

Mikroprozessoren

Aufbau und Funktionsweise

Christian Richter

Ausgewählte Themen der Multimediakommunikation
SS 2005

Gliederung

1 Was ist ein Mikroprozessor?

- Aufgaben
- Möglichkeiten der Einteilung

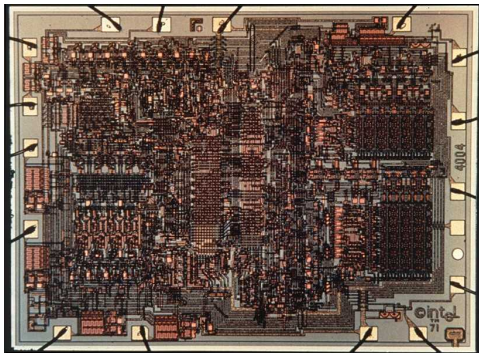
2 Aufbau und Funktionsweise

- Komponenten eines Mikroprozessors
- Ein beispielhafter Programmablauf
- Interrupts

3 Optimierungen

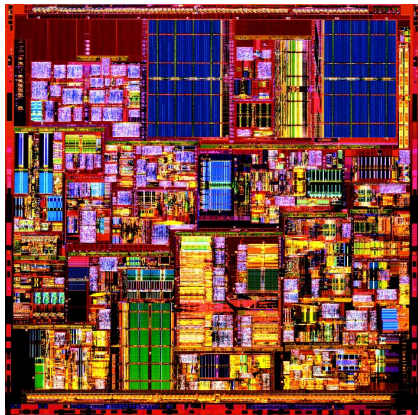
4 Zukünftige Möglichkeiten

Was ist ein Mikroprozessor?



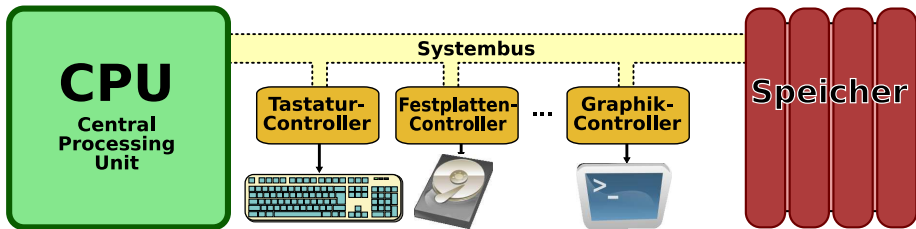
1971: Intel 4004
2.250 Transistoren auf 12mm^2 ($10\mu\text{m}$)

QUELLE: STANFORD UNIVERSITY, VLSI-GROUP



2002: Intel Pentium4 *Northwood*
55 Mio. Trans. auf 131mm^2 (130nm)

Aufgaben

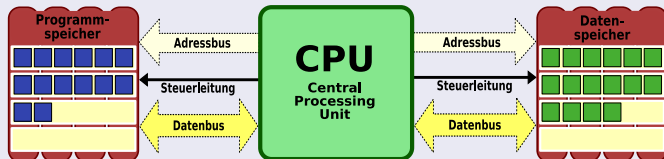


- **Ansteuerung der Peripheriegeräteecontroller**
Einlesen der Programme und Daten, Ausgabe der Ergebnisse
- **Speicherzugriff**
- **Verarbeitung der Daten**
Logische (AND, OR, XOR, NOT) und arithmetische Operationen

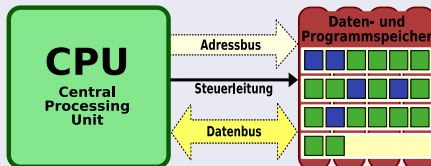
Möglichkeiten der Einteilung I

Harvard vs. von Neumann

- **HARVARD-Architektur** (getrennte Daten- und Programmspeicher)



- **VON NEUMANN-Architektur** (gemeinsamer Speicher)



