

## Script generated by TTT

Title: Einf\_HF (13.05.2013)

Date: Mon May 13 14:27:15 CEST 2013

Duration: 78:48 min

Pages: 24



- Prof. J. Schlichter
    - Lehrstuhl für Angewandte Informatik / Kooperative Systeme
- Fakultät für Informatik, TU München  
E-Mail: [schlichter@in.tum.de](mailto:schlichter@in.tum.de)  
Tel.: 089-289 18654  
URL: <http://www11.in.tum.de/>

[Übersicht](#)

[Einführung](#)

[Datenbanken und Informationssysteme](#)

[Rechnerarchitektur](#)

[Systemsoftware](#)

[Grundlagen der Programmierung](#)

[Datenstrukturen und Algorithmen](#)

[Software-Entwicklung](#)

[Grundlagen von Rechnernetzen](#)

[Anwendungen von Rechnernetzen](#)

[Zusammenfassung](#)

Generated by Targeteam



- Fragestellungen des Abschnitts:
  - Aus welchen (Hardware-)Elementen setzt sich ein Rechner zusammen?
  - Wie kommunizieren die einzelnen Komponenten eines Rechners?
  - Wie sieht die Schnittstelle zwischen Hardware und Software aus (d.h. Maschinenbefehle)?
  - Wie werden Zahlen, Text, Bilder, und Töne intern dargestellt?

[Aufbau eines Rechners](#)

[Maschinenbefehle](#)

[Befehlszyklus](#)

[Interndarstellung von Information](#)



Generated by Targeteam



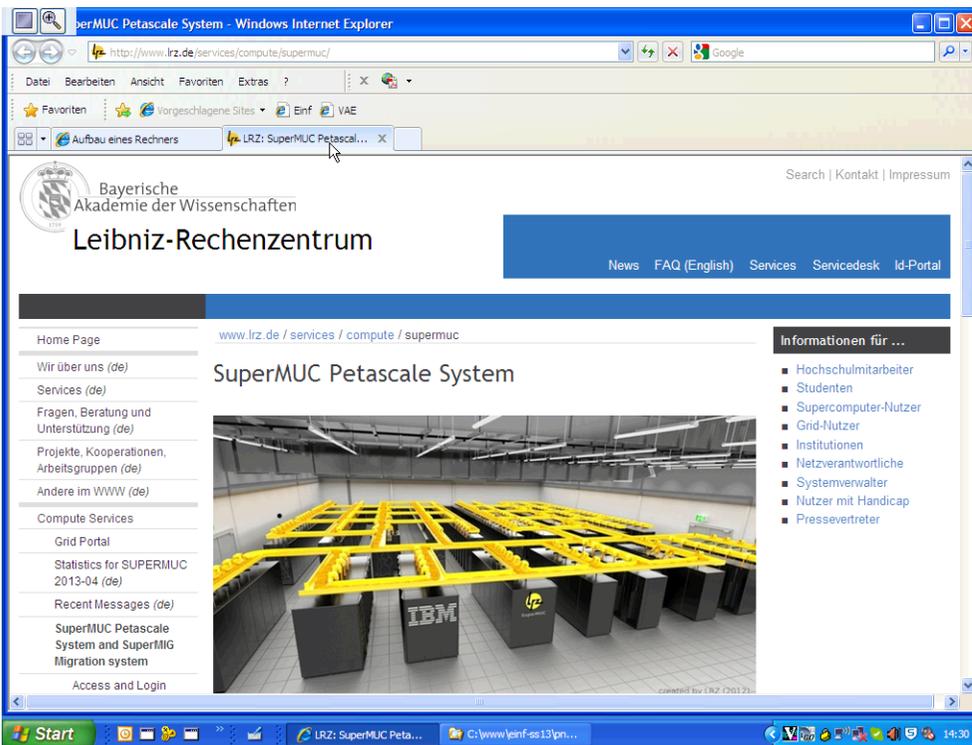
Hardware eines Computersystems: Rechner und Peripheriegeräte, z.B. Tastatur, Maus, Bildschirm, Drucker, Scanner. Sehr unterschiedliche Leistungsfähigkeit.

