

# Grundlagen der Informatik

- Computer-Hardware Forts.-

Prof. Dr. Klaus Volbert



Hochschule für angewandte Wissenschaften  
Fakultät Informatik und Mathematik

Wintersemester 2010/11  
Regensburg, 03./04. November 2010

# Erstes Rechenmodell: Registermaschine

- Registermaschine (engl. Random Access Machine, RAM)
  - Zentrale Recheneinheit inkl. Rechen- und Steuerwerk
    - Akkumulator (Adresse 0 im Datenspeicher)
    - Befehlsregister
    - Befehlszähler (Programm Counter, PC)
  - Programmspeicher (Program Memory, PM, Adressen 1, 2, ...)
  - Datenspeicher (Data Memory, DM, Adressen 1, 2, ...)
  - Vereinfachte Ein-/Ausgabe durch direkte Befehle
- Inhalt des Datenspeichers sei beschrieben durch  $f: \text{IN} \rightarrow \text{IN}$ 
  - $f(0)$  ist der **Inhalt des Akkumulators**
  - $f(\text{adresse})$  ist der **Inhalt des Datenspeichers** an der Stelle *adresse*
- Befehlsstruktur
  - $\langle \text{Adresse PM} \rangle \langle \text{Befehl} \rangle \langle \text{Adresse PM/DM} \rangle$ 
    - 8 Bit
    - 8 Bit

# Befehle der Registermaschine I

- Verarbeitungs-, Transport- und Ein-/Ausgabebefehle:

Codierung	Syntax	Semantik	Beschreibung
0x01	ADD <i>adresse</i>	$f(0) = f(0) + f(\textit{adresse})$	Addieren
0x02	SUB <i>adresse</i>	$f(0) = f(0) - f(\textit{adresse})$	Subtrahieren
0x03	MUL <i>adresse</i>	$f(0) = f(0) * f(\textit{adresse})$	Multiplizieren
0x04	DIV <i>adresse</i>	$f(0) = f(0) / f(\textit{adresse})$	Dividieren
0x05	LDA <i>adresse</i>	$f(0) = f(\textit{adresse})$	Laden
0x06	LDK <i>zahl</i>	$f(0) = \textit{zahl}$	Konstante Laden
0x07	STA <i>adresse</i>	$f(\textit{adresse}) = f(0)$	Speichern
0x08	INP <i>adresse</i>	$f(\textit{adresse}) = \langle \text{Eingabe} \rangle$	Eingeben
0x09	OUT <i>adresse</i>	$\langle \text{Ausgabe} \rangle = f(\textit{adresse})$	Ausgeben
0x0A	HLT 99		Programmende