Möglichkeiten der Einteilung II

RISC vs. CISC

RISC (Reduced Instruction Set Computing):

- Wenige, möglichst einfache Befehle
- schnelle Ausführung (meist 1 Befehl/Takt)

CISC (Complex Instruction Set Computing):

- Sehr viele, komplexe Befehle
- Langsame Ausführung, dafür sehr mächtig

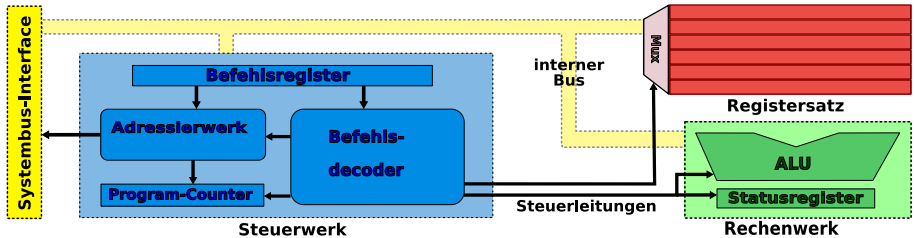
Universal- vs. Spezialprozessoren

- Einteilung nach Flexibilität der Prozessoren

Gliederung

- 1 Was ist ein Mikroprozessor?
 - Aufgaben
 - Möglichkeiten der Einteilung
- 2 Aufbau und Funktionsweise**
 - **Komponenten eines Mikroprozessors**
 - **Ein beispielhafter Programmablauf**
 - **Interrupts**
- 3 Optimierungen
- 4 Zukünftige Möglichkeiten

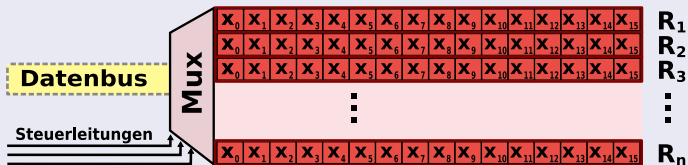
Komponenten eines Mikroprozessors



Grobe Aufteilung in Funktionsblöcke:

- **Steuerwerk**
- **Register**
- **Rechenwerk (ALU)**
- **Systembus-Interface**

Register



- aufgebaut aus Flipflops direkt auf der CPU (voller Prozessortakt)
- direkt durch Steuerleitungen wählbar (keine Adressierung nötig)
- können in einem Takt gelesen bzw. beschrieben werden

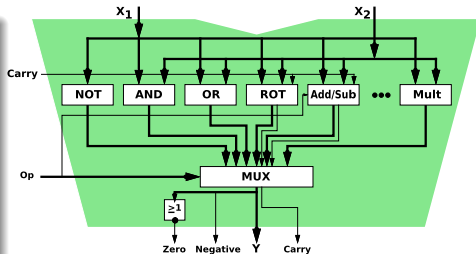
General Purpose Register können frei gelesen/beschrieben werden
(für Operanden, Adressen und Ergebnisse)

Spezialregister für die interne Funktion des Prozessors benötigt
z.B. Programmzähler, Befehlsregister, Statusregister ...
(nur eingeschränkt les-/beschreibbar)

Rechenwerk

ALU (Arithmetic Logical Unit)

- Schaltnetz (\Rightarrow gedächtnislos)
- verknüpft zwei Operanden **logisch** (AND, OR, NOT ...) oder **arithmetisch** (+, -, ·, /)
- Operation durch Multiplexer wählbar



