

Betriebssysteme: Einführung in die Rechnerarchitektur

WS 2016/17

Michael Jäger

11. Oktober 2016



Von-Neumann-Architektur

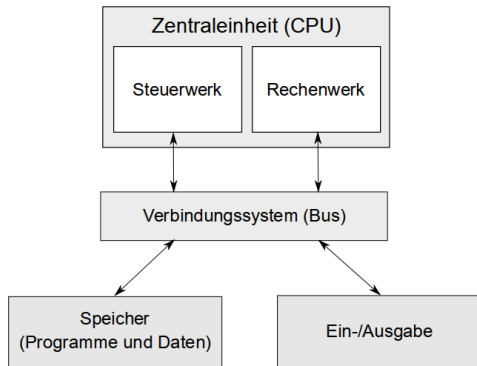
(Speicherprogrammierter Rechner)

John von Neumann

- Ungarischer Mathematiker, 1903-1957
- Begründer der modernen Spieltheorie
- Pionier der Monte-Carlo-Simulation
- Mitentwickler der Wasserstoffbombe
- Mitbegründer der modernen Rechnerarchitektur



Architekturmodell des speicherprogrammierten Rechners



Bussysteme

Bus

- mehrere parallele Leitungen („Busbreite“ = Anzahl der Leitungen)
- mehrere Funktionseinheiten angeschlossen
- Informationsaustausch zeitmultiplex: Zu jedem Zeitpunkt sind immer nur zwei Einheiten (Sender, Empfänger) miteinander verbunden.
- unidirektional oder bidirektional

Interne und externe Busse

Unterschiedliche Busse sind hierarchisch miteinander verknüpft

- CPU-internes Bussystem verbindet CPU-Komponenten
- Systembus verbindet CPU, Hauptspeicher und E/A-Komponenten
- weitere Busse zum Anschluss jeweils mehrerer Bus-kompatibler Geräte, z.B. USB (Universal Serial Bus), PCI (Peripheral Component Interconnect)

Bustypen

- Adress-Bus (A-Bus)
 - unidirektional
 - zur Auswahl von Speicherzellen, Bausteinen, E/A-Registern
- Daten-Bus (D-Bus)
 - bidirektional
 - 8-64 Leitungen
 - zum Datentransport zwischen Bausteinen
- Steuer-Bus (Control-Bus, S-Bus)
 - Unterschiedliche Übertragungsrichtungen, jede Leitung für sich unidirektional
 - typisch: 4-20 Leitungen
 - zur Steuerung der Zusammenarbeit der einzelnen Baugruppen

Bus-Treiber

Problem bei Bus-Systemen

Zu jedem Zeitpunkt darf immer nur ein einziger Sender am Bus aktiviert sein

Aufgaben des Bus-Treibers

- An- und Abschalten der angeschlossenen Bausteine
- Durchschalten der gewünschten Übertragungsrichtung bei bidirektionalen Busanschlüssen
- Ziel: Hohe Anschlußkapazität („Fan-Out“)

ALU – Arithmetisch-logische Einheit

Typische Merkmale

- Operationen:
 - Binäre Ganzzahl-Arithmetik
 - Logische Operationen (UND, ODER, NOT, XOR)
 - Verschiebeoperationen (arithmetisch, logisch)
- Zwei Eingabebusse für Operanden
- Ein-/Ausgabebus für Ergebnis
- Steuereingang vom Leitwerk
- Statusausgang

Sonstige Recheneinheiten

Moderne Prozessoren haben oft neben der ALU weitere „Recheneinheiten“ für spezielle Zwecke, z.B.

- Grafikberechnungen
- Gleitkomma-Arithmetik
- Kryptographische Berechnungen
- Hauptspeichermanagement
 - MMU – Memory Management Unit)
 - Cache/Snoop - Unit

Maschinenbefehle

- Compiler erzeugt Maschinenbefehle aus den Anweisungen einer höheren Programmiersprache
- Ausführung:
 - Maschinenbefehle des Programms im Hauptspeicher
 - Prozessor kopiert jeweils einen Befehl in das Befehlsregister
 - Prozessor dekodiert den Befehl und führt ihn aus
- Befehls-Bestandteile:
 - Operationscode („OpCode“) identifiziert den Maschinenbefehl
 - Operanden – abhängig von Befehl
Beispiele: Registernummern, Speicheradressen, Direktoperanden
- Verschiedene Befehlsformate
- Befehle ggf. unterschiedlich lang