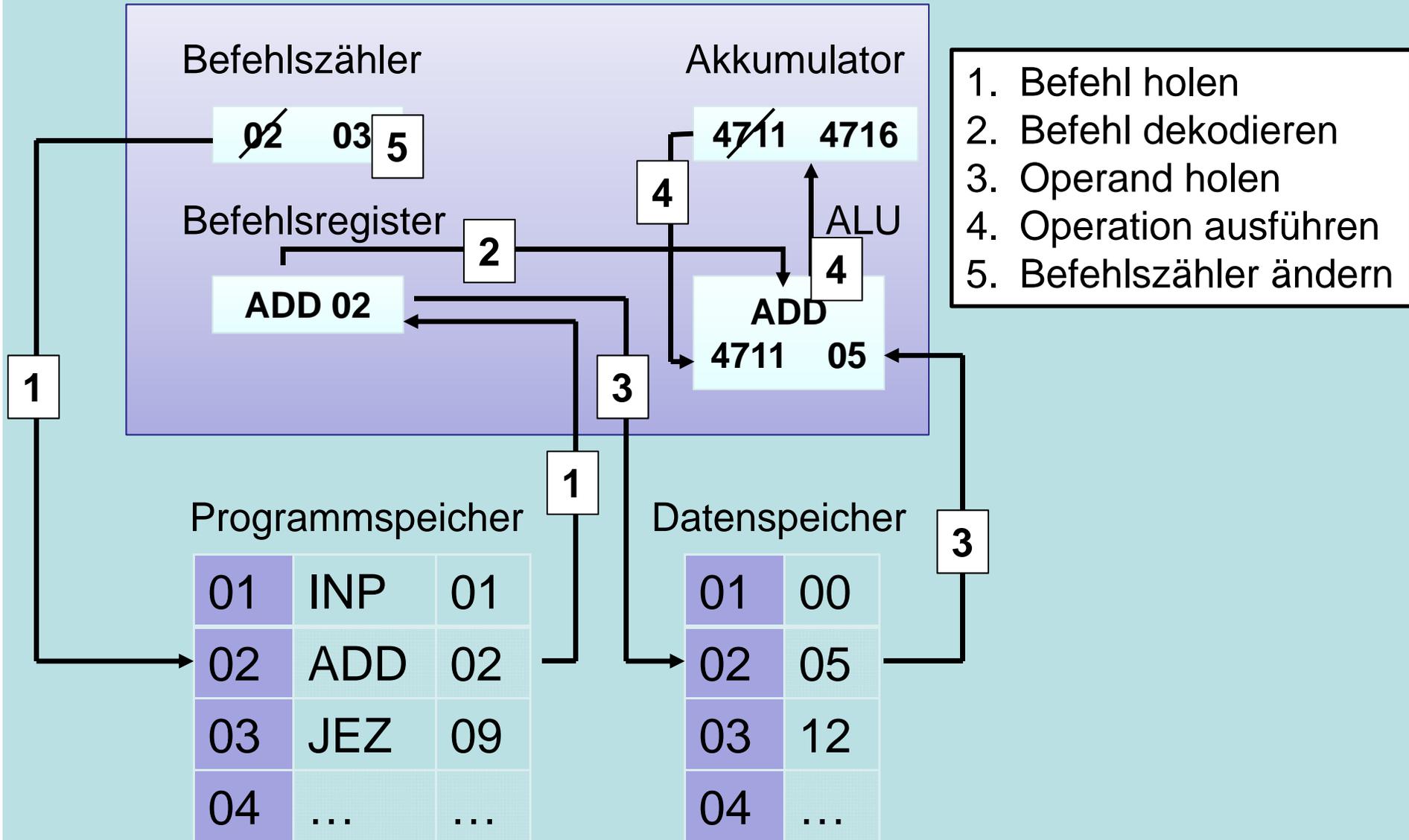
# Befehle der Registermaschine II

- Sprungbefehle:

Codierung	Syntax	Semantik	Beschreibung
0x0B	JMP <i>adresse</i>	PC = <i>adresse</i>	Verzweigen
0x0C	JEZ <i>adresse</i>	Falls $f(0) = 0$ , dann PC = <i>adresse</i>	
0x0D	JNE <i>adresse</i>	Falls $f(0) \neq 0$ , dann PC = <i>adresse</i>	
0x0E	JLZ <i>adresse</i>	Falls $f(0) < 0$ , dann PC = <i>adresse</i>	
0x0F	JLE <i>adresse</i>	Falls $f(0) \leq 0$ , dann PC = <i>adresse</i>	
0x10	JGZ <i>adresse</i>	Falls $f(0) > 0$ , dann PC = <i>adresse</i>	
0x11	JGE <i>adresse</i>	Falls $f(0) \geq 0$ , dann PC = <i>adresse</i>	

- Nach jedem Befehl, wenn es **keine Verzweigung** gab:
  - PC = PC + 1

# Befehlszyklus der Registermaschine



1. Befehl holen
2. Befehl dekodieren
3. Operand holen
4. Operation ausführen
5. Befehlszähler ändern

# Beispielprogramm

- Was tut folgendes Programm?
  - 06000702080105010C09010207020B0309020A99

01	06	00
02	07	02
03	08	01
04	05	01
05	0C	09
06	01	02
07	07	02
08	0B	03
09	09	02
10	0A	99



01	LDK	00
02	STA	02
03	INP	01
04	LDA	01
05	JEZ	09
06	ADD	02
07	STA	02
08	JMP	03
09	OUT	02
10	HLT	99

Akku auf 0
Akku → Adr. 2
Eingabe → Adr. 1
Adr. 1 → Akku
Akku = 0 ? Ja: 09
Akku + Adr. 2
Akku → Adr. 2
Weiter bei 03
Ausgabe Adr. 2
Programmende

