

1. **FETCH**: Abzuarbeitenden Befehl aus dem Speicher in das Befehls-Register der CPU laden
2. **DECODE**: Steuerwerk löst den Befehl in Schaltinstruktionen für das Rechenwerk auf
3. **FETCH OPERANDS**: Parameter (Operanden) für den Befehl aus dem Speicher holen
4. **EXECUTE**: Rechenwerk führt die Operation aus
5. **UPDATE INSTRUCTION POINTER**: Befehlszähler wird erhöht. Zyklus beginnt von vorne und der nächste Befehl wird ausgeführt

Auch heute folgen die meisten gängigen Mikroprozessoren und Rechnersysteme dem Von-Neumann-Prinzip (Ausnahme: Bus)

Die Busleitungen (1)

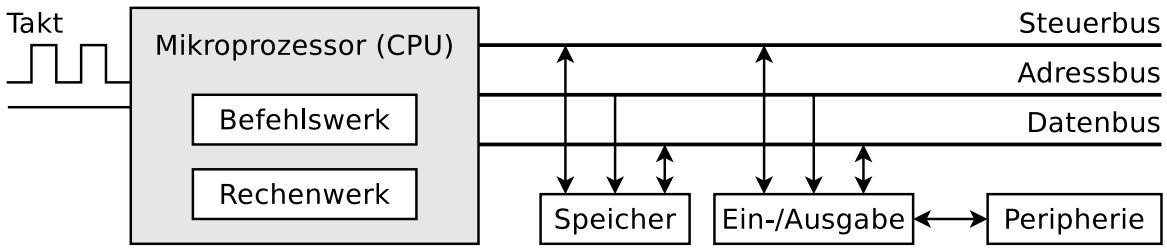
- Die Komponenten eines modernen Rechnersystems sind durch 3 digitale Busse verbunden: **Steuerbus**, **Adressbus** und **Datenbus**



- Steuerbus, Adressbus und Datenbus zusammen sind der **Systembus** oder **Fron Side Bus** (FSB)

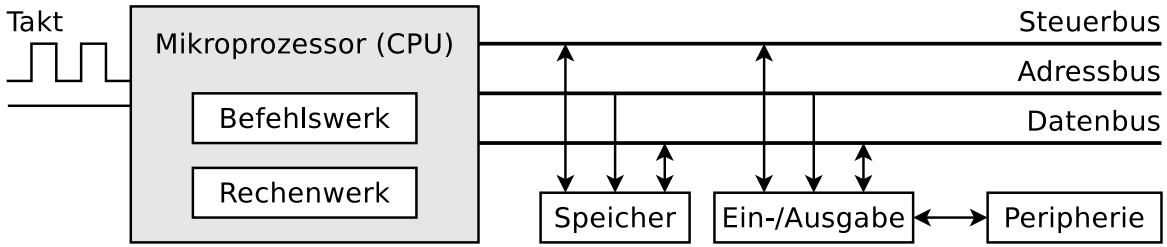
Die Busleitungen – Datenbus

- Über den bidirektionalen **Datenbus** werden Daten zwischen Prozessor, Arbeitsspeicher und Peripherie übertragen
- Anzahl der Datenbusleitungen legt fest, wie viele Bytes pro Takt übertragen werden können
- Üblicherweise ist die Bandbreite (Anzahl der Datenbusleitungen) des Datenbusses gleich der Größe der Arbeitsregister des Prozessors
- Moderne Prozessoren verfügen über eine Datenbusbreite von 64 Bit
  - Der Prozessor kann somit 64 Datenbits gleichzeitig (innerhalb eines Taktes) an und vom Arbeitsspeicher weg übertragen
- Die Datenfreigabe (Berechtigung zum Senden von Daten) erfolgt durch den Prozessor jeweils nur für eine Komponente



Die Busleitungen – Adressbus

- Der unidirektionale **Adressbus** ist nur für die Übertragung von Speicheradressen zuständig
- Es werden nicht nur die einzelnen Speicherzellen über den Adressbus angesprochen (adressiert), sondern auch Peripherie-Geräte
- Die Breite des Adressbusses (Anzahl der digitalen Signalleitungen) legt die maximale Anzahl der adressierbaren Speicherzellen fest