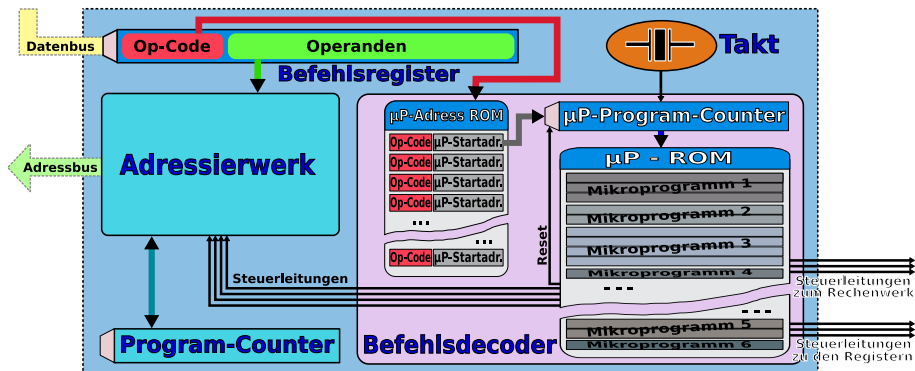
Nach: Volkmar Sieh »Script TI II« S.178

Statusregister (Condition Code Register CCR)

Speichert Informationen über das Ergebnis der letzten Operation z.B:

- **Z(ero)-Bit** Ergebnis = 0 ?
- **N(egative)-Bit** Ergebnis < 0 ?
- **O(overflow)-Bit** arithm. Überlauf?
- **C(arry)-Bit** Übertrag

Steuerwerk



- Befehl wird ins **Befehlsregister** eingelesen
- **Befehlsdecoder** startet Op-Code-spezifisches μ Programm (bestimmt, wann welche Steuerleitungen geschaltet werden)
- **Program-Counter** enthält immer die Adresse des *nächsten* Befehls

Ein Programm als Beispiel

Programmstück für einen MC6800-Prozessor

Adresse	Hex-Code	Assembler	Bedeutung
\$0020	0E		8-Bit Wert \$0E (Dezimal: 14)
\$0021	05		8-Bit Wert \$05 (Dezimal: 5)
...
\$2000	96 20	LDA A \$20	Wert aus \$20 → Register A
\$2002	5F	CLR B	Setze Register B auf Null
\$2003	5C	INC B	Wert von Register B um 1 erhöhen
\$2004	9D 21	SUB A \$21	Register A minus Wert aus \$21
\$2006	2E 20 03	BGT \$2003	Sprung zu \$2003, wenn Ergebnis > 0
\$2009	5A	DEC B	Wert von Register B um 1 verringern

Ablauf des Programmes im Prozessor

Programm

- 1 \$2000 LDA A \$20
- 2 \$2002 CLR B
- 3 \$2003 INC B
- 4 \$2004 SUB A \$21
- 5 \$2006 BGT \$2003
- 6 \$2009 DEC B

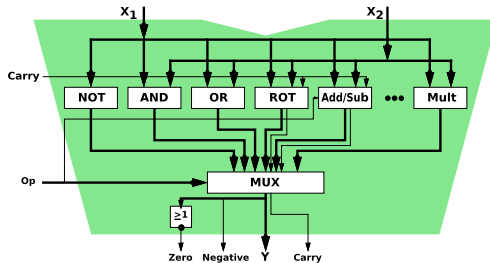
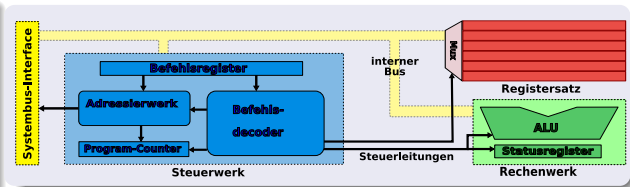
Register

PC: \$2004

CCR: Z=0, N=0

A: 0E (dez: 14)

B: 01



ALU

(Nach Volkmar Sieh »Script TI II« S.178)

