

# Rechnerorganisation

## Studiengang PSC

- Dipl.-Inf., Dipl.-Ing. (FH) Michael Wilhelm
- Hochschule Harz
- FB Automatisierung und Informatik
- [mwilhelm@hs-harz.de](mailto:mwilhelm@hs-harz.de)
- Raum 2.202
- Tel. 03943 / 659 338

# Lernziele

## ■ Überblick

- Rechner: Hardware / Software

## ■ Hardware

- Motherboard, CPU
- Bildschirm (CRT, TFT), Nullpunkt
- Netzwerk, Lan, Wlan, TCP/IP, IP, ipv4, ipv6

## ■ BS-Software

- Windows BS, Registry
- Explorer
- Backup, XCOPY, SUBST, CD, MD, DIR
- Partitionen

## ■ Virtualisierung

- VMWare, MS VMWare

# Inhalt

## **1. Hardware**

## 2. Betriebssystem Software

## 3. Anwendungssoftware

# 1. Komponenten eines Computers (Hardware)



# Komponenten eines Computers

- Hardware
- BIOS (Basic Input Output System)
- Betriebssystem (im engeren Sinne)
- Betriebssystem (im weiteren Sinne, Unix)
- Compiler
- Anwendungssoftware
- Zusatzsoftware

# Weitere Begriffe

- Computer Architekturen
- Interrupts (Hardware, IRQ, Interrupt-Request)
- Interrupts (Software-Aufruf)
- Speichertechniken
- Ports (wichtig beim Geräten und Servern)
- I/O-Anschlüsse für Festplatten
  - LPT, COM
  - IDE, EIDE, PATA, SATA, SCSI
  - USB2, USB3
  - Light-Peak (Apple)
- Lan 1394, Firewire (Netzwerkverbindung)
- Protokolle (TCPIP, ipv4, ipv6)

# Monitor

- **Ausgabegerät**
  - PC, Workstation
  - Thin-Client, Citrix, Client-Server Betrieb
- **Nur in Verbindung mit einer VGA-Karte**
  - D-Sub, Analog (blau)
  - DVI, Digital (weiß)
  - Umwandlung ?
  - HDMI
  - **DisplayPort** – der neue Videoanschluss (2560 x 1600 mit 10 Bit Farbtiefe)
  - **Thunderbolt** – Nachfolger DisplayPort (serieller Anschluss mehrerer Monitore)
- **Technische Typen**
  - CRT = Cathode Ray Tube
  - TFT-Monitore
- **Frequenzen**
  - Vertikal und Horizontal
  - Übertragungsbandbreite