- Hat der Adressbus eine Busbreite von 32 Bit bedeutet das, das 2^{32} Speicherzellen, also ca. 4 Gigabyte Arbeitsspeicher adressierbar sind
  - Formel zur Ermittlung der maximal nutzbaren Speichergröße, die ein Prozessor ansprechen kann (in Byte):

$$\text{Maximal adressierbare Speicherplätze} = 2^{\text{Anzahl der Signalleitungen}}$$

Adressbus- und Datenbusbreite einiger Prozessoren

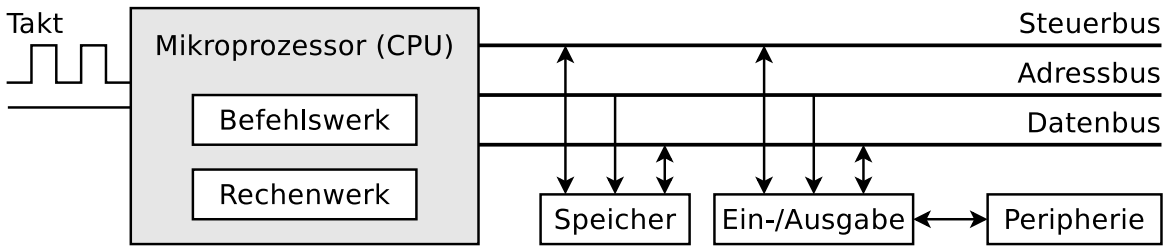
Prozessor	Adressbus	max. adressierbar	Datenbus
8088	20 Bit	$2^{20} = 1$ MB	8 Bit
8086 (XT)	20 Bit	$2^{20} = 1$ MB	16 Bit
80286 (AT)	24 Bit	$2^{24} = 16$ MB	16 Bit
80386-SX	32 Bit	$2^{32} = 4$ GB	16 Bit
80386-DX und 80468	32 Bit	$2^{32} = 4$ GB	32 Bit
Pentium und Pentium MMX	32 Bit	$2^{32} = 4$ GB	64 Bit
Pentium Pro	36 Bit	$2^{36} = 64$ GB	64 Bit
Pentium II, III und IV	32 Bit	$2^{32} = 4$ GB	64 Bit
Core 2 Duo, Core 2 Quad und Core i7	32 Bit	$2^{32} = 4$ GB	64 Bit
Itanium	44 Bit	$2^{44} = 17$ TB	64 Bit
Phenom-II (AMD)	48 Bit	$2^{48} = 281$ TB	64 Bit

- Bei einem Adressbus mit 64 Bit (z.B. bei Itanium 2 und AMD64) können 16 Etabyte adressiert werden

- Beim Pentium Pro können durch Physical Address Extension (PAE) mehr als 4 GB Arbeitsspeicher adressiert werden
- PAE ist eine Erweiterung in der Paging-Einheit
- Der pro Prozess nutzbare Arbeitsspeicher ist jedoch weiterhin auf 4 GB begrenzt