Ablauf des Programmes im Prozessor

Programm

- 1 \$2000 LDA A \$20
- 2 \$2002 CLR B
- 3 \$2003 INC B
- 4 \$2004 SUB A \$21
- 5 \$2006 BGT \$2003
- 6 \$2009 DEC B

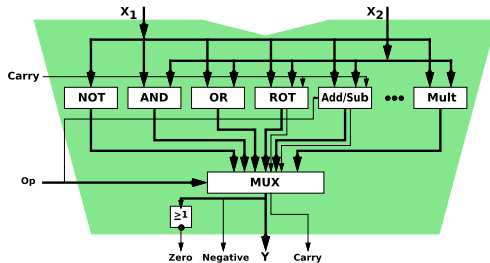
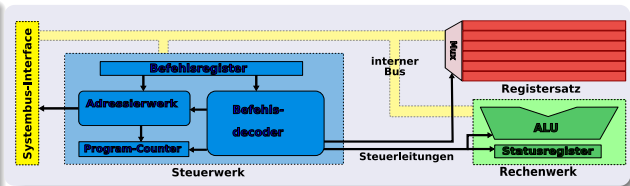
Register

PC: \$2006

CCR: Z=0, N=0

A: 09

B: 01



ALU

(Nach Volkmar Sieh »Script TI II« S.178)

Ablauf des Programmes im Prozessor

Programm

- 1 \$2000 LDA A \$20
- 2 \$2002 CLR B
- 3 \$2003 INC B
- 4 \$2004 SUB A \$21
- 5 \$2006 BGT \$2003
- 6 \$2009 DEC B

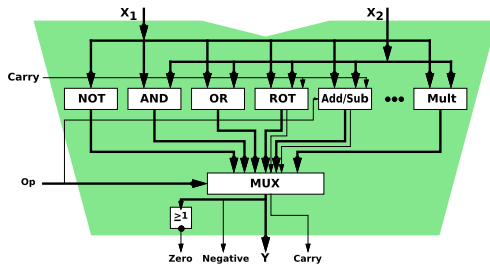
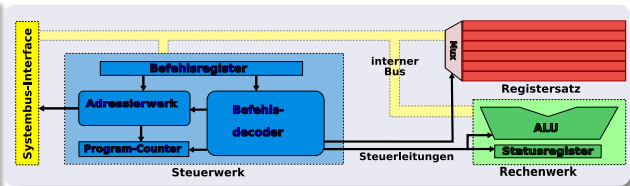
Register

PC: \$2003

CCR: Z=0, N=0

A: 09

B: 01



ALU

(Nach Volkmar Sieh »Script TI II« S.178)

Ablauf des Programmes im Prozessor

Programm

- 1 \$2000 LDA A \$20
- 2 \$2002 CLR B
- 3 \$2003 INC B
- 4 \$2004 SUB A \$21
- 5 \$2006 BGT \$2003
- 6 \$2009 DEC B

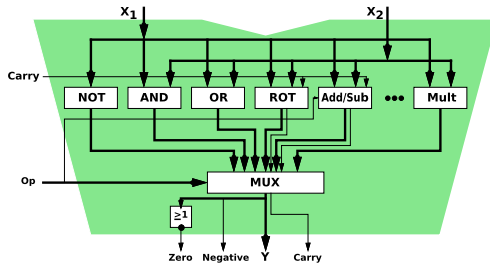
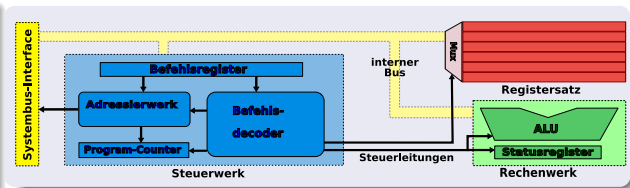
Register

PC: \$2004

CCR: Z=0, N=0

A: 09

B: 02



ALU

(Nach Volkmar Sieh »Script TI II« S.178)