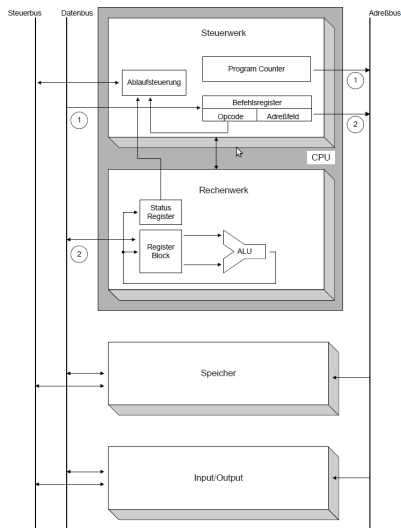
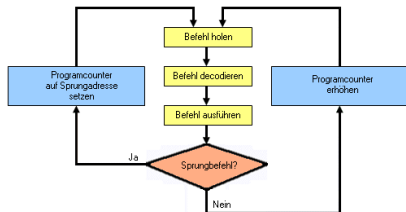
Befehlstypen

- Transferbefehle
Kopieren von Daten aus Speicherzellen oder Register in andere Speicherzellen oder Register
- Flagbefehle
Manipulation der Flags im Statusregister, z.B. Carry-/Interrupt
Vorbelegungen für Addition/Subtraktion
Programmteile vor Unterbrechungen schützen
- Arithmetische/logische Instruktionen
Addition, Subtraktion, Multiplikation und Division
AND, OR und XOR
- Rotations- und Shiftbefehle
- Kontrollflussbefehle
unbedingten Sprungbefehle, bedingte Branchbefehle (Verzweigungen),
Unterprogrammaufruf, Rücksprunginstruktionen
- Weitere Befehle
NOP, STOP

Mikroprogrammierung

- Maschinenbefehle müssen nicht immer in Hardware realisiert sein
- Mikroprogrammierung:
 - Ein komplexer Maschinenbefehl wird als Sequenz primitiver Maschinenbefehle („Mikroprogramm“) implementiert
 - Mikroprogramme sind im Prozessor gespeichert
 - Ausführung des komplexen Befehls durch Abarbeitung des Mikroprogramms

Ablauf der Befehlsausführung



Typischer Ablauf der Befehlsausführung

FETCH-Phase

- Befehl, auf den der Programmzähler verweist, aus dem Hauptspeicher in das Befehlsregister kopieren

DECODE-Phase

- Befehl dekodieren, Opcode und Operanden validieren, bei illegalem Befehl: Ausnahmestand
- Programmzähler erhöhen

FETCH-OPERANDS-Phase

bei lesenden Hauptspeicherzugriffen

- Adressen der Speicheroperanden berechnen
- Speicher-Operanden in den Prozessor laden

EXECUTE-Phase

- Befehl ausführen (Ausführung einer Rechenoperation durch ALU)
- Ggf. Ausnahmebehandlung (Division durch 0, Überlauf, ...)
- Ergebnis in das Zielregister speichern
- Bei Sprungbefehl Programmzähler aktualisieren

WRITEBACK-Phase

bei schreibenden Hauptspeicherzugriffen

- Adressen der Speicheroperanden berechnen
- Wert in den Hauptspeicher kopieren

Zahlen zum technischen Fortschritt

Anzahl der Transistoren

Intel 8080 Mikroprozessor	4 500
Intel 80286 Mikroprozessor	134 000
Intel 80386 Mikroprozessor	275 000
Intel Pentium Mikroprozessor	3 100 000
Intel Pentium 4 Mikroprozessor	42 000 000
AMD KG (Athlon 64) Mikroprozessor	105 900 000
Intel Core i7 Mikroprozessor	731 000 000
Eight Core Xeon Nehalem-EX Mikroprozessor	2 300 000 000
RV820 ATI/AMD Grafikprozessor	2 154 000 000
NVIDIA GTX Titan X Grafikprozessor	8 000 000 000

Zahlen zum technischen Fortschritt

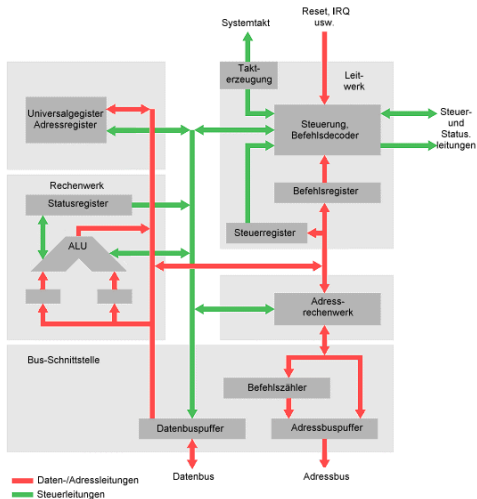
Festplattenkosten pro Gigabyte (in US-Dollar)

