

## **BA-Mannheim, 3. Semester IT – Rechnertechnik (Hr. Glaser)**

---

Dies ist nun also die freundlicherweise von mir mitgetippte Fassung der Vorlesung  
**Rechnertechnik** (3. Semester) bei **Hr. Glaser** an der **BA-Mannheim**.

Ich hoffe ihr könnt damit was anfangen. ;-)

Fehler, Kritik, Anregungen und alles was euch sonst noch dazu einfällt mailt bitte an  
[himself@stephan-tost.de](mailto:himself@stephan-tost.de)

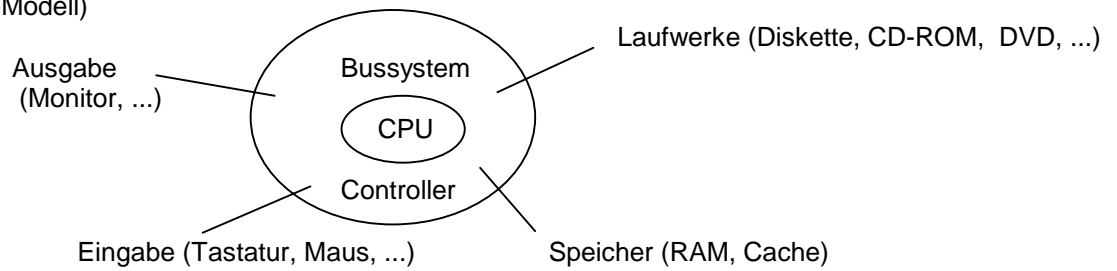
---

**12 Seiten**

Rechneraufbau	<b>2</b>
Busse	<b>2</b>
CPU-Generationen	<b>3</b>
CPU	<b>5</b>
Speicherhierarchie	<b>5</b>
Bootvorgang	<b>5</b>
Mikroprozessor	<b>6/7</b>
Aufbau 8086	<b>8</b>
Systembusschnittstelle	<b>9</b>
Schreib-Lese-Speicher – Matrix	<b>9</b>
Phasenunterteilung der Befehlsabarbeitung	<b>10</b>
RESET - Schaltung	<b>11</b>
Physikalischer Speicher- und Portzugriff	<b>11</b>
Prozessor-Hauptspeicher	<b>12</b>
Prozessor-Ports	<b>12</b>

**Rechneraufbau**

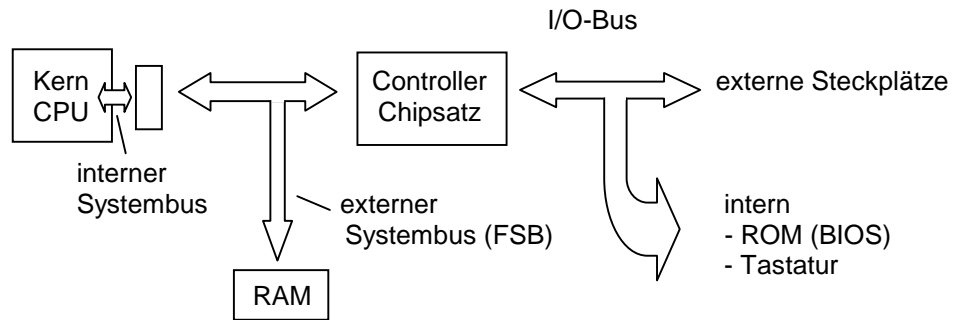
(Neumann-Modell)



**Bussysteme**

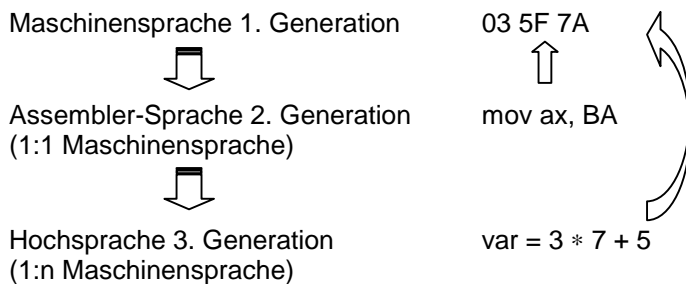
- IDE (16 Bit)
- SCSI
- EISA (32 Bit)
- PCI (32 – 64 Bit)
- USB
- AGP
- ISA (IBM, 8 – 16 Bit)
- XT, AT,
- VLB (Vesa Local Bus, 32 Bit)
- Firewire
- Systembus (intern / extern)