Beispiel eines Supercomputers

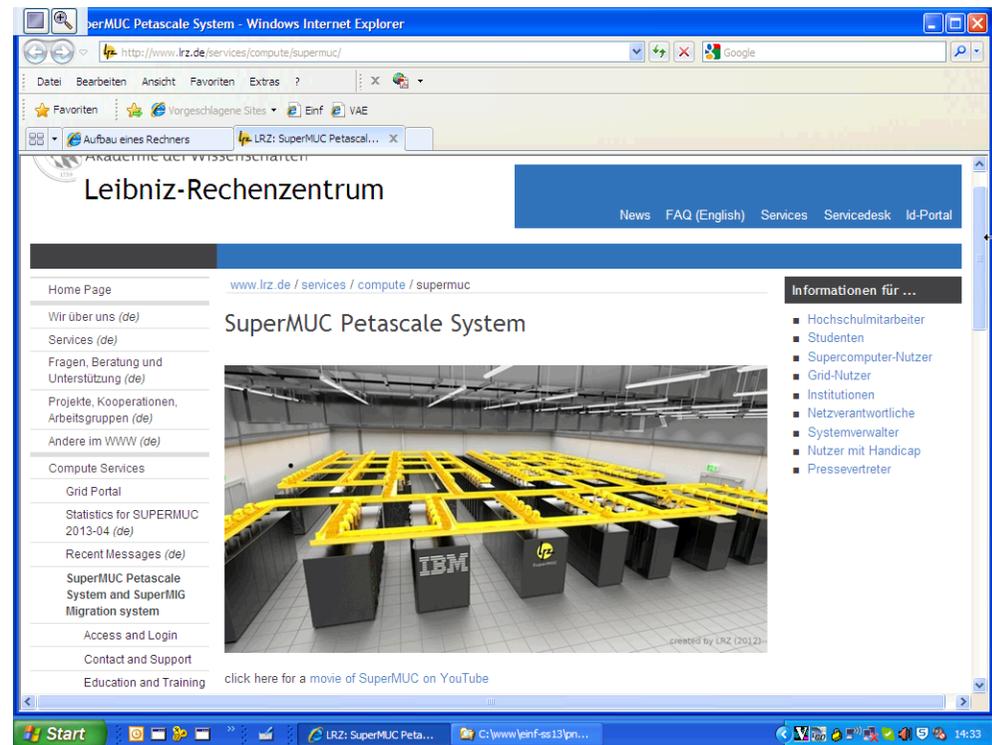


[Struktureller Aufbau eines Rechners](#)

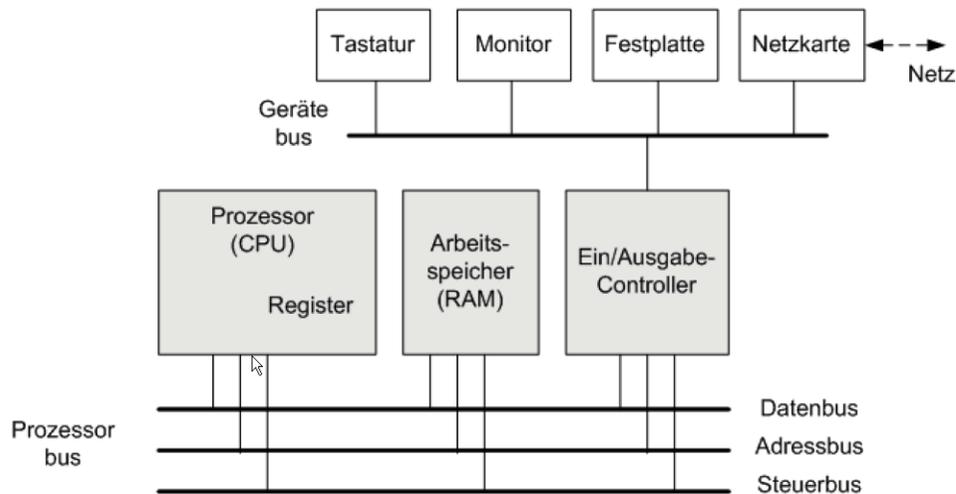
Generated by Targeteam



Struktureller Aufbau eines Rechners



Komponenten eines Rechners



**Komponenten eines Rechners**

**Busse zur Kommunikation**

**Kommunikation zwischen Arbeitsspeicher und CPU**

Im Prozessor: alle Rechen- und Umformvorgänge und die Ablaufsteuerung der Programme.