1990	53 000
1995	850
2000	20
2005	1
2010	0.10
2015	0.04

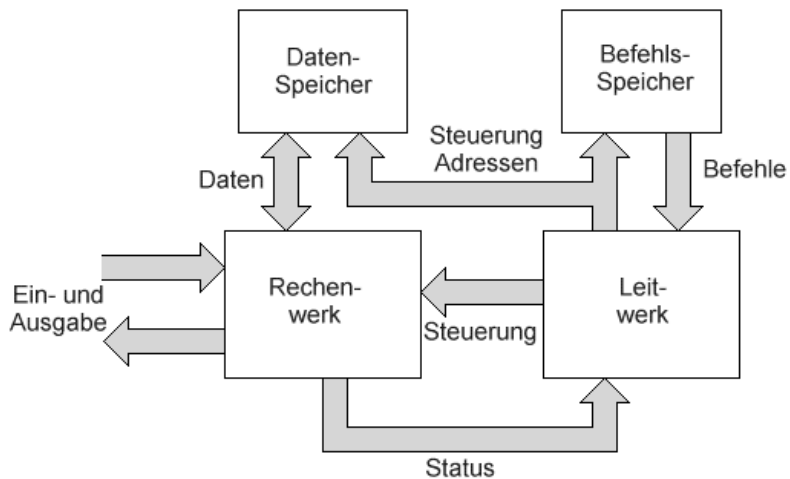
RAM-Kosten pro Gigabyte (in US-Dollar)

1990	120 000
1995	33 000
2000	1 400
2005	190
2010	20
2015	10

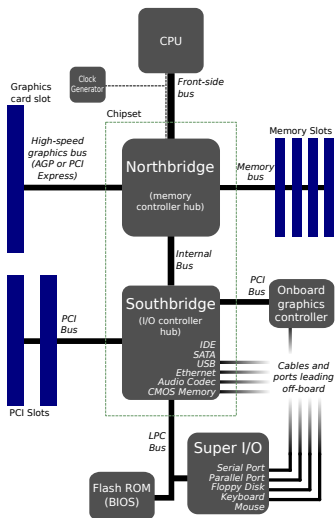
Von-Neumann-Rechner



Harvard-Architektur



PC-Architektur mit externem Speichercontroller



Cache

Begriffsdefinition

Ein Cache ist ein schnell zugreifbarer Speicher, in dem Kopien von Daten aus einem langsameren größeren Speicher gehalten werden. Wenn häufig benötigte Daten im Cache stehen, kann die Zugriffsgeschwindigkeit gesteigert werden.

Anwendungsbeispiele

- Ein Prozessor lädt vom Hauptspeicher gleich mehrere hintereinander gespeicherte Befehle in den Befehls-cache (PREFETCH). Wenn keine Sprungbefehle dabei sind, spart man FETCH-Operationen.
- Ein Prozessor speichert die zuletzt zugewiesenen Daten in einem Datencache. Eine Geschwindigkeitssteigerung ergibt sich, wenn Daten mehrfach zugewiesen werden.

Performance

- Die Trefferrate („hit rate“) gibt an zu welchem Prozentsatz Datenzugriffe aus dem Cache befriedigt werden können.
- Ein Lesezugriff auf nicht im Cache stehende Daten („read miss“) verzögert die Ausführung. Ein entsprechender Schreibzugriff („write miss“) führt nicht unbedingt zu einer Verzögerung.

Cache-Einträge

bestehen aus

- Datenblock: Kopie der Daten/Befehle, z.B. 64 Byte große „cache lines“
- Etikett („tag“): Anfangsadresse der Originaldaten im Hauptspeicher
- Flags: verschiedene Bits zur Steuerung der Auslagerungs- und Schreibstrategien. Bsp: Ein „valid“-Bit ist gesetzt, wenn der Cacheblock gültige Daten enthält. Bei „write back“-Caches gibt es ein „dirty“-Bit, das bei Modifikation gesetzt wird

Auslagerungsstrategie

- Wird bei vollem Cache ein weiterer Cache-Platz benötigt, muss ein anderer Platz freigeräumt werden.
- Häufig wird der Platz, der am längsten nicht mehr zugegriffen wurde, freigeräumt (LRU-Strategie, „least recently used“).

Cache-Hierarchie

Oft sind mehrere Caches in einer Hierarchie angeordnet, z.B.

- Level 1: kleiner, schneller Cache für Daten und Befehle nahe beim Prozessor(-Kern). Größenbeispiel: 64Kb
- Level 2: großer Cache für Daten, um Hauptspeichierzugriffe einzusparen. Größenbeispiel: 1Mb
- Level 3: Cache, der von mehreren Prozessorkernen gemeinsam benutzt wird. Ein Ziel ist dabei der schnelle Datenaustausch zwischen den Prozessorkernen über den Cache.

Schreibstrategie

- Ein „write through“-Cache synchronisiert bei jeder Modifikation den Inhalt sofort mit dem Hauptspeicher
- Ein „write back“-Cache verzögert das Zurückschreiben modifizierter Daten. Wird ein modifizierter Platz freigeräumt („dirty bit“ gesetzt), muss dessen Inhalt zunächst in den Hauptspeicher zurück geschrieben werden. Dies kann sowohl beim Lesen als auch beim Schreiben von noch nicht im Cache vorhandenen Daten passieren.
- Mischstrategien verzögern das Durchschreiben so lange, bis mehrere Cache-Lines zurückgeschrieben werden müssen. Dadurch spart man Busanforderungen.