# Leistungsmerkmale von Prozessoren

- Wortbreite eines Prozessors setzt sich aus der Wortbreite (WB) seiner einzelnen Komponenten zusammen
  - Arbeits- und Datenregister: WB bestimmt **maximale Zahlendarstellung**
  - Datenbus: WB gibt Anzahl Bits an, die geschrieben/gelesen werden können
  - Adressregister und Adressbus: WB gibt **maximale Speichergröße** an
  - Steuerbus: WB bestimmt Art der Peripherieanschlüsse
  - Beispiele: 4 Bit-, 8 Bit-, 16 Bit-, 32 Bit-, 64 Bit-Rechner
- Taktfrequenz
  - Bestimmt im Wesentlichen wie viele **Takte** ein Prozessor **pro Sekunde** ausführen kann
  - Vorsicht: Ein Prozessor kann für die Befehlsausführung bestimmter Befehle länger brauchen als ein anderer, was eine höhere Taktfrequenz wieder niedriger macht
  - Je höher die Taktfrequenz, desto größer sollte auch der Arbeitsspeicher sein, damit nicht ständig Daten transportiert werden müssen (Swapping)
- Typische Maße (insbesondere für Benchmarks):
  - Million Instructions per Sec.: **MIPS**, Floating Point Operations per Sec.: **FLOPS**

# Typische Einsatzbereiche

- Prozessoren für Personal Computer (PCs)
  - Aktuelle 64-Bit-Varianten erlauben insbesondere die Erweiterung des Arbeitsspeichers auf mehr als 4 Gigabyte

Prozessor	Datenbreite	Adressbreite
8086	16 Bit	20 Bit
80286	16 Bit	24 Bit
80386, 80486, Pentium, Athlon	32 Bit	32 Bit
Itanium (Intel), Opteron(AMD)	64 Bit	64 Bit
PowerPC V	64 Bit	64 Bit

- Proprietäre Prozessoren
  - Großrechner (main frames): Hard- und Software aus einer Hand
  - Unterhaltungselektronik
- Prozessoren in eingebetteten Systemen (embedded systems)
  - Riesige Typenvielfalt (4-Bit, 8-Bit, 16-Bit, 32-Bit)
  - Bekannte Serien: ARM, MSP, viele zusätzliche Komponenten/Preise

# Leistungssteigerung durch Cache

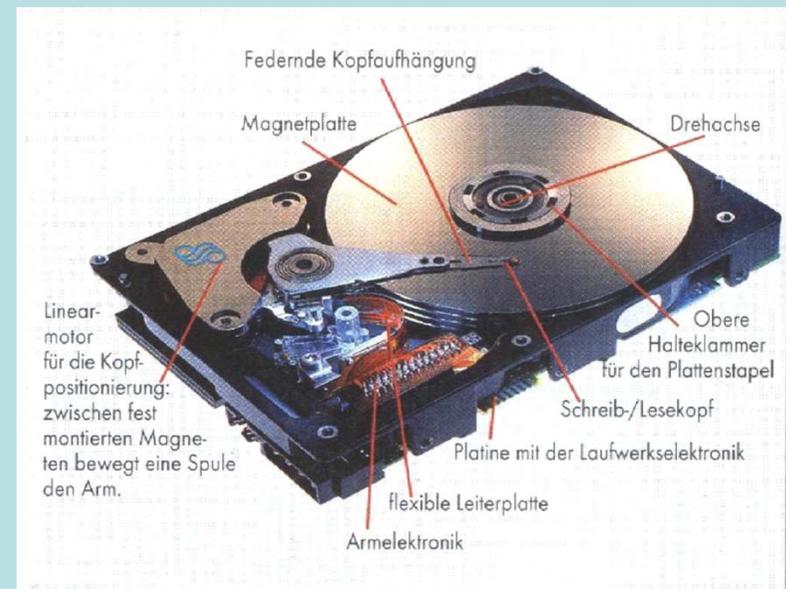
- Ein **Cache** dient dazu, Daten, die bereits mindestens einmal im Arbeitsspeicher geladen waren, beim nächsten Zugriff schneller verfügbar zu machen (Puffer-Speicher)
  - Vermeidung des Von-Neumann-Flaschenhalses
  - Speicherzugriffsgeschwindigkeit bestimmt entscheidend die Performanz eines Computersystems
- **Level-1-Cache**
  - Direkt im Prozessorkern untergebracht (wenige Kilobyte)
  - Statt einzelner Befehle zu laden, werden Blöcke geladen
- **Level-2-Cache** (ggf. weitere, z.B. **Level-3-Cache**)
  - Im Prozessor oder auf dem Mainboard untergebracht (nicht im Kern)
  - Schneller als RAM-Arbeitsspeicher, langsamer als L1, aber größer
- **Verdrängungsstrategien:**
  - LIFO, FIFO, LRU, LFU, Round-Robin, Random, ...
  - Optimal: Offline-Verfahren (Programmablauf muss vorher bekannt sein!)