moderne Prozessoren bestehen aus mehreren Kernen

Register sind spezielle Speicherzellen im Prozessor.

Arbeitsspeicher (ASP; "main memory"): direkt adressierbare Speicherzellen der Größe 8 Bit zur Ablage von Programmen und binären Daten.

Ein/Ausgabe-Controller: setzen Nachrichten auf dem Gerätebus für Ein/Ausgabe-Geräte um, stellen Eingaben der Geräte am Bus bereit. Anschluss von Graphikkarten, Netzwerkkarten etc über den Gerätebus.

Den Ablauf der Informationsverarbeitungsvorgänge im Rechner steuern Programme.

*Generated by Targeteam*



Rechnerkomponenten kommunizieren über Prozessorbus und Gerätebus

Prozessorbus: schnelle Übertragung

Gerätebus (z.B. PCI): etwas langsamer, da Geräte langsamer als CPU oder Arbeitsspeicher.

Der Prozessorbus unterteilt sich in

Adressbus: zur Übergabe einer Adresse (32 oder 64 Bit)

Datenbus: zur Übermittlung von Daten (32 oder 64 Bit)

Steuerbus: mehrere Steuerleitungen

Generated by Targeteam

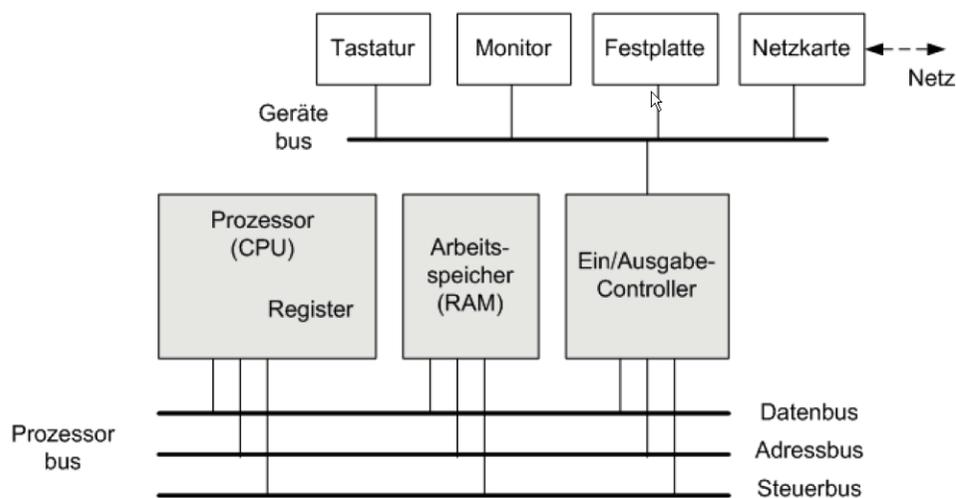
CPU und Arbeitsspeicher arbeiten nicht synchron ⇒ Koordination notwendig.