Ablauf des Programmes im Prozessor

Programm

- 1 \$2000 LDA A \$20
- 2 \$2002 CLR B
- 3 \$2003 INC B
- 4 \$2004 SUB A \$21
- 5 \$2006 BGT \$2003
- 6 \$2009 DEC B

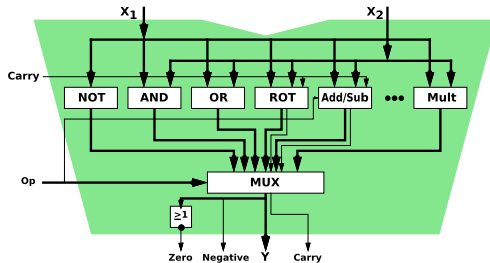
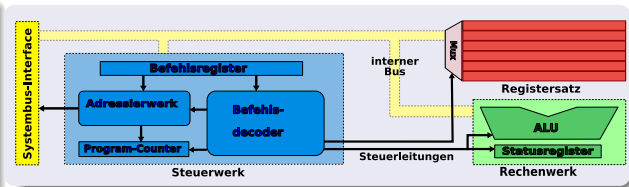
Register

PC: \$2006

CCR: Z=0, N=0

A: 04

B: 02



ALU

(Nach Volkmar Sieh »Script TI II« S.178)

Ablauf des Programmes im Prozessor

Programm

- 1 \$2000 LDA A \$20
- 2 \$2002 CLR B
- 3 \$2003 INC B
- 4 \$2004 SUB A \$21
- 5 \$2006 BGT \$2003
- 6 \$2009 DEC B

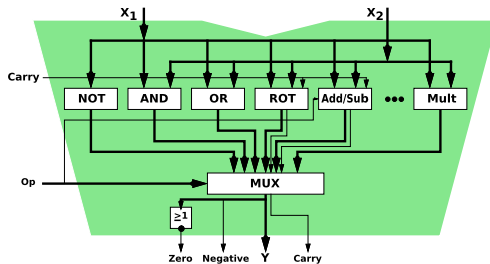
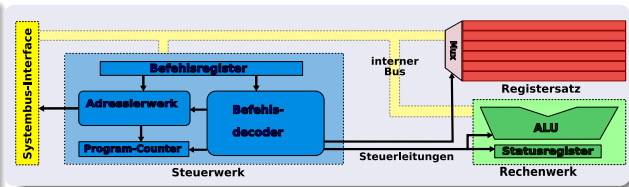
Register

PC: \$2003

CCR: Z=0, N=0

A: 04

B: 02



ALU

(Nach Volkmar Sieh »Script TI II« S.178)

Ablauf des Programmes im Prozessor

Programm

- 1 \$2000 LDA A \$20
- 2 \$2002 CLR B
- 3 \$2003 INC B
- 4 \$2004 SUB A \$21
- 5 \$2006 BGT \$2003
- 6 \$2009 DEC B

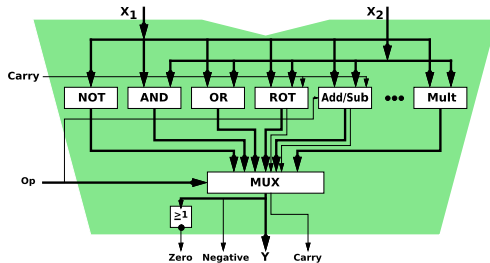
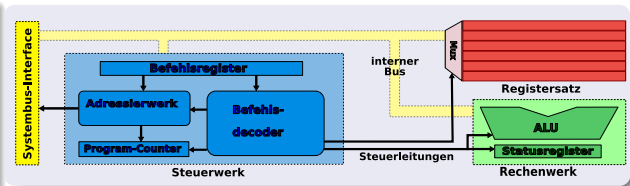
Register

PC: \$2004

CCR: Z=0, N=0

A: 04

B: 03



ALU

(Nach Volkmar Sieh »Script TI II« S.178)