**Motherboard**



**Busaufbau**

- Adressbus – zur Adressierung von Befehlen
- Steuerbus
- Datenbus



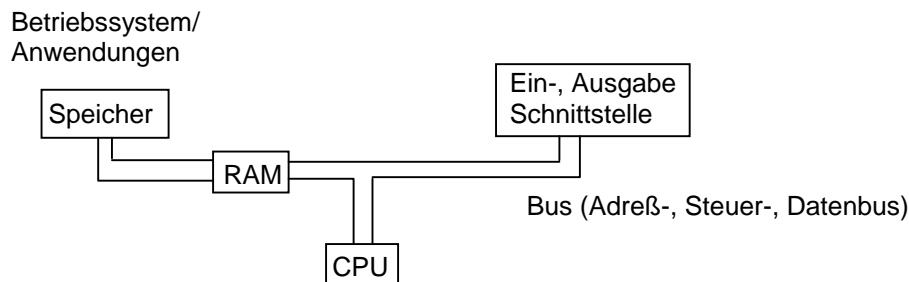
Software



Firmware (meist in HW integrierte Befehlsdaten zur Steuerung eines Gerätes)



Hardware



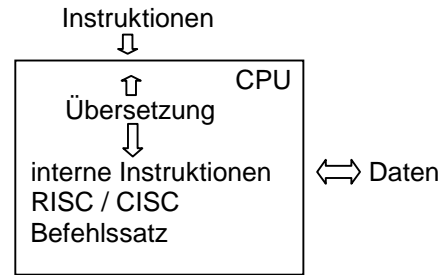
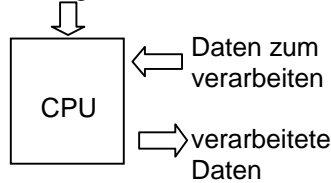
### **CPU Generationen**

0. Generation
  - a. kam nicht wirklich zum Einsatz
1. Generation
  - a. 8086 (1978)
    - i. erste 16 bit CPU auf dem Markt mit 16 bit breitem Datenbus
    - ii. 4,77 MHz, Real Mode
    - iii. Hardware (16 bit) dafür zu teuer → schlechte Verkaufszahlen
  - b. 8088 (1979)
    - i. auch noch für 8 bit Datenbus auf Motherboard
  - c. 8087 Coprozessor
2. Generation
  - a. 80286 (1982)
    - i. 16 bit
    - ii. höhere Taktfrequenz (6 bis 12 MHz)
    - iii. optimierte Befehlsabarbeitung
    - iv. Protected-Mode - Betrieb möglich (24 bit virtueller Adressraum)
3. Generation
  - a. 80386
    - i. 1985
    - ii. erste 32 bit CPU
    - iii. konnte bis zu 4 Gb Speicher ansprechen
    - iv. 16 bis 33 MHz (Clone 40 MHz)
    - v. virtueller 8086 Modus
  - b. 386 SX
    - i. ‚billige Variante‘
    - ii. 24 bit anstatt 32 bit Adressierung
    - iii. max. 16Mb RAM
  - c. 387 Coprozessor
4. Generation
  - a. 80486 (1989)
    - i. fast doppelt so schnell (25 bis 133 MHz)
    - ii. erste interne Taktverdopplung
    - iii. effizienterer x86-Befehlssatz
    - iv. noch 32 bit CPU
    - v. eingebauter Coprozessor
    - vi. dazu 8Kb Level 1-Cache
5. Generation
  - a. Pentium (Classic), 1993
    - i. Namensänderung wg. Patentrecht
    - ii. superskalar (d. h. mehr als eine Anweisung pro Taktzyklus)
    - iii. große Veränderung an externem Systembus
      1. 64 bit Datenbus
      2. Geschwindigkeit: 60 / 66 MHz
    - iv. P90, P120, P133, P150, P166, P200

