

# Grundlagen Rechnertechnik

Prof. Dr. Peter Gerwinski

27. November 2012

# 3 Architekturmerkmale von Prozessoren

## 3.1 Speicherarchitekturen

- *Von-Neumann-Architektur*

Es gibt nur 1 Hauptspeicher, in dem sich sowohl die Befehle als auch die Daten befinden.

Vorteil: Flexibilität in der Speichernutzung

Nachteil: Befehle können überschrieben werden.

→ Abstürze und Malware möglich

- *Harvard-Architektur*

Es gibt 2 Hauptspeicher. In einem befinden sich die Befehle, im anderen die Daten.

Vorteil: Befehle können nicht überschrieben werden

→ sicherer als Von-Neumann-Architektur

Nachteile: Leitungen zum Speicher (Bus) müssen doppelt vorhanden sein, freier Befehlsspeicher kann nicht für Daten genutzt werden.

- Weitere Kombinationen

Hauptspeicher und I/O-Ports gemeinsam oder getrennt,

Hauptspeicher und Prozessorregister gemeinsam oder getrennt

## 3 Architekturmerkmale von Prozessoren

### 3.2 Registerarchitekturen

- Mehrere Register, einzeln ansprechbar
- *Akkumulator*: Nur 1 Register kann rechnen.
- *Stack-Architektur*: Stapel, „umgekehrte Polnische Notation“

## 3 Architekturmerkmale von Prozessoren

### 3.2 Registerarchitekturen

- Mehrere Register, einzeln ansprechbar
- *Akkumulator*: Nur 1 Register kann rechnen.
- *Stack-Architektur*: Stapel, „umgekehrte Polnische Notation“

Operationen: typischerweise nur  $=$ ,  $+=$ ,  $-=$ ,  $*=$ ,  $/=$ , ...

# 3 Architekturmerkmale von Prozessoren

## 3.2 Registerarchitekturen

- Mehrere Register, einzeln ansprechbar
- *Akkumulator*: Nur 1 Register kann rechnen.
- *Stack-Architektur*: Stapel, „umgekehrte Polnische Notation“

Operationen: typischerweise nur  $=$ ,  $+=$ ,  $-=$ ,  $*=$ ,  $/=$ , ...

Beispiel:  $c = a + 2 * b;$

# 3 Architekturmerkmale von Prozessoren

## 3.2 Registerarchitekturen

- Mehrere Register, einzeln ansprechbar
- *Akkumulator*: Nur 1 Register kann rechnen.
- *Stack-Architektur*: Stapel, „umgekehrte Polnische Notation“

Operationen: typischerweise nur  $=$ ,  $+=$ ,  $-=$ ,  $*=$ ,  $/=$ , ...

Beispiel:  $c = a + 2 * b;$

C:

$R = b;$

$R *= 2;$

$R += a;$

$c = R;$

# 3 Architekturmerkmale von Prozessoren

## 3.2 Registerarchitekturen

- Mehrere Register, einzeln ansprechbar
- *Akkumulator*: Nur 1 Register kann rechnen.
- *Stack-Architektur*: Stapel, „umgekehrte Polnische Notation“

Operationen: typischerweise nur `=`, `+=`, `-=`, `*=`, `/=`, ...

Beispiel: `c = a + 2 * b;`

C:	Mehrere Register:
<code>R = b;</code>	<code>movl (b), %eax</code>
<code>R *= 2;</code>	<code>imull \$2, %eax, %eax</code>
<code>R += a;</code>	<code>addl (a), %eax</code>
<code>c = R;</code>	<code>movl %eax, (c)</code>

(IA-32-Assembler)

# 3 Architekturmerkmale von Prozessoren

## 3.2 Registerarchitekturen

- Mehrere Register, einzeln ansprechbar
- *Akkumulator*: Nur 1 Register kann rechnen.
- *Stack-Architektur*: Stapel, „umgekehrte Polnische Notation“

Operationen: typischerweise nur `=`, `+=`, `-=`, `*=`, `/=`, ...