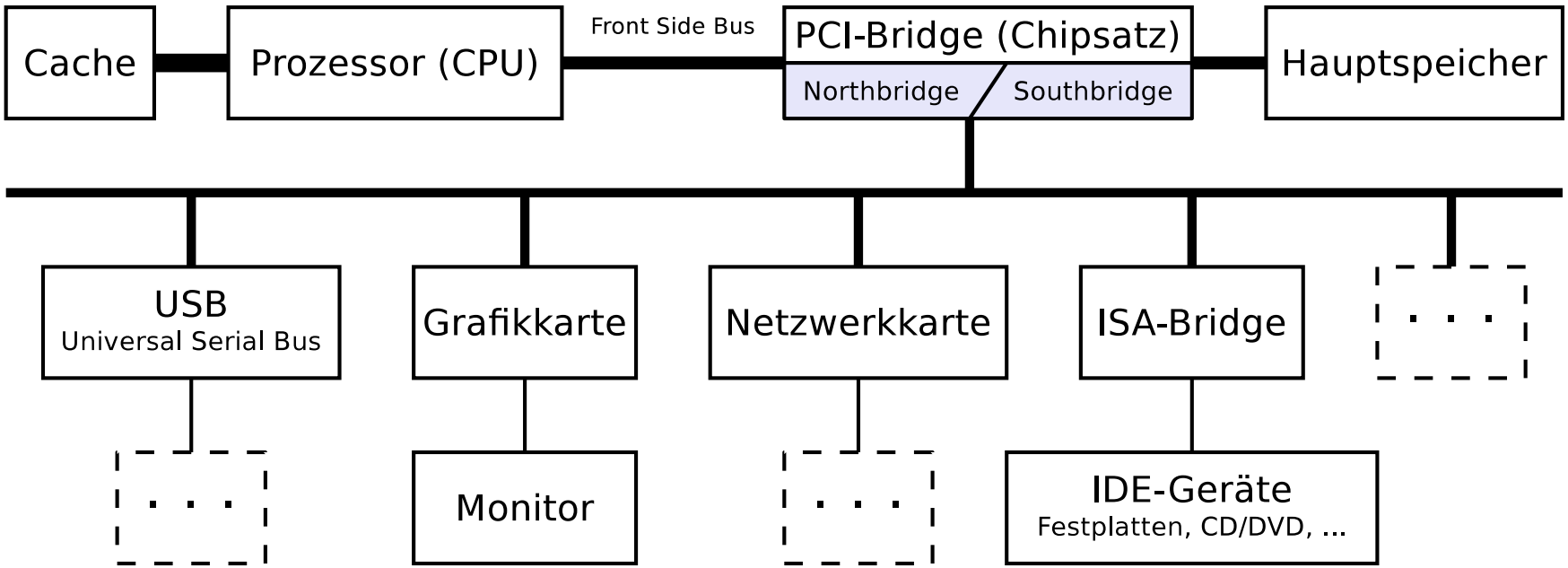
Die Busleitungen – Steuerbus

- Der bidirektionale **Steuerbus** koordiniert exklusive Lese- und Schreibanweisungen auf den Daten- und Adressbus und damit zwischen den Komponenten des Computersystems
- Eine über den Adressbus angesprochene Komponente, wird über den Steuerbus angewiesen, was sie zu tun hat
- Der Steuerbus beinhaltet auch die Interrupt-Leitungen
  - Über diese können Peripherie-Geräte dem Prozessor eine Unterbrechungsanforderung signalisieren
- Typische Bandbreiten beim Steuerbus: 5-10 Leitungen



Busse in modernen Rechersystemen

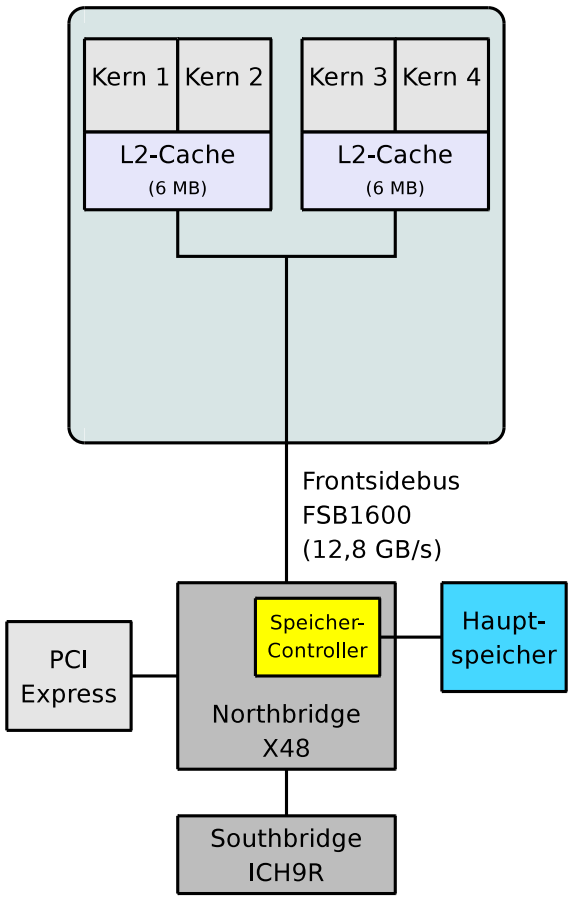
- Die verbindende Element ist die PCI-Bridge (der sog. **Chipsatz**)
  - Besteht aus 2 Komponenten:
    - **Northbridge** liegt dicht an der CPU, um Daten schnell übertragen zu können. Zuständig für Speicherzugriffe und Grafikkarte
    - **Southbridge** ist für langsamere Verbindungen wie PCI, ISA, SATA, USB, Firewire, usw. zuständig