- b. erweiterte 5. Generation
  - i. Pentium MMX (1997)
    - 1. 166 bis 233 MHz
    - 2. neue zusätzliche Befehle für Multimedia (MMX)
    - 3. 57 Integer Befehle, vier Datentypen, acht 64 bit Register
    - 4. 32 Kb L1-Cache
    - 5. neue Motherboards → Dual Voltage (interne Spannung, I/O Spannung)
- 6. Generation
  - a. Pentium Pro (1995)
    - i. erster RISC-Prozessor
    - ii. für 32 bit Anwendungen wie WinNT, OS/2 optimiert
    - iii. L2-Cache integriert (256 / 512 Kb) synchron zur CPU (64 bit Back Side Bus)
    - iv. 4 Pipelines für simultane Befehlsabarbeitung
    - v. patentiertes Protokoll (auf Sockel und Chipsatz)
    - vi. multiple branch prediction → Versuch, nächste Anweisung vorherzusehen
    - vii. speculative execution → Versuch, Ergebnis von Berechnungen vorherzusagen
  - b. erweiterte 6. Generation
    - i. 1. Generation Pentium II (1997)
      - 1. L2-Cache wurde mit halbem Takt betrieben (nicht auf der CPU) → billiger
      - 2. Slot 1 (Sockel)
      - 3. MMX-Befehle (wie Pentium MMX)
      - 4. verbesserte Ausführung von 16 bit Programmen
      - 5. doppelt so großer L1-Cache
      - 6. höherer Takt (300 statt 233 MHz)
    - ii. 2. Generation Pentium II (1998)
      - 1. Version
        - a. 0,25 anstatt 0,35 Mikron Technik (Chipdicke)
        - b. 50% Stromersparnis
        - c. Spannung von 2,8 auf 2V intern
      - 2. Version
        - a. Systembus mit 100 MHz
        - b. Taktmultiplikator 3,5 / 4, 4,5
    - iii. Celeron (1998))
      - 1. wegen schlechter Vermarktung des teuren PII
      - 2. Pentium II ohne L2-Cache
      - 3. erster Celeron für Slot 1 mit 66 MHz Systemtakt (CPU-Takt: 266 / 300MHz)
    - iv. Celeron (Ende 1998)
      - 1. wieder L2-Cache integriert (128 Kb) und läuft mit CPU-Takt
      - 2. ab 300 MHz, mit 66 MHz Systemtakt
      - 3. in weiteren Versionen 0,18 Mikron Technik
      - 4. seit 1999 weitere Baureihe für PGA370 Sockel (parallel zu Sockel 7)
    - v. Pentium III (1999)
      - 1. 1999
        - a. verbesserter MMX2-Befehlssatz (70 neue Graphikfunktionen)
        - b. nutzt acht 128 bit breite Register
        - c. 500 MHz, Slot 1
      - 2. Juli 1999
        - a. 600 MHz
        - b. 100 MHz Systembus
        - c. mit Aluminium-Technik aufgebaut
        - d. Kernspannung 1,65 V
        - e. 256 Kb L2-Cache
        - f. Bus zwischen L2-Cache und CPU hat 256 bit breiten Datenbus
- 7. Generation
  - a. Pentium 4 (2001)

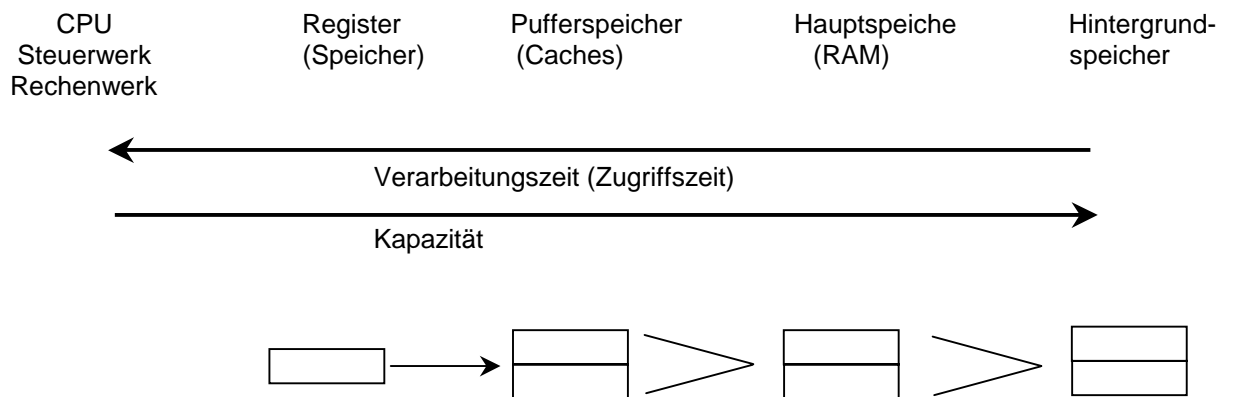
**Was macht die CPU**

Befehle zur Verarbeitung der Daten



CISC – Complex Instruction Set Computing (es wird mit komplexem Befehlssatz gerechnet)  
 RISC – Reduced Instruction Set Computing (kleiner Befehlssatz)

**Mehrstufige Speicherhierarchie**



Einlagerung der Daten in prozessornaher Speicher wird durch Zugriff des Prozessors, durch die Hardware (CPU-Register ↔ Cache, RAM) und auch durch Software (CPU ↔ Hintergrundspeicher) organisiert.