Beispiel: `c = a + 2 * b;`

C:	Mehrere Register:	Akkumulator:
<code>R = b;</code>	<code>movl (b), %eax</code>	<code>load (b)</code>
<code>R *= 2;</code>	<code>imull \$2, %eax, %eax</code>	<code>mul \$2</code>
<code>R += a;</code>	<code>addl (a), %eax</code>	<code>add (a)</code>
<code>c = R;</code>	<code>movl %eax, (c)</code>	<code>store (c)</code>
	(IA-32-Assembler)	(Pseudo-Assembler)



# 3 Architekturmerkmale von Prozessoren

## 3.2 Registerarchitekturen

- Mehrere Register, einzeln ansprechbar
- *Akkumulator*: Nur 1 Register kann rechnen.
- *Stack-Architektur*: Stapel, „umgekehrte Polnische Notation“

Operationen: typischerweise nur `=`, `+=`, `-=`, `*=`, `/=`, ...

Beispiel: `c = a + 2 * b;`

C:	Mehrere Register:	Akkumulator:	Register-Stack:
<code>R = b;</code>	<code>movl (b), %eax</code>	<code>load (b)</code>	<code>push (a)</code>
<code>R *= 2;</code>	<code>imull \$2, %eax, %eax</code>	<code>mul \$2</code>	<code>push (b)</code>
<code>R += a;</code>	<code>addl (a), %eax</code>	<code>add (a)</code>	<code>push \$2</code>
<code>c = R;</code>	<code>movl %eax, (c)</code>	<code>store (c)</code>	<code>mul</code>
	(IA-32-Assembler)	(Pseudo-Assembler)	<code>add</code>
			<code>pop (c)</code>

# 3 Architekturmerkmale von Prozessoren

## 3.2 Registerarchitekturen

Beispiele:

- Intel IA-32 (i386, Nachfolger und Kompatible):  
Mehrere Register, für verschiedene Zwecke spezialisiert (unübersichtlich),  
Fließkommaregister: Stack-Architektur
- Atmel AVR (z. B. ATmega):  
32 Register
- 6502 (heute: Renesas-Mikro-Controller):  
3 Register: A, X, Y. Nur A kann rechnen → Akkumulator
- Java Virtual Machine (JVM):  
Stack-Architektur
- Redcode:  
Jede Speicherzelle fungiert als Register

## 3 Architekturmerkmale von Prozessoren

### 3.3 Befehlssätze

- *Complex Instruction Set Computer (CISC)*

Umfangreiche Befehlssätze, mächtige Befehle

→ komfortable manuelle Programmierung in Assembler

→ längere Abarbeitungszeit der einzelnen Befehle

Realisierung: „Prozessor im Prozessor“ – *Mikroprogramme*

Beispiele: IA-32, AMD-64

- *Reduced Instruction Set Computer (RISC)*

wenige, wenig mächtige Befehle

→ Programmierung in Assembler für Menschen unkomfortabel

→ schnelle Abarbeitung der Befehle

Beispiele: Atmel AVR, Redcode

## 3 Architekturmerkmale von Prozessoren

### 3.3 Befehlssätze

- *Complex Instruction Set Computer (CISC)*
- *Reduced Instruction Set Computer (RISC)*
- *Very Long Instruction Word (VLIW)* und *Explicitly Parallel Instruction Computing (EPIC)*

mehrere Befehle gleichzeitig ausführbar

→ mehr Rechenleistung möglich

→ Programmierung sehr aufwendig

Beispiel: IA-64

## 3 Architekturmerkmale von Prozessoren

### 3.3 Befehlssätze

- *Complex Instruction Set Computer (CISC)*
- *Reduced Instruction Set Computer (RISC)*
- *Very Long Instruction Word (VLIW)* und *Explicitly Parallel Instruction Computing (EPIC)*

mehrere Befehle gleichzeitig ausführbar

→ mehr Rechenleistung möglich

→ Programmierung sehr aufwendig

Beispiel: IA-64

- *Orthogonaler Befehlssatz*