- Besteht im Wesentlichen aus Halbleiterbausteinen, die als **Random Access Memory (RAM)** bezeichnet werden
  - Auf Daten kann direkt lesend und schreibend zugegriffen werden
  - Der Inhalt ist „flüchtig“, d.h. er bleibt maximal solange erhalten, wie die Speicherbausteine mit Strom versorgt sind
  - Jede Speicherzelle ist 1 Byte groß und besitzt eine eigene Adresse
  - Bus-/Registerbreite des Prozessors bestimmt die in einem Befehl verarbeitbare bzw. adressierbare Anzahl von Bytes
- Beispiele für RAM-Arten:
  - Dynamic RAM (DRAM): langsam und kostengünstig
  - Static RAM (SRAM): schneller und teurer
- Beispiele für RAM-Bauformen:
  - Synchronous Dynamic Ram (SD-RAM), DIMM-Module
  - Double Date Rate (DDR-RAM), DIMM-Module
  - Rambus RAM (RD-RAM), RIMM-Module

- Anschluss von Peripherie und Kommunikation zwischen den einzelnen Bestandteilen des Mainboards
- Internes Bussystem
  - Datenbus: Bidirektionale Datenübertragung zwischen den Einheiten
  - Adressbus: Unidirektionale Adressübermittlung zum Speicher
  - Steuerbus: Koordination exklusiver Zugriffe auf Daten- und Adressbus
- Wesentliche Kommunikationsarten
  - Regelmäßiges Abfragen (**Polling**)
  - Unterbrechungen (**Triggering** durch Interrupt Requests, IRQs)
- Datenübertragungen via
  - E-/A-Port bzw. Basisadressen
  - Vordefinierter Speicherbereiche (Memory Mapped I/O)
  - Direkte Übertragung von Daten in den Arbeitsspeicher im Hintergrund, d.h. ohne Belegung des Prozessors (Direct Memory Access, DMA)

- Erweiterungskarten
  - Peripheral Component Interface (PCI)
  - Accelerated Graphics Port (AGP)
  - Personal Computer Memory Card International Association (PCMCIA, PC-Card)
- Laufwerke
  - Enhanced Integrated Device Electronics (EIDE)
  - Small Computer System Interface (SCSI)
  - Serial-Advanced Technology Attachment (S-ATA)
- Peripherie
  - Universal Serial Bus (USB)
  - FireWire (IEEE-1394)
  - Infrarot, Bluetooth, WLAN, ZigBee
- Art des Anschlusses häufig **seriell** oder **parallel**

- Massenspeicher(Festplatte, USB-Stick, CD, DVD)
  - Magnetische Datenträger:  
magnetische Bereiche mit gegensätzlicher Polarität
  - Optische Datenträger:  
reflektierende Metallfläche, die per Laserstrahl abgetastet wird
  - Magneto-optische Datenträger:  
Mischverfahren: durch Laserstrahl magnetisch veränderbare Fläche
- Eingabegeräte  
(Tastatur, Maus, Scanner, Digitalkamera)
- Ausgabegeräte  
(Grafikkarte, Bildschirm, Drucker)



# Speicherhierarchie