**Speicherhierarchie mit typischen Zugriffszeiten**

	Zugriffszeit	Kapazität
Prozessorregister	<2,5 ns	256 – 1024 Bytes
Primärer Cache	2,5 – 10 ns	1 – 128 Kbyte
Sekundärer Cache	5 – 50 ns	256 KB – 4 MB
Hauptspeicher (RAM)	50 – 500 ns	bis 1 GB
Hintergrundspeicher	5 – 15 ms	bis 100 GB
Archivspeicher	>>50 ms	bis mehrere TBV

**Bootvorgang**

1. Schritt

- StartUp-Programm ROM-BIOS
  - o ermittelt die Hardwareressourcen und –anforderungen
  - o liest DIP-Schalter (Microschalter) und Jumper
  - o liest im CMOS-Chip gespeicherte Konfigurationsinformationen und vergleicht sie mit den vorhandenen
  - o Einige Hardwaregeräte verfügen über eigene BIOSse
    - vom Startup-BIOS werden ihnen Speicheradressen zur Verfügung gestellt
    - StartUp-BIOS enthält Ressourcenanfrage von diesen Geräten und weist ihnen Ressourcen zu

**2. Schritt**

- Betriebssystem suchen und laden
  - o sucht nach Betriebssystem (Konfigurationsdaten auf CMOS-Chip)
  - o wechselt auf entsprechendes Laufwerk
  - o liest die ersten Dateien ein zw. schreibt sie in den Hauptspeicher
  - o übergibt die Kontrolle an das Betriebssystem

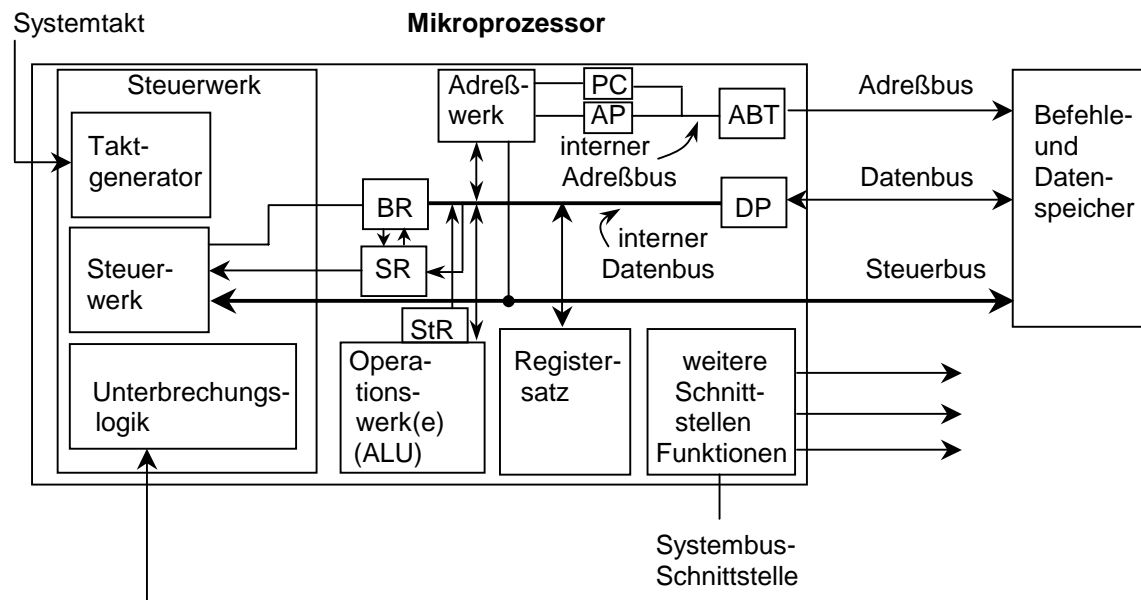
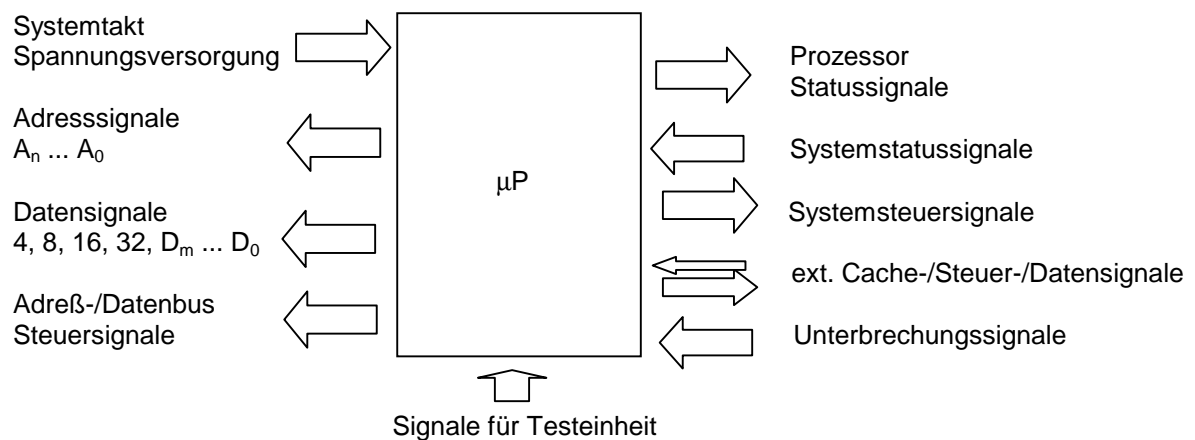
**3. Schritt**

- Betriebssystem konfiguriert das System
  - o Betriebssystem überprüft einige bereits vom StartUp-BIOS geprüfte Dinge (z.B. vorhandene Hardware)
  - o lädt Software zur Steuerung der Maus, Tastatur, CD-Rom, ...