Ablauf des Programmes im Prozessor

Programm

- 1 \$2000 LDA A \$20
- 2 \$2002 CLR B
- 3 \$2003 INC B
- 4 \$2004 SUB A \$21
- 5 \$2006 BGT \$2003
- 6 \$2009 DEC B

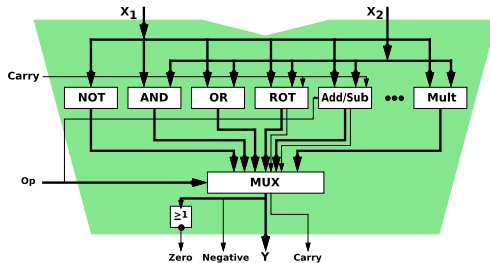
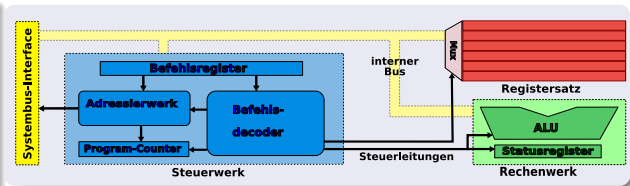
Register

PC: \$2006

CCR : Z=0, N=1

A: -1

B: 03



ALU

(Nach Volkmar Sieh »Script TI II« S.178)

Ablauf des Programmes im Prozessor

Programm

- 1 \$2000 LDA A \$20
- 2 \$2002 CLR B
- 3 \$2003 INC B
- 4 \$2004 SUB A \$21
- 5 \$2006 BGT \$2003
- 6 \$2009 DEC B

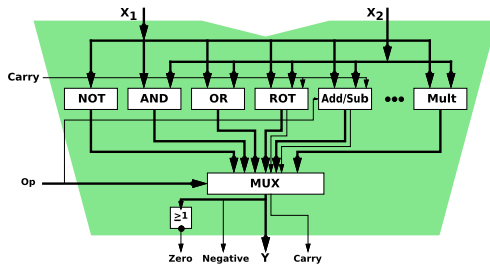
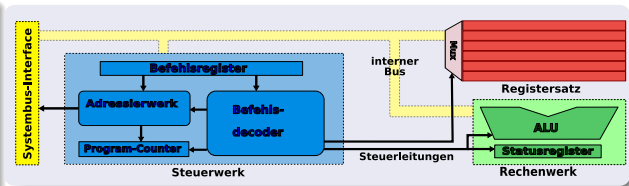
Register

PC: \$2009

CCR : Z=0, N=1

A: -1

B: 03



ALU

(Nach Volkmar Sieh »Script TI II« S.178)

Ablauf des Programmes im Prozessor

Programm

- 1 \$2000 LDA A \$20
- 2 \$2002 CLR B
- 3 \$2003 INC B
- 4 \$2004 SUB A \$21
- 5 \$2006 BGT \$2003
- 6 \$2009 DEC B

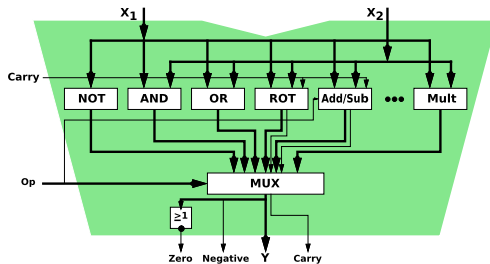
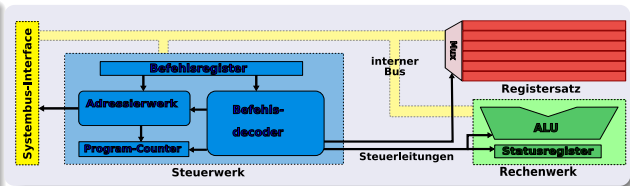
Register

PC: \$2009

CCR : Z=0, N=1

A: -1

B: 02



ALU

(Nach Volkmar Sieh »Script TI II« S.178)

Interrupts

Problem

Was passiert, wenn der Prozessor auf plötzliche Ereignisse reagieren muß?