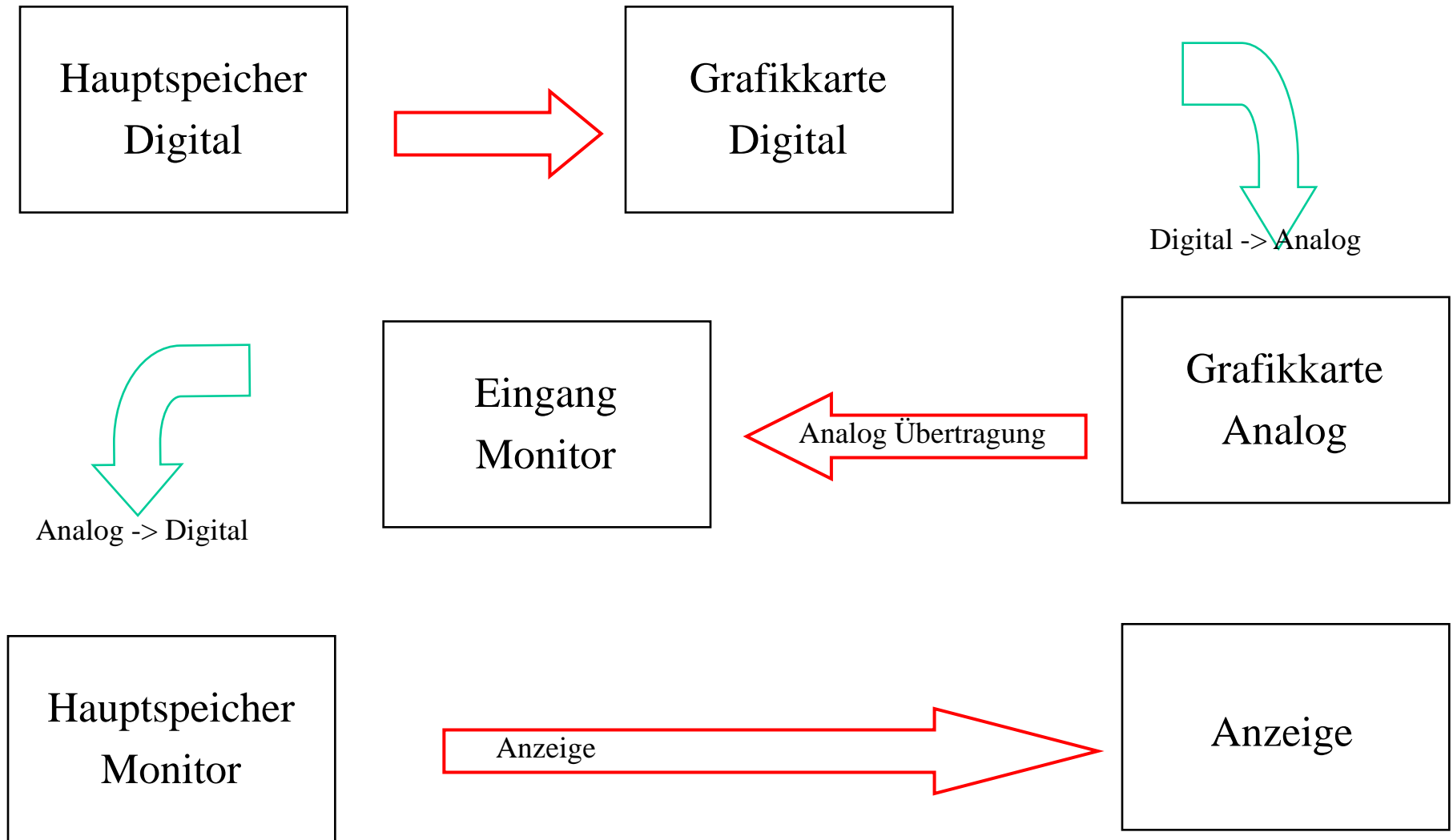
# Monitor

## ■ Frequenzen

- Beispieldaten:
  - 1024x768 Bildschirmauflösung
  - 48 Zeilen Auslastung, liegen zusätzlich zwischen den Zeilen
  - 70 mal werden die Bilder wiederholt
- Vertikalfrequenz
  - Anzahl der dargestellten Bilder pro Sekunde: 70 Hz
- Horizontalfrequenz
  - zeigt die aktuelle Leistungsfähigkeit bzgl. der Auflösung und der Vertikalfrequenz an  $(1024+48)*70 = 75040 \text{ Hz} = 75 \text{ kHz}$
- Leistungsfähigkeit des Monitors
  - Ist die Obergrenze für die Horizontalfrequenz
  - Vorgabe Beispiel:  $97 \text{ kHz} \Rightarrow 97000/(1024+48) \Rightarrow \text{maximale mögliche Vertikalfrequenz} = 90,48 \text{ Hz}$



# VGA-Karte (Video Graphic-Adapter)



# Monitor

## ■ Auflösungen

- CGA, Hercules, EGA, VGA, SVGA, ...
- 320×200 bis 1920 ×1200 und mehr
- 4×3, 16×10, 16×9

## ■ Farben

- Schwarz-weiß
- 8 Bit
- 16 Bit
- 32 Bit

## ■ Speicherbedarf

- Pro Pixel
- Pro Bild

## Beispiel

Farbe: schwarz-weiß

1 Bit / Pixel

Auflösung: 1024x768

Speicherbedarf:

$1024 \times 768 \times 1 \text{ Bit} =$

$786432 \text{ Bits} =$

$98304 \text{ Bytes} = 96 \text{ kByte}$

Bei einer Farbtiefe von 32 Bit:

Speicherbedarf:  $96 \text{ kB} * 32 =$   
 $3072 \text{ kB} = 3 \text{ MB}$

# Monitor / Fernseher

- CRT-Monitor
- TFT (thin film transistor)
- Plasma
- LCD
- LED

**LaCie 730 LCD Monitor**  
**30 Zoll**

**Auflösung: 2560 x 1600**

**Preis: ca. 3.600,00 EUR**



# Monitor

- CRT-Monitor
- TFT (thin film transistor)
- Plasma





# Darstellungen von Monitoren

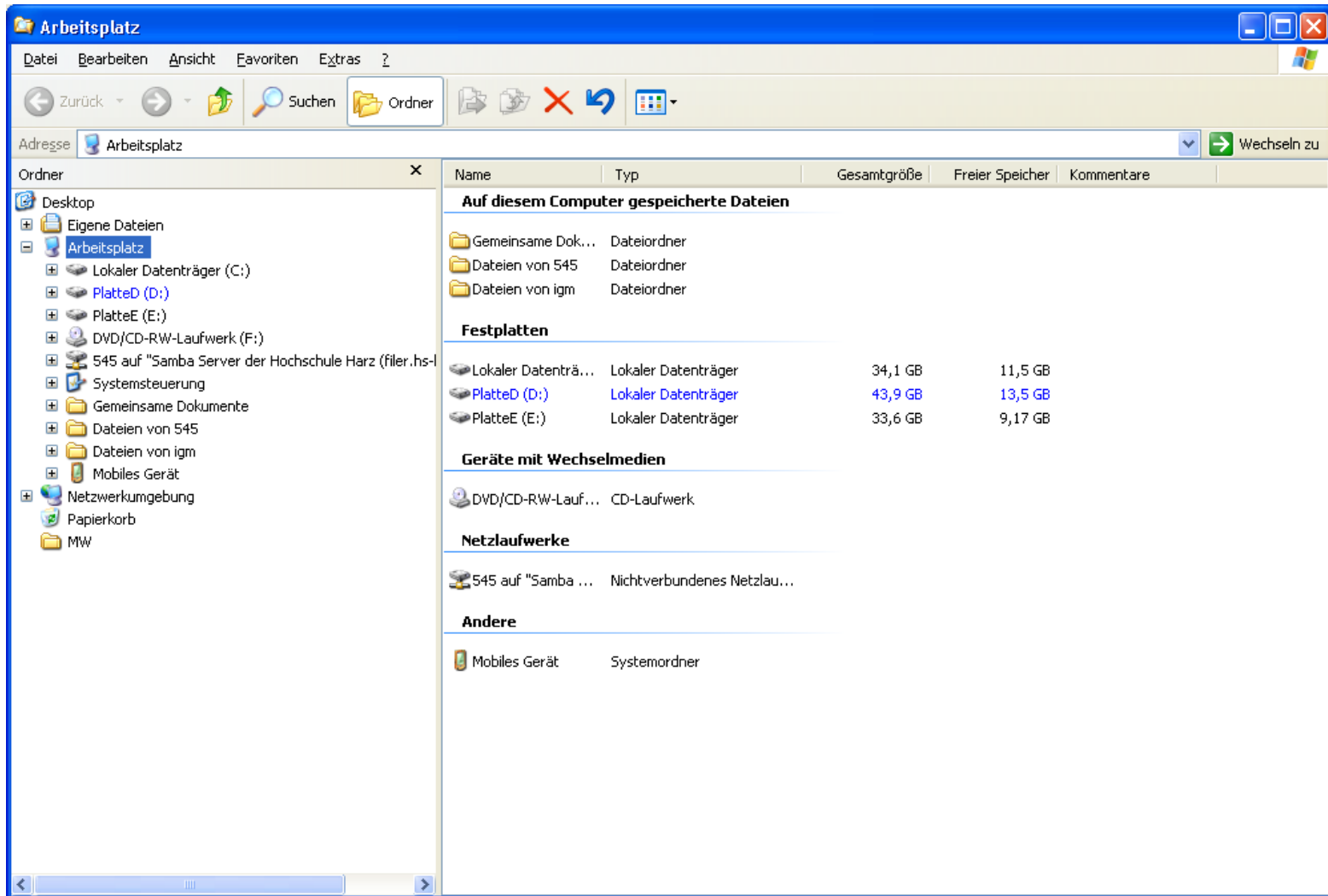


```
C:\WINDOWS\system32\cmd.exe
13.03.2007 09:49 <DIR>
15.04.2006 10:24 <DIR>
23.11.2006 12:45 67 .iboconsole
18.04.2006 21:00 <DIR> .junit.session
23.05.2006 09:03 995 .netbeans
23.05.2006 09:04 1.038 1.txt
23.05.2006 09:06 <DIR> 44
11.10.2006 12:42 <DIR> Anwendungsdaten
12.10.2006 19:37 <DIR> Application Data
19.03.2007 22:04 <DIR> Desktop
15.04.2006 00:18 <DIR> Druckumgebung
26.02.2007 19:16 <DIR> Eigene Dateien
14.04.2006 23:37 <DIR> Favoriten
25.10.2006 20:02 <DIR> Netzwerkumgebung
28.06.2006 08:33 600 PUTTY.RND
14.10.2006 10:23 <DIR> SendTo
15.04.2006 15:10 <DIR> Startmenü
18.04.2006 20:33 <DIR> UserData
14.04.2006 23:25 <DIR> Vorlagen
15.04.2006 03:22 <DIR> WINDOWS
17.10.2006 13:34 <DIR> workspace
4 Datei(en) 2.700 Bytes
18 Verzeichnis(se), 12.447.854.592 Bytes frei
C:\Dokumente und Einstellungen\545>
```