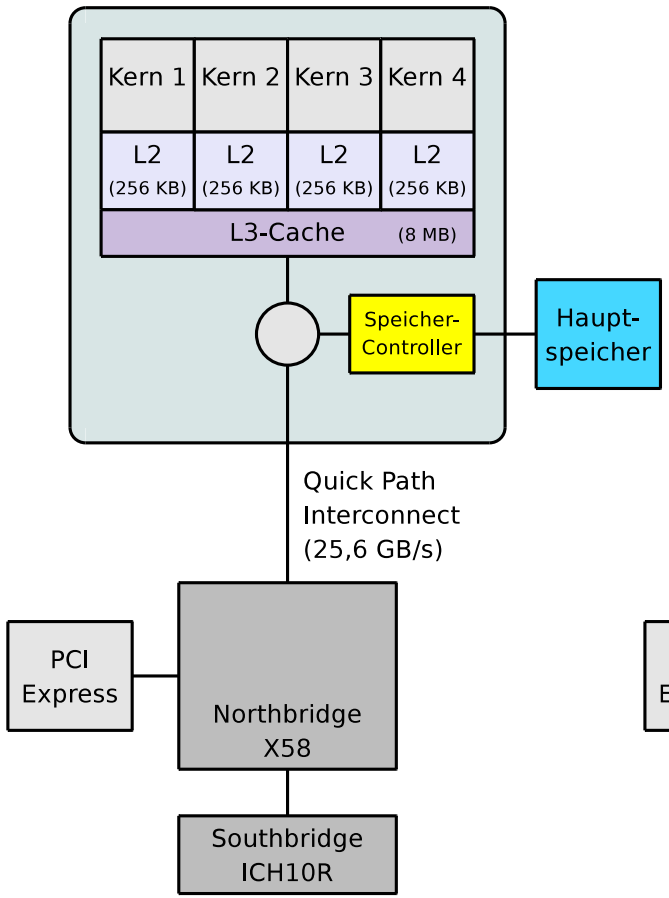
Die Busleitungen (2)

- **Front-Side-Bus** (FSB) ist der Bus zwischen CPU und Chipsatz und enthält Adressbus, Datenbus und Steuerbus
- Aus Geschwindigkeitsgründen verlagert man immer mehr Bestandteile des Chipsatzes in den Prozessor
  - In modernen Architekturen gibt es den klassischen Systembus nicht mehr
  - Geräte werden wegen der unterschiedlichen technologischen Realisierungen der Geräte nicht mehr direkt mit dem Prozessor verbunden
  - Rechnersysteme enthalten heute verschiedenen seriellen und parallele Bussysteme, die für die jeweilige Erfordernisse ausgelegt sind
  - Es werden auch immer häufiger Punkt-zu-Punkt-Verbindungen eingesetzt
  - Eingabe-/Ausgabecontroller treten als Vermittler zwischen den Geräten und dem Prozessor auf
- Einige ausgewählte Bussysteme:
  - **Parallele, Rechner-interne Busse**: PATA (IDE), PCI, ISA, SCSI,...
  - **Serielle, Rechner-interne Busse**: SATA, PCI-Express...
  - **Parallele, Rechner-externe Busse**: PCMCIA, SCSI...
  - **Serielle, Rechner-externe Busse**: Ethernet, FireWire, USB, eSATA...