**4. Schritt**

- Anwender führt Anwendungssoftware aus

**Anschlussbelegung eines Mikroprozessors**



**ABT** - Adressbusspeicher

**DP** - Datenpuffer

**Systembus-Schnittstelle**

- stellt Zwischenspeicherregister (Puffer) für kurzzeitige Aufbewahrung jedweder Daten auf / in den Prozessor und dient der zeitlichen Anpassung zwischen schnellem Prozessor und langsamer externer Peripherie.

### **Registersatz**

- Register = Speicherzellen mit minimaler Zugriffszeit
- oft besitzen einige Register spezielle Funktionen, die durch Steuerleitungen aktiviert werden (z.B. Rücksetzen, Inkrementieren, Dekrementieren, bitwises Verschieben)
- spezielle Register:
  - o Datenregister
    - dienen zur Zwischenspeicherung der Operanden bei allen arithmetischen und logischen Operationen (z.B. AX-, BX-, CX-, DX-Register)
  - o Adressregister
    - enthalten die Adressen oder Teile davon, die zur Auswahl eines Operanden oder eines Befehls im Speicher herangezogen werden (z.B. Basisregister, Indexregister)
- einfache Prozessoren enthalten nur wenige Register (bis zu acht)
- RISC-Prozessoren bis zu einigen hundert Register

### **Steuerwerk (Control Unit)**

- kontrolliert die Steuersignale zu den Komponenten des Systems (int. / ext.) über den Steuerbus
- Sorgt für den zeitgerechten (mit Hilfe des Taktgenerators) Ablauf aller Vorgänge

### **BR (Befehlsregister)**

- gehört zum Steuerwerk
- speichert kurzzeitig den augenblicklich ausgeführten Befehl

### **Unterbrechungslogik**

- Bearbeitung aller anstehenden Unterbrechungsanforderungen

### **SR (Steuerregister)**

### **Operationswerke**

- führt im einfachsten Fall die vom Steuerwerk verlangten arithmetischen und logischen Operationen aus
- komplexe Prozessoren besitzen mehrere Rechenwerke (evtl. mit spez. Aufgaben z.B. FPU [Floating Point Unit])
- häufig benutzte Operanden werden im Registersatz zwischengespeichert

### **Statusregister**

- auch Zustandsregister genannt
- besteht aus einzelnen logischen, unabhängigen Bits (auch Marker oder Flag genannt)
- Die Flags werden aus den Operanden und dem Ergebnis einer Operation abgeleitet und reflektieren damit den Zustand des Rechenwerkes nach dieser Operation.
- Sie können aber auch durch Befehle beeinflusst werden bzw. dienen zur Beeinflussung eines ablaufenden Maschinenprogrammes.

### **Adresswerk(e)**

- berechnet nach den Vorschriften des Steuerwerks die Adresse eines Operanden oder eines Befehls im Speicher (Prozessoren mit Harvard-Architektur verfügen über mehrere Adresswerke für Befehle, Daten, ..)

### **PC (Programmzähler)**

- oder Befehlszähler, enthält stets die Adresse der Speicherzelle in der der nächste auszuführende Befehl liegt (IP – Instruction Pointer)

### **AP (Adresspuffer)**

- Die für die Ausführung eines Maschinenbefehls benötigten Operanden werden durch das Adresspufferregister selektiert (wird nach bestimmten fest vorgegebenen Verfahren berechnet)

### **Interner Bus**

- Komponenten im Mikroprozessor sind durch ein System von internen Bussen miteinander verbunden
- es gibt keine Unterscheidung zwischen Adreß- und Datenbus möglich, da über den internen Datenbus beide Informationsarten übertragen werden

**Aufbau des 8086**

**BUI (Bus Interface Unit)**

- leitet die Ein- und Ausgabe der vom Steuerwerk angeforderten Daten
- Vermittler zwischen Steuerwerk und Bussystem
- auch Bus-Schnittstellen-System genannt