**Beispiel:** Übertragung von Daten aus dem Arbeitsspeicher zum Prozessor (CPU).

1. CPU legt die ASP-Adresse des gewünschten Datenelements auf den Adressbus.
2. CPU legt das Signal "Lesen" auf eine bestimmte Leitung im Steuerbus.
3. CPU legt das Signal "Adresse gültig" auf eine bestimmte Leitung im Steuerbus.
4. ASP ist passiv, "hört" aber auf Adress- und Steuerbus mit.
5. ASP erkennt, dass eine Adresse gültig ist und prüft, ob er mit dieser Adresse gemeint ist. Falls ja, führe Schritte 6-12 aus.
6. ASP prüft das Transportrichtungssignal im Steuerbus (es erkennt das Signal "Lesen").
7. ASP sucht die adressierte Speicherzelle und holt die Daten.
8. ASP legt die (adressierten) Daten auf den Datenbus.
9. ASP legt das Signal "Daten bereit" auf eine bestimmte Leitung im Steuerbus.
10. CPU hat gewartet, bis das Signal "Daten bereit" erscheint und übernimmt Daten vom Bus in das Zielregister.
11. CPU legt das Signal "Daten übernommen" auf eine bestimmte Leitung im Steuerbus.
12. ASP deaktiviert daraufhin den Datenbus und das Signal "Daten bereit".
13. Sobald das Signal "Daten bereit" vom ASP weggenommen wird, deaktiviert die CPU die von ihm verwendeten Adress- und Steuerleitungen.

Solcher Ablauf aus Aktions- und Kommunikationsschritten zwischen Kommunikationspartnern heißt "Protokoll".

Der oben beschriebene Ablauf heißt **Busprotokoll**.



Komponenten eines Rechners

Busse zur Kommunikation

Kommunikation zwischen Arbeitsspeicher und CPU

Generated by Targeteam

CPU und Arbeitsspeicher arbeiten nicht synchron ⇒ Koordination notwendig.

**Beispiel:** Übertragung von Daten aus dem Arbeitsspeicher zum Prozessor (CPU).

1. CPU legt die ASP-Adresse des gewünschten Datenelements auf den Adressbus.
2. CPU legt das Signal "Lesen" auf eine bestimmte Leitung im Steuerbus.
3. CPU legt das Signal "Adresse gültig" auf eine bestimmte Leitung im Steuerbus.
4. ASP ist passiv, "hört" aber auf Adress- und Steuerbus mit.
5. ASP erkennt, dass eine Adresse gültig ist und prüft, ob er mit dieser Adresse gemeint ist. Falls ja, führe Schritte 6-12 aus.
6. ASP prüft das Transportrichtungssignal im Steuerbus (es erkennt das Signal "Lesen").
7. ASP sucht die adressierte Speicherzelle und holt die Daten.
8. ASP legt die (adressierten) Daten auf den Datenbus.
9. ASP legt das Signal "Daten bereit" auf eine bestimmte Leitung im Steuerbus.
10. CPU hat gewartet, bis das Signal "Daten bereit" erscheint und übernimmt Daten vom Bus in das Zielregister.
11. CPU legt das Signal "Daten übernommen" auf eine bestimmte Leitung im Steuerbus.
12. ASP deaktiviert daraufhin den Datenbus und das Signal "Daten bereit".
13. Sobald das Signal "Daten bereit" vom ASP weggenommen wird, deaktiviert die CPU die von ihm verwendeten Adress- und Steuerleitungen.

Solcher Ablauf aus Aktions- und Kommunikationsschritten zwischen Kommunikationspartnern heißt "Protokoll".

Der oben beschriebene Ablauf heißt **Busprotokoll**.



- Fragestellungen des Abschnitts:

- Aus welchen (Hardware-)Elementen setzt sich ein Rechner zusammen?
- Wie kommunizieren die einzelnen Komponenten eines Rechners?
- Wie sieht die Schnittstelle zwischen Hardware und Software aus (d.h. Maschinenbefehle)?
- Wie werden Zahlen, Text, Bilder, und Töne intern dargestellt?

[Aufbau eines Rechners](#)[Maschinenbefehle](#)[Befehlszyklus](#)[Interndarstellung von Information](#)

Generated by Targeteam

**Transportbefehle**

z.B. LOAD, STORE. LOAD: Transportieren von Daten vom Arbeitsspeicher in ein Register; STORE spezifiziert den umgekehrten Weg.