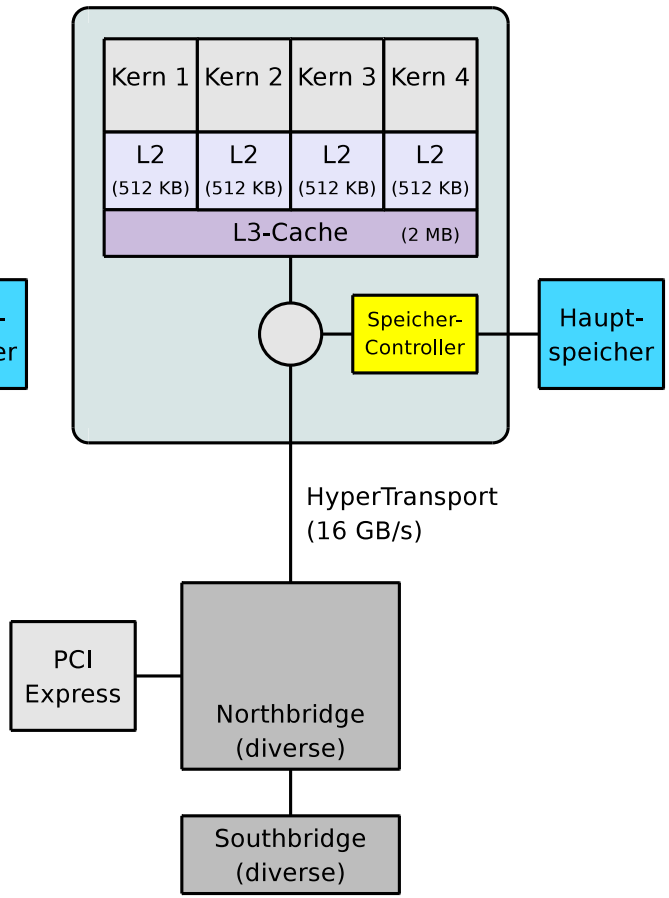
Verlagerung des Speichercontrollers in die CPU



Intel Core 2 Extreme QX9770



Intel Core i7-965 Extreme Edition



AMD Phenom X4 9950

Quelle: c't 25/2008

Speicher

- Speicher nimmt Daten und die auszuführenden Programme auf
- Speicher bildet durch Busse verbunden eine Hierarchie
  $\Longrightarrow$ **Speicherpyramide**
- Grund für Speicher-Hierarchie: Preis/Leistungsverhältnis
  $\Longrightarrow$ Je schneller ein Speicher ist, desto teurer und knapper ist er