**EU (Execution Unit)**

- Verarbeitungseinheit
- Allzweckregister (16 Bit / 8 Bit-Register)
  - o AX (AH, AL), auch Akkumulator genannt
    - AX – 16 Bit, H für HIGH (obere 8 Bit), L für LOW (untere 8 Bit)
    - dient hauptsächlich zu arithmetischen Operationen
  - o BX (BH, BL), Basisregister
    - zum direkten Ansprechen des Arbeitsspeichersinhaltes
    - z.B. Tabellen- und Stringverarbeitung
  - o CX (CH, CL), Zählregister
    - speziell für Schleifenoperationen
  - o DX (DH, DL), Datenregister
    - Erweiterung des Basisregister um mehr als 16 Bit zur Verfügung zu haben
    - z.B. für 32 Bit Multiplikation und Division
- Segmentregister
  - o CS – Codesegment
    - steht in Verbindung mit dem [IP]
    - dort sind die Maschinenbefehle abgelegt
    - IP
      - Befehlszeiger [CS]
      - adressiert in Verbindung mit dem Codesegment (CS:IP) den nächsten auszuführenden Befehl
  - o DS – Datensegment
    - dort sind Variablen, Felder, Tabellen abgelegt
    - SI
      - Quell Index [DS]
      - Unterstützung als Offsetregister für die Adressierung in Datei- und Extrasegment
  - o ES – Extrasegment
    - Zusatz für das Datensegment, wenn es nicht ausreicht
    - für Stringoperationen
    - DI
      - Zielindex [ES]
      - siehe SI
  - o SS – Stacksegment
    - interner Zwischenspeicher (von Rücksprungadressen bei Unterprogrammen, kurzzeitige Parameterübergabe)
    - BP
      - Basis Pointer [SS]
      - Zugriff auf das Stacksegment als Index
    - SP
      - Stackzeiger
      - zeigt mit dem Stacksegment auf die aktuelle Position im Stacksegment
      - sollte nicht von Programmierern verwendet werden, da es vom System mitbenutzt wird
- Flagregister
  - o Statusflags 

	Bit-Nr.
▪ CF (Carry Flag [Übertragsflag])	1
▪ PF (Parity Flag [Paritätsflag])	2
▪ AF (Auxiliary Flag [Hilfübertragsflag])	4
▪ ZF (Zero Flag [Nullflag])	6
▪ OF (Overflow Flag [Überlaufflag])	11
  - o Kontrollflags 

▪ TF (Trap Flag [Einzelschrittflag])	8
▪ IF (Interrupt Flag [Unterbrechungsflag])	9

**Allgemein**

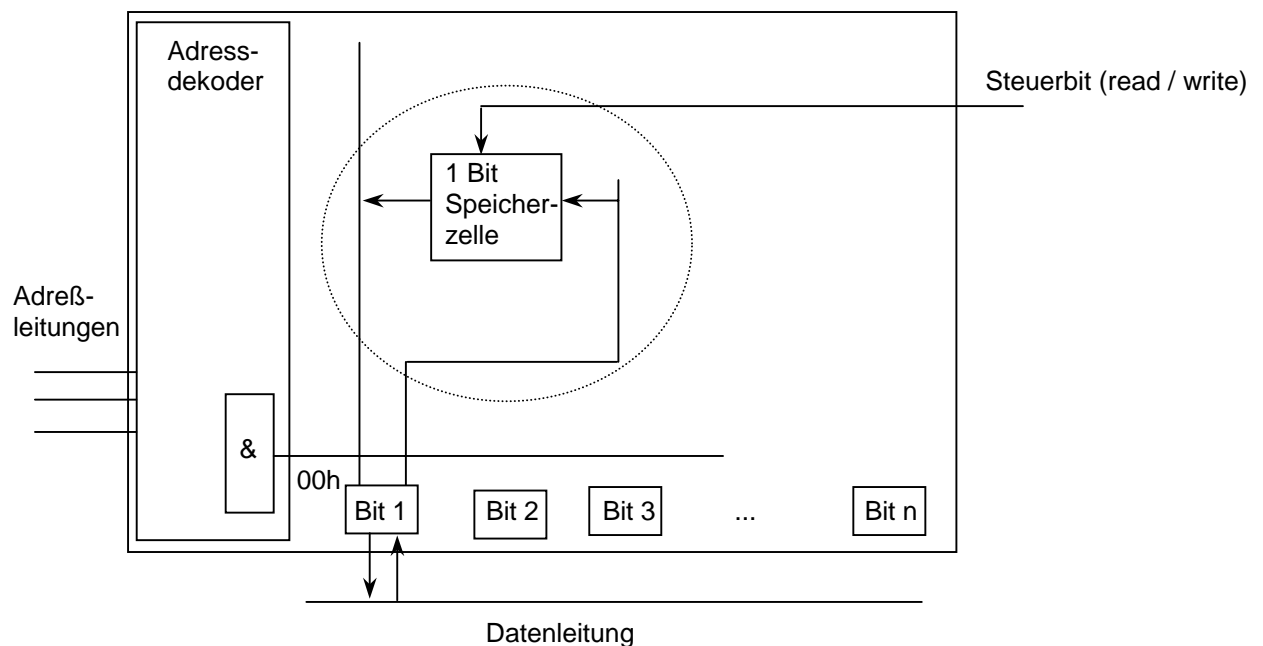
Enthält die Anfangsadresse eines maximal 64kByte großen Segmentes, welches mittels eines Offsets durch das dazugehörige Register [ ] adressiert wird.