**Arithmetische und logische Befehle**

z.B. ADD, SUB, AND, OR, CMP

**Schiebepfeile**

z.B. SH (Shift links, rechts), ROT (Schieben im Kreis)

**Sprungbefehle**

z.B. JMP (Jump), JGT (Jump Greater Than) - (bedingte) Änderung der Ablaufreihenfolge

**Sonderbefehle**

Behandlung von Unterbrechungen (z.B. Alarm bei Division durch 0), Änderungen des Maschinenstatus, Rückmeldungen von E/A Geräten, Laden von Prozessbeschreibungen, Synchronisationsbefehle bei Speicherzugriff etc.

Generated by Targeteam

**Darstellung in Maschinensprache**

```
0100  LOAD 0118 -- lade Inhalt der Speicherzelle 118 in CPU
0102  STORE 0116
0104  LOAD 0114
0106  JUMPZERO 011a
0108  SUB 0118
010a  STORE 0114
010c  LOAD 0116
010e  ADD 0116
0110  STORE 0116
0112  JUMP 0104
0114  #2 -- Wert von x, d.h. die Zahl 2
0116  #0
0118  #1 -- Wert von y, d.h. die Zahl 1
011a  STOP
```

**Darstellung in Hochsprache**

```
y = 1;
while (x!=0) {
  x = x - 1;
  y = y + y;
}
```

**Darstellung in Maschinensprache**

```
0100  LOAD 0118 -- lade Inhalt der Speicherzelle 118 in CPU
0102  STORE 0116
0104  LOAD 0114
0106  JUMPZERO 011a
0108  SUB 0118
010a  STORE 0114
010c  LOAD 0116
010e  ADD 0116
0110  STORE 0116
0112  JUMP 0104
0114  #2 -- Wert von x, d.h. die Zahl 2
0116  #0
0118  #1 -- Wert von y, d.h. die Zahl 1
011a  STOP
```

**Darstellung in Hochsprache**

```
y = 1;
while (x!=0) {
  x = x - 1;
  y = y + y;
}
```

Java/C

2<sup>4</sup>      2·2·2·2  
2<sup>0</sup> - 1

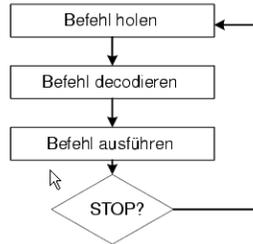


## Befehlszyklus



Ausführung eines Maschinenbefehls: festes Schema (Bereitstellung, Bereitstellen der Operanden, Befehl entschlüsseln, Ausführung). Dieses Schema heißt Befehlszyklus.

### Sequentielle Bearbeitung



kein ausführbares Programm ⇒ Ausführung von NOP ("No Operation").

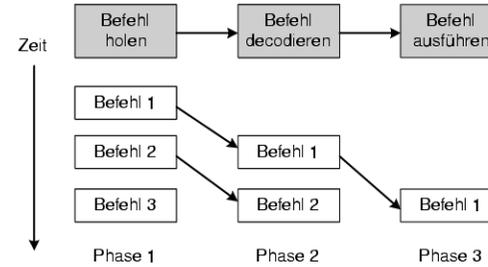
### Fließband Bearbeitung (Pipelining)



## Fließband Bearbeitung (Pipelining)



Bearbeitung jeden Befehls in mehreren Phasen. Überlappende Verarbeitung. Quasi-parallele Ausführung mehrerer Maschinenbefehle.



### Pipelining Animation

Generated by Targeteam

Generated by Targeteam



## Pipelining Animation



## Pipelining Animation



TUM MMP WS 02

### Beispiel: sequentieller Waschsalon

18:00 19:00 20:00 21:00 22:00

A B C

- Der sequentielle Waschsalon braucht 4 Stunden 30 Minuten für die 3 Waschladungen
- Wie lange würde es mit **Pipelining** dauern?

Animation läuft noch

Generated by Targeteam

TUM MMP WS 02

### Beispiel: Pipelining Waschsalon

18:00 19:00 20:00 21:00 22:00

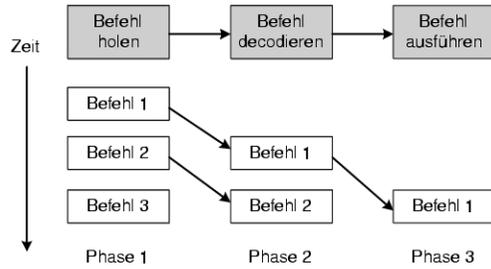
A B C

Animation läuft noch

Generated by Targeteam



Bearbeitung jeden Befehls in mehreren Phasen. Überlappende Verarbeitung. Quasi-parallele Ausführung mehrerer Maschinenbefehle.