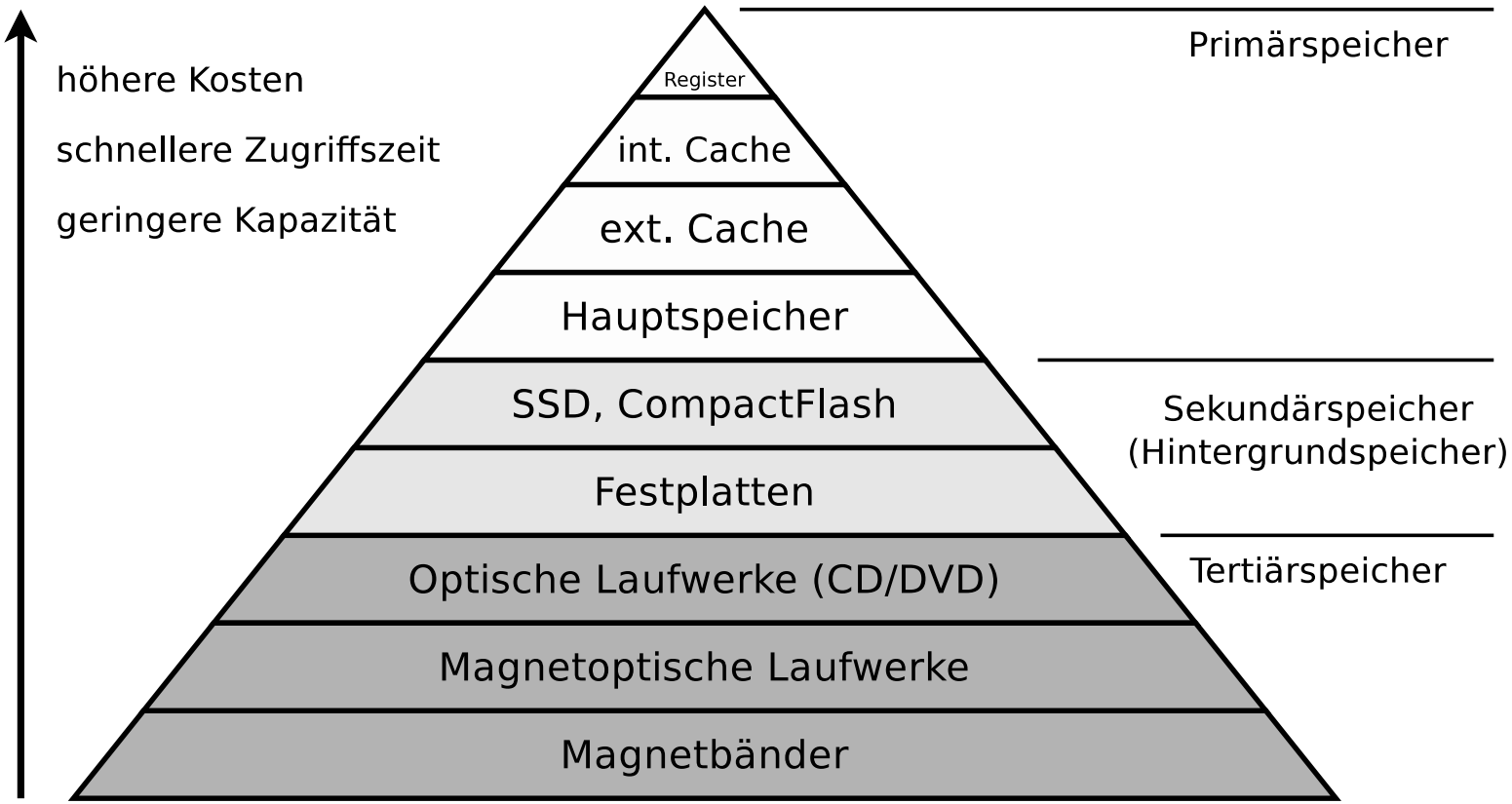
Welche digitalen Datenspeicher kennen Sie?

Digitale Datenspeicher

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

- Wahlfreier Zugriff bedeutet, 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

Die Speicherpyramide



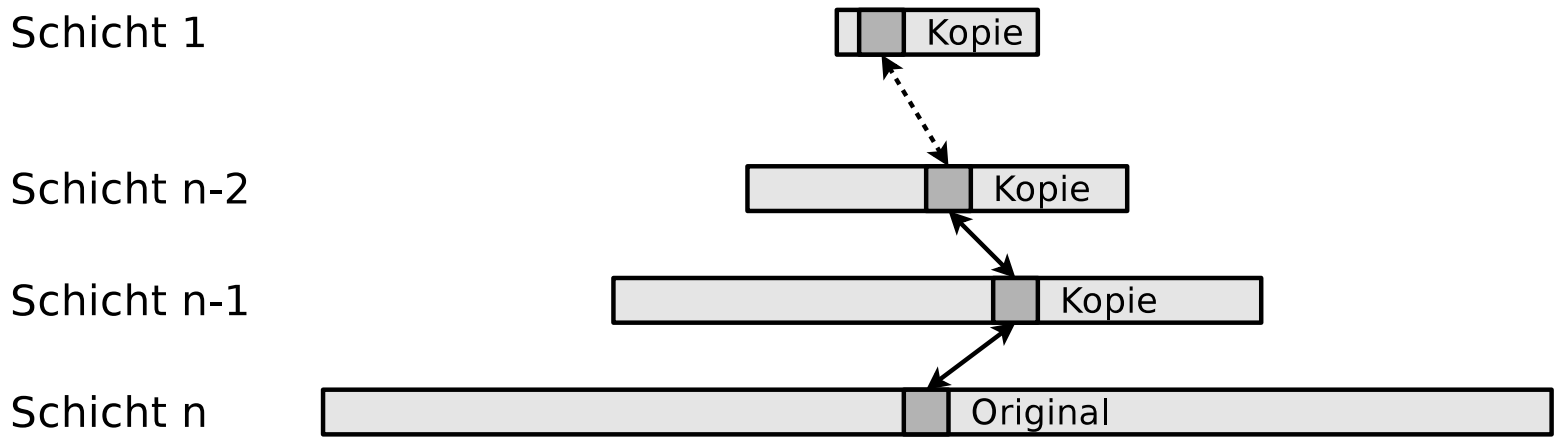
- **Primärspeicher**: Darauf kann der Prozessor direkt zugreifen
- **Sekundärspeicher**: Hintergrundspeicher, der über einen Controller angesprochen wird
- **Tertiärspeicher**: Nicht dauerhaft verfügbar, oder über ein Laufwerk mit dem Rechner verbunden. Hauptaufgabe ist Archivierung

