

## RISC & CISC

**CISC** - Complex Instruction Set Computing

- klassischer Befehlssatz
- Ziel: Komplexität in die CPU  $\Rightarrow$  Speichersparnis
- viele leistungsfähige Einzelbefehle
- unterschiedliche Befehlsformate/breite
- Nachteile: hoher Decodierungsaufwand, längere Einzelausführungszeit
- ursprünglich mikroprogrammiert
- *Beispielarchitektur* x86  
(ab Pentium Pro interne Umsetzung in RISC/  $\mu$ Op)

**RISC** - Reduced Instruction Set Computing

- reduzierter Befehlssatz, einheitliche Befehlsbreite
  - kurze Decodierzeit & schnelle Einzelbefehle
- $\Rightarrow$  schnellere Interrupts
- nur *Load* und *Store* greifen auf Speicher zu
- $\Rightarrow$  großer Registersatz
- führt die SPEC-Top500 an
  - teilweise trotzdem komplexere Befehle etwa MMX oder AltiVec
  - *Beispielarchitektur* PPC, ARM, AT-Mega

eine Karte

Eine Karte mit einem Label

noch eine Karte

Und ein Verweis auf die LabelKarte (siehe # 2).