### Pipelining Animation

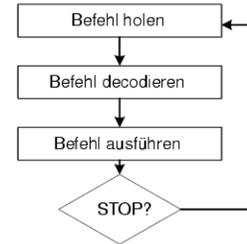


Generated by Targeteam



Ausführung eines Maschinenbefehls: festes Schema (Bereitstellung, Bereitstellen der Operanden, Befehl entschlüsseln, Ausführung). Dieses Schema heißt Befehlszyklus.

### Sequentielle Bearbeitung



kein ausführbares Programm  $\Rightarrow$  Ausführung von NOP ("No Operation").

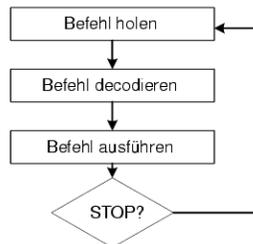
### Fließband Bearbeitung (Pipelining)

Generated by Targeteam



Ausführung eines Maschinenbefehls: festes Schema (Bereitstellung, Bereitstellen der Operanden, Befehl entschlüsseln, Ausführung). Dieses Schema heißt Befehlszyklus.

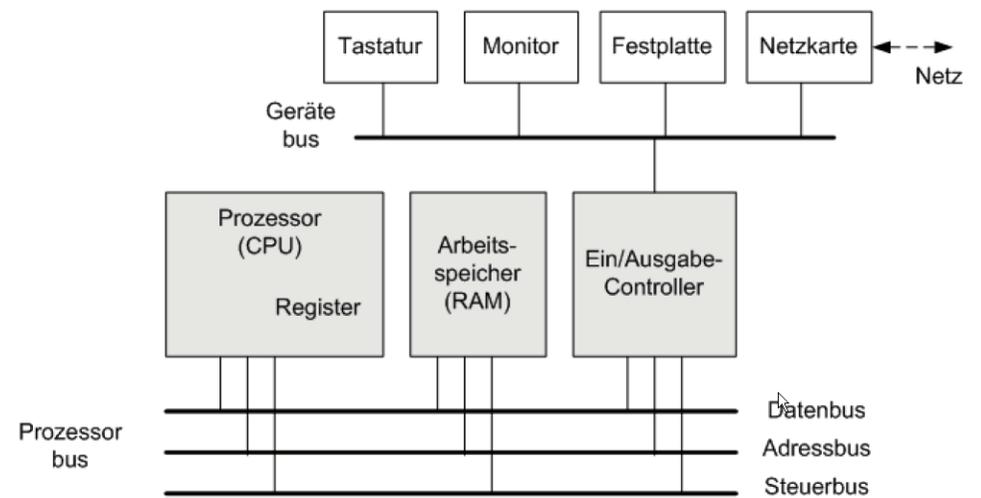
### Sequentielle Bearbeitung



kein ausführbares Programm  $\Rightarrow$  Ausführung von NOP ("No Operation").

### Fließband Bearbeitung (Pipelining)

Generated by Targeteam



### Komponenten eines Rechners

### Busse zur Kommunikation

### Kommunikation zwischen Arbeitsspeicher und CPU

Generated by Targeteam



- Fragestellungen des Abschnitts:
  - Aus welchen (Hardware-)Elementen setzt sich ein Rechner zusammen?
  - Wie kommunizieren die einzelnen Komponenten eines Rechners?
  - Wie sieht die Schnittstelle zwischen Hardware und Software aus (d.h. Maschinenbefehle)?
  - Wie werden Zahlen, Text, Bilder, und Töne intern dargestellt?

#### Aufbau eines Rechners

#### Maschinenbefehle

#### Befehlszyklus

#### Interndarstellung von Information