Primär-/Sekundär-/Tertiärspeicher

- Primärspeicher und Sekundärspeicher sind **Onlinespeicher**, da sie eine feste Verbindung zum Computer und dadurch geringe Zugriffszeiten auf die Daten haben
- 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 (Sekundärspeicher) 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

- Für Schreibzugriffe auf den Cache gibt es 2 Möglichkeiten

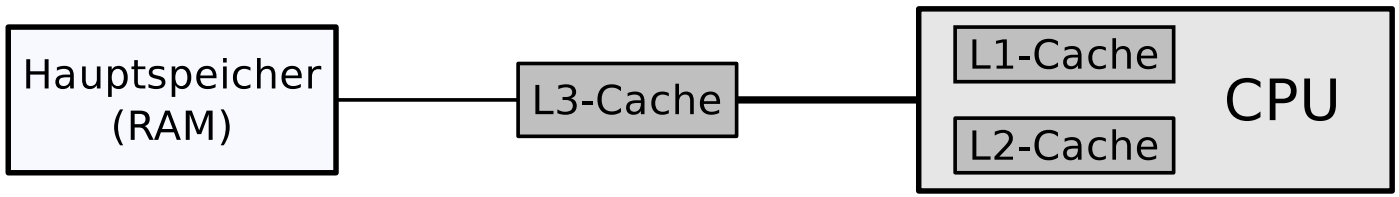
1. **Write-Back**
   - Schreibzugriffe werden nicht direkt an die tieferen Speicherebene weitergegeben
   - Dadurch kommt es zu Inkonsistenzen zwischen den Daten im Cache und auf dem zu cachenden Speicher
   - Die Daten werden erst zurückgeschrieben, wenn der betreffende Datenblock aus dem Cache verdrängt wird $\Longrightarrow$ **Zurückschreiben**
   - **Vorteil**: Höhere Systemgeschwindigkeit
   - **Nachteil**: Daten können beim Systemausfall verloren gehen
2. **Write-Through**
   - Schreibzugriffe werden sofort an die tieferen Speicherebenen weitergegeben $\Longrightarrow$ **Durchschreiben**
   - **Vorteil**: Datenkonsistenz ist gesichert
   - **Nachteil**: Geringere Systemgeschwindigkeit

Register, Cache und Hauptspeicher (1)

- **Register** in der CPU enthalten die Daten, auf die die CPU sofort zugreifen kann und sind genauso schnell getaktet wie die CPU selbst
  - Kapazität der Register: In der Regel $\leq 1\,\mathrm{KB}$
  - Die Entscheidung, was in den Registern gespeichert wird, fällen die laufenden Programme
- **Cache** (Pufferspeicher) enthält Kopien von Teilen des Arbeitsspeichers um den Zugriff auf diese Daten zu beschleunigen
  - Der Cache ist oft in 2 oder 3 Ebenen verteilt
    - Erste Ebene (**First Level Cache**) ist der CPU am nächsten platziert und in die CPU-Architektur integriert
    - Zweite Ebene (**Second Level Cache**) ist langsamer und größer und befindet sich außerhalb der CPU (auf dem Mainboard)
    - Die Entwicklung geht dahin, dass immer häufiger der Second Level Cache in die CPU-Architektur integriert wird, was zur Umbenennung des externen Cache in **Third Level Cache** führte