Interne: Division durch Null

Externe: Taste gedrückt

Lösung: Interrupts

- Unterbrechung des laufenden Programmes wird angefordert (*Interrupt Request, IRQ*)
- extra Programmstück wird ausgeführt um den IRQ zu behandeln (*Interrupt Service Routine, ISR*)
- normales Programm wird fortgesetzt

Interrupts

Problem

Was passiert, wenn der Prozessor auf plötzliche Ereignisse reagieren muß?

Interne: Division durch Null

Externe: Taste gedrückt

Lösung: Interrupts

- Unterbrechung des laufenden Programmes wird angefordert (*Interrupt Request, IRQ*)
- extra Programmstück wird ausgeführt um den IRQ zu behandeln (*Interrupt Service Routine, ISR*)
- normales Programm wird fortgesetzt

Gliederung

- 1 Was ist ein Mikroprozessor?
 - Aufgaben
 - Möglichkeiten der Einteilung
- 2 Aufbau und Funktionsweise
 - Komponenten eines Mikroprozessors
 - Ein beispielhafter Programmablauf
 - Interrupts
- 3 Optimierungen**
- 4 Zukünftige Möglichkeiten

Höhere Taktfrequenz

Funktioniert!

Es wird 1 μ Schritt pro Taktzyklus ausgeführt.
höherer Takt \Rightarrow mehr Befehle pro Sekunde

ABER: Verlustleistung steigt linear mit der Taktfrequenz

$$P = C \cdot U_{Core}^2 \cdot f_{Takt}$$

- **Prozessor wird heiß**
 - hoher Kühlaufwand \Rightarrow Laut!
- **Hoher Stromverbrauch**
 - schlecht für mobile Geräte (begrenzte Akku-Kapazität)
 - schlecht für Umwelt und Geldbeutel

Höhere Taktfrequenz

Funktioniert!

Es wird 1 μ Schritt pro Taktzyklus ausgeführt.
höherer Takt \Rightarrow mehr Befehle pro Sekunde

ABER: Verlustleistung steigt linear mit der Taktfrequenz

$$P = C \cdot U_{Core}^2 \cdot f_{Takt}$$

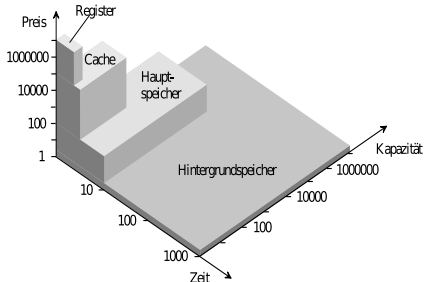
- **Prozessor wird heiß**

- hoher Kühlaufwand \Rightarrow Laut!

- **Hoher Stromverbrauch**

- schlecht für mobile Geräte (begrenzte Akku-Kapazität)
- schlecht für Umwelt und Geldbeutel

Beschleunigter Speicherzugriff



Quelle: Axel Böttcher »Foliensatz Rechnerntechnik«

Speicher ist um Größenordnungen
langsamer als Prozessor



häufige Wartetakte, bis benötigte
Daten zur Verfügung stehen

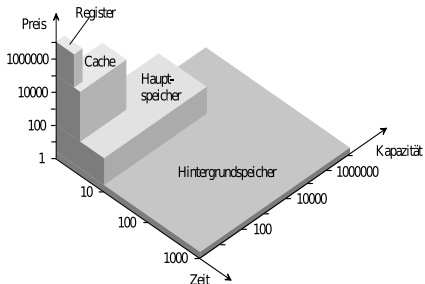
Lösung Speicherhierarchie

Cache: schneller Zwischenspeicher direkt auf dem Prozessor

Hauptspeicher: meist normaler DRAM-Speicher

Hintergrundspeicher: Festplatte, Netzwerk etc.