**Die Systembus Schnittstelle**

**Allgemein**

- stellt Verbindung des Mikroprozessors zu seinen Komponenten (extern) her
- übernimmt die (logische) Funktion der Zwischenspeicherung von Adressen und Daten
- übernimmt auch die elektrische Anpassung von internen und externen Signalpegeln (durch Treiber)
- Realisierung der Busankopplung
  - o Tristate - Gatterform
    - ermöglichen es, mehrere Gatterausgänge auf eine gemeinsame Leitung zusammen zuschalten
    - für jede Bit-Leitung muß ein solcher Treiber vorhanden sein
  - o Bidirektionaler Datenbustreiber
    - Steuereingang zum Sperren zw. zum Freischalten der Ein- und Ausgänge
    - DIR-Signal zur Richtungsumschaltung
  - o Bidirektionaler Datenbustreiber mit Puffer - Flip-Flop
    - die Übernahme eines neuen Datums in das Flip-Flop wird durch ein spezielles Taktsignal (T) erzwungen
    - G aktiviert bzw. deaktiviert die Treiber (G = 0 → alle Tristate-Gatter hochohmig)
    - DIR selektiert die gewünschte Richtung
    - zwei zusätzliche Steuersignale ermöglichen die Deaktivierung der Ausgangsgatter
    - dadurch kann der Flip-Flop – Inhalt verändert werden, ohne dass der Ausgang auf die Leitung gelegt wird

**Schreib – Lese – Speicher – Matrix**



### **Phasenunterteilung der Befehlsabarbeitung**

#### **Allgemein**

Ein Programm für den Mikroprozessor besteht aus einer Folge von Maschinenbefehlen (Makrobefehle). Diese werden vom Steuerwerk (Adresswerk) sukzessiv in den Prozessor geladen, dekodiert und ausgeführt.

#### **Holphase**

- Maschinenbefehl wird vom Steuerwerk ins Befehlsregister (BR) geladen  
Da einige Prozessoren Operationscodes besitzen die sich in der Anzahl der benutzten Wörter unterscheiden, werden häufig Blocks aus mehreren Registern nachgeladen.

#### **Prefetch-Phase**

- Maßnahmen zur Steigerung der Verarbeitungsgeschwindigkeit  
Während der aktuelle Befehl dekodiert (bearbeitet) wird, werden schon einer oder mehrere Folgebefehle aus dem Speicher geladen. (FIFO - Speicher)

#### **Dekodierphase**

Durch den Befehlsdekodierer wird der Befehl entschlüsselt (interpretiert).

- um welche Befehlsgruppe handelt es sich
- welche Prozessorkomponenten sind für die Ausführung zuständig
- werden Operanden benötigt, wenn ja: welche

#### **Ausführungsphase**

Überwacht und angeleitet durch die Steuerlogik wird der Befehl durch die von Dekoder bestimmten Prozessorkomponenten ausgeführt.

Dazu generiert die Steuerlogik die erforderlichen Steuersignale. Durch Meldesignale können die Komponenten dem Steuerwerk Zustandsinformationen liefern und dadurch Einfluß auf die Befehlsausführung nehmen.

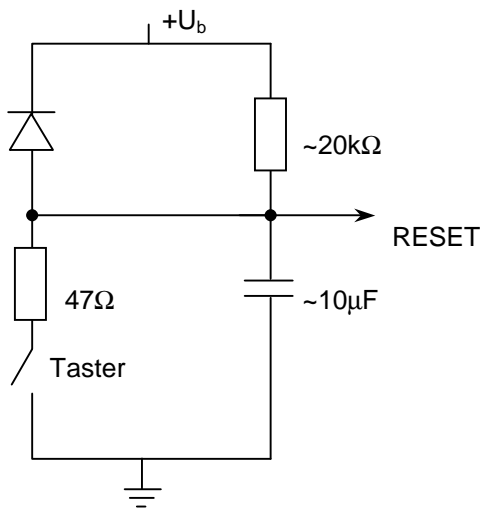
#### **Steuerregister**

- Ablage bestimmter Informationen
- z.B. bestimmte Vorgaben
  - o Unterbrechungsanforderungen zulassen
  - o virtuelle Speicherverwaltung aktiviert
  - o arbeitet Prozessor im Benutzer- oder Betriebssystemmodus

#### **Rückschreibphase**

Am Ende werden die berechneten Ergebnisse in die Register oder in den Arbeitsspeicher zurückgeschrieben.

