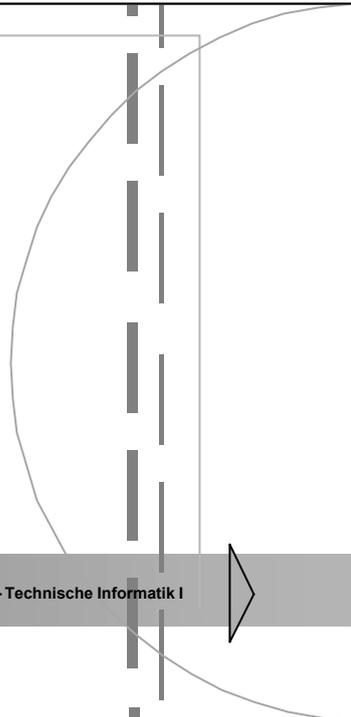


## Kapitel 3

### 3.1 Befehlssatz

### 2.2 Pipelining

Bernd Becker – Technische Informatik I



## 3.1 Befehlssatz

### ■ Die Extreme:

### CISC-Rechner und RISC-Rechner

### Begriffe

- CISC = Complex Instruction Set Computer
- RISC = Reduced Instruction Set Computer
- Vergleich der beiden Prinzipien

BB TI I

3.1/2

## CISC – Complex Instruction Set Computer

- Charakterisierung eines CISC Rechners
  - großer Satz von Maschinenbefehlen, zum Teil auch Maschinenbefehle, die recht komplexe Operationen auszuführen haben
    - sehr unterschiedliche Ausführungszeiten der verschiedenen Maschinenbefehle
    - Steuerung nicht durch Hardware, sondern durch ein Mikroprogramm, das in einem ROM auf dem Prozessor abgespeichert ist.
- Erster moderner Rechner (IBM 369) war ein CISC Rechner

BB TI I

3.1/3

## Argumente für CISC

- Komplexe Befehle schließen die semantische Lücke zwischen den verwendeten Hochsprachen und der Hardware
  - **Maschine, die „direkt“ Hochsprache verarbeiten kann**
- **Komplexe Befehle verringern den Hauptspeicherbedarf** für den Programmcode.
- **Je kürzer das Maschinenprogramm, desto schneller läuft es ab**, da weniger auf den Hauptspeicher zugegriffen werden muss.
- **Komplexe Befehle vereinfachen den Compiler.**
- Je mehr Funktionalität unterhalb der Maschinsprache, desto zuverlässiger ein Rechner (übliche Ansicht: "**Hardware ist sicher, Software enthält Fehler**")

BB TI I

3.1/4

## Argumente gegen CISC

- Viele **Zugriffe auf den Programmspeicher** können aufgrund von „Cache-Strategien“ (siehe nächstes Kapitel) **schnell durchgeführt** werden.  
Grund: Lokalitätseigenschaften der Codeausführung.
- **Hardware ist nicht zuverlässiger als Software.**  
Hardware-Fehler sind schwieriger zu korrigieren als Software-Fehler
- **Die komplexen Befehle**, die eigentlich die Compiler vereinfachen sollten, **werden kaum von den Compilern benutzt.**

BB TI I

3.1/5

## Statistik für CISC-Rechner (ca. 1990)

Befehlstypen	Ausführungshäufigkeit
Einfache Befehle (move, add register, branch, call, return, ...)	84%
Boolesche Operationen auf Wörtern (and, not, or, ...)	7%
Gleitkommaoperationen	4%

- Mehr als 90% der ausgeführten Befehle sind einfach, könnten schnell ausgeführt werden !
- Die Unterstützung der restlichen 10% verlangsamt in der Regel die Ausführungsgeschwindigkeit der obigen 90% !

BB TI I

3.1/6

## RISC – Reduced Instruction Set Computer

- Charakterisierung
  - einfache Befehle, kleine Befehlssätze
  - in einem Verarbeitungsschritt ausführbar
  - großer, interner Registersatz
  - leistungsfähige Speicherhierarchie
  - Einsparung des Mikrobefehlssatzes, festverdrahtetes Steuerwerk
  
- Beispiel: MIPS RS2000/3000 (1982-89)

BB TI I 3.1/7

## Vergleich RISC versus CISC

	RISC	CISC
Ausführungszeit	1 Datenpfadzyklus	≥1 Datenpfadzyklen
Instruktionsanzahl	klein	groß
Instruktionsformat	einfach / einheitlich	variabel
Steuerung über	Hardware	Mikroprogramm
Hauptspeicherzugriffe	LOAD / STORE-Architektur	keine Einschränkungen
Pipelining	möglich	kaum möglich
Verlagerung d. Komplexität	Compiler	Hardware

BB TI I 3.1/8

## Designprinzipien für Rechner

- Was sind die Schlüsseloperationen für diesen speziellen Rechnertyp ?
- Entwurf des Datenpfades (Rechenwerk mit Verbindungsstruktur) für diese Schlüsseloperationen
- Entwurf der Maschinenbefehle mit dem Ziel, dass möglichst jeder Befehl in einem Datenpfadzyklus (Fetch-Decode-Execute) ausführbar ist (kein Mikroprogramm)
- Erweiterung des Befehlssatzes nur dann, wenn dadurch die Maschine nicht langsamer wird.