Beschleunigter Speicherzugriff



Quelle: Axel Böttcher »Foliensatz Rechnertechnik«

Speicher ist um Größenordnungen
langsamer als Prozessor



häufige Wartetakte, bis benötigte
Daten zur Verfügung stehen

Lösung Speicherhierarchie

Cache: schneller Zwischenspeicher direkt auf dem Prozessor

Hauptspeicher: meist normaler DRAM-Speicher

Hintergrundspeicher: Festplatte, Netzwerk etc.

Paralleles Ausführen von Befehlen

Pipeline-Architektur (Synchronparallele Organisation)



Befehlsausführung wird in mehrere Phasen (laden, decodieren ...) eingeteilt. Ausführung je um eine Phase verschoben.

- **PRO** mit jedem Takt wird ein Befehl fertig (Fließband)
- **CON** Problem bei Sprüngen und Datenabhängigkeiten

Superskalare Prozessoren (Instruction Level Parallelism, ILP)

- Mehrere Funktionseinheiten (ALUs ...) vorhanden.
- Befehle im Voraus laden. Zuteilung durch Dispatcher.
- Nebenläufiges (spekulatives) Ausführen der Befehle.

Paralleles Ausführen von Befehlen

Pipeline-Architektur (Synchronparallele Organisation)



Befehlsausführung wird in mehrere Phasen (laden, decodieren ...) eingeteilt. Ausführung je um eine Phase verschoben.

- **PRO** mit jedem Takt wird ein Befehl fertig (Fließband)
- **CON** Problem bei Sprüngen und Datenabhängigkeiten

Superskalare Prozessoren (Instruction Level Parallelism, ILP)

- Mehrere Funktionseinheiten (ALUs ...) vorhanden.
- Befehle im Voraus laden. Zuteilung durch Dispatcher.
- Nebenläufiges (spekulatives) Ausführen der Befehle.

Befehlsatzerweiterungen

**Speziell optimierte Befehle für bestimmte,
häufig auftretende Operationen**

Vor allem für die Verarbeitung von Multimediadaten

Beispiele

- MMX
- SSE1/2/3
- 3Dnow!
- AltiVec

Gliederung

- 1 Was ist ein Mikroprozessor?
 - Aufgaben
 - Möglichkeiten der Einteilung
- 2 Aufbau und Funktionsweise
 - Komponenten eines Mikroprozessors
 - Ein beispielhafter Programmablauf
 - Interrupts
- 3 Optimierungen
- 4 Zukünftige Möglichkeiten

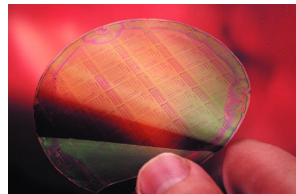
Zukünftige Möglichkeiten

● Dual-Core Prozessoren

- zwei vollständig getrennte Prozessorkerne auf einem Chip (ab Mitte 2005 verfügbar)

● Polymer-Prozessoren

- Transistoren auf Basis halbleitender Polymere
- biegsam und einfach zu verarbeiten
- aber noch sehr langsam (<13kHz) und empfindlich gegen Umwelteinflüsse



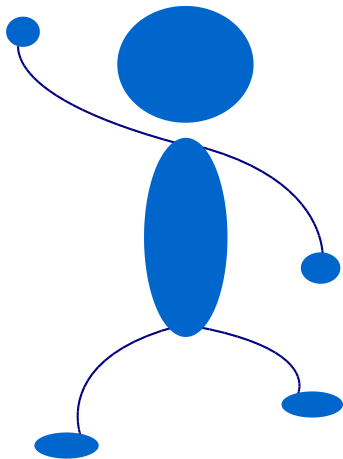
Quelle: c't-Magazin 2/1999 S.80

● Prozessoren mit supraleitenden Elementen (RSFQ)

- extrem geringe Verlustleistungen (z.B. 9mW bei 24GHz)

● Optische Datenverarbeitung

Vielen Dank!



Folien und Seminararbeit
sind zu finden unter:

<http://www.ch-r.de/>

(Natürlich erst *nach* Ende der Veranstaltung.)