**Schaltung zum Zurücksetzen des Prozessors (RESET)**



kombinierte Schaltung zum Zurücksetzen des Prozessors

- automatisch beim Einschalten der Betriebsspannung
- manuell über Taster

Funktionen beim Start

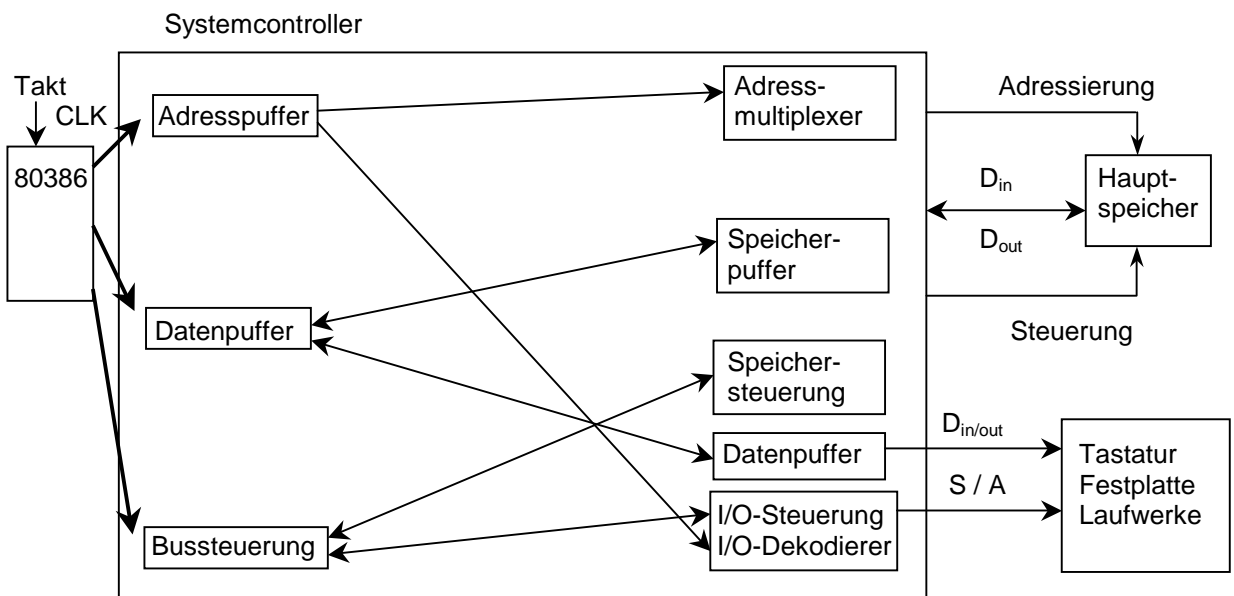
- nach Einschalten der Spannungsversorgung steigt die Betriebsspannung  $+U_b$  sehr rasch dem Endwert entgegen (Verzögerung durch Aufladen aller Kapazitäten des Systems)
- nach Erreichen des Endwertes  $+U_b$  beginnt der Oszillator mit dem Systemtakt
- mit der Widerstand-Kondensator – Kombination vorgegebene Zeitkonstante  $\tau = R * C$  (hier:  $\tau = 200ms$ ) wird der Kondensator allmählich aufgeladen
- erst wenn der Kondensator einen bestimmten Schwellenwert  $U_s$  überschreitet (am Ausgang RESET) wird mit der nächsten Schwingung des Systemtakts der Programmzähler mit der Adresse des ersten Maschinenbefehls geladen und dadurch die Programmausführung gestartet

Rücksetzen

Durch den Zurücksetztaster kann sich der Kondensator gegen Masse entladen (Reihenwiderstand für Entladungsbegrenzung).

Durch Abschalten der Betriebsspannung ( $U_b = 0$ ) sorgt die Diode für ein schnelles Entladen des Kondensators.

**Der physikalische Speicher- und Portzugriff**



**Der Pfad zwischen Prozessor und Hauptspeicher**

Der Prozessor greift normalerweise nicht unmittelbar auf den Speicher zu. Stattdessen befindet sich zwischen CPU und dem RAM ein Buscontroller zur Erzeugung der Steuersignale für den Bus und verschiedene Puffer zur Zwischenspeicherung und Verstärkung der Daten- und Adresssignale.

Die Speichersteuerung kontrolliert den Hauptspeicher, sodaß dieser auch die korrekten Daten liefert oder an die richtige Stelle schreibt. Die in der Abbildung als individuelle Komponenten dargestellten Puffer- und Steuerbausteine sind heute Teil eines einzigen Systemcontrollers.

**Der Pfad zwischen Prozessor und Ports**

Der 80386 greift nicht unmittelbar auf den I/O-Adressbereich zu sondern (wie beim Hauptspeicher) nur unmittelbar über den Buscontroller. Der Buscontroller erzeugt die Steuersignale für den Bus und die verschiedenen Puffer zur Zwischenspeicherung und Verstärkung der Daten- und Adresssignale. Auch diese Zwischenschaltkreise sind heute in einem einzigen Systemcontroller integriert.