Register, Cache und Hauptspeicher (2)

- Typische Kapazitäten der Cache-Ebenen:
  - First Level Cache: 4 KB bis 256 KB
  - Second Level Cache: 256 KB bis 4 MB
  - Third Level Cache: 1 MB bis 16 MB
- Der **Hauptspeicher**, auch **Arbeitsspeicher** oder **RAM** (*Random Access Memory* = Speicher mit wahlfreiem Zugriff) hat aktuell Größen von wenigen hundert MB bis mehreren GB
  - Alle Anfragen der CPU, die nicht vom Cache beantwortet werden können, werden an den Hauptspeicher weitergeleitet



RAM und ROM

Im Gegensatz zum flüchtigen Lese- und Schreibspeicher **RAM** ist ein **ROM** (*Read Only Memory*) ein nicht-flüchtiger Lesespeicher. Ein solcher Speicher kann nur ausgelesen, aber nicht beschrieben werden

Lokalitätsausnutzung

- Cache ist schneller, teurer und knapper Speicher
  - Cache kann nie alle Daten gleichzeitig vorrätig gespeichert haben
- Die Entscheidung, welche Daten im Cache gehalten werden, ist von den Lokalitätseigenschaften der Zugriffe abhängig:
  - **Zeitliche Lokalität**: Bei Schleifen z.B. ist es wahrscheinlich, dass sich Zugriffe auf Daten mehrmals wiederholen. Diese Daten sollten also bevorzugt im Cache gehalten werden. Ältere Daten müssen aus Platzgründen aus dem Cache verdrängt werden
    $\Longrightarrow$ **Verdrängung**
  - **Räumliche Lokalität**: Wahrscheinlichkeit, dass Daten in benachbarten Adressbereichen zusammengehören ist hoch. Wegen der räumlichen Lokalität speichert man bei Caches nicht einzelne Bytes sondern die Daten ganzer Adressbereiche
    $\Longrightarrow$ **Cache-Block** (starker Einfluss auf die Systemleistung!)

Cache-Hit und Cache-Miss

- Bei einer Daten-Anfrage an den Cache sind 2 Ergebnisse möglich
  - **Cache-Hit**: Angefragte Daten sind vorhanden (Treffer)
  - **Cache-Miss**: Angefragte Daten sind nicht vorhanden (verfehlt)
- 2 Kennzahlen bewerten die Effizienz eines Cache
  - **Hitrate**: Anzahl der Anfragen an den Cache mit Ergebnis Cache-Hit, geteilt durch die Gesamtanzahl der Anfragen
    - Ergebnis liegt zwischen Null und Eins
    - Je höher der Wert, desto höher ist die Effizienz des Caches
  - **Missrate**: Anzahl der Anfragen an den Cache mit Ergebnis Cache-Miss, geteilt durch die Gesamtanzahl der Anfragen
    - Missrate $= 1 -$ Hitrate

Grundlagen zum Speicher und Speicherverwaltung

- Bislang haben wir bzgl. Speicher geklärt:
  - Speicher nimmt Daten und die auszuführenden Programme auf
  - Speicher bildet eine Hierarchie ($\Longrightarrow$ Speicherpyramide)
  - Grund für die Speicher-Hierarchie: 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 werden, um Platz zu schaffen
- Wird ein Datenelement im Speicher verändert, müssen die Änderungen nach unten durchgereicht (zurückgeschrieben) werden
- Es ist zu klären:
  - Wie wird der Speicher angesprochen?
  - Wie funktioniert Speicherverwaltung (Scheduling und Swapping)?

Nächste Vorlesung

Nächste Vorlesung:

10.11.2011