# Darstellungen von Monitoren



# Darstellungen von Monitoren





# Ausgabe einer Grafik

- Umschalten in den Grafikmodus
  - Bestimmen der Abmessungen
  - Nullpunkt X-Achse
  - Nullpunkt Y-Achse
  - Stift erzeugen
  - Farbe erzeugen
  - Zeichnen
  - Objekte freigeben
- 
- Rest im 3. Semester

# Benutzeroberflächen: Zeichnen in Java



## Benötigte Elemente:

- Linien (Farbe, Strichdicke, Strichart)
- Zeichenblatt (Panel, JFrame)

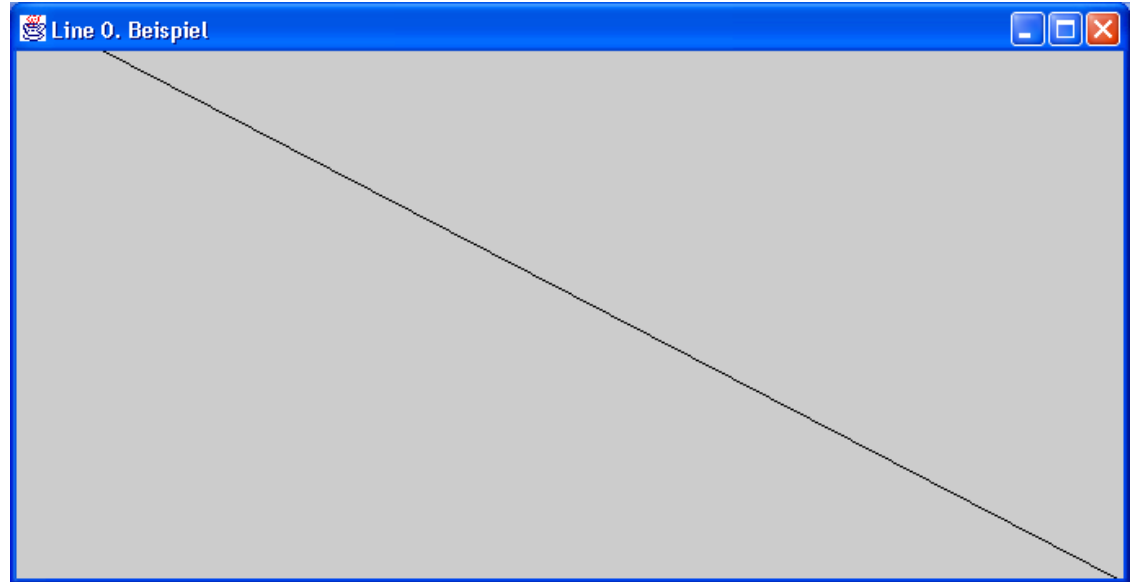
# Benutzeroberflächen: Zeichnen in Java

<b>Methode</b>	<b>Beschreibung</b>
<code>clearRect()</code>	Löscht ein Gebiet der Zeichenfläche
<code>copyArea()</code>	kopiert ein Rechteck innerhalb der Zeichenfläche
<code>drawArc()</code>	zeichnet einen Bogen
<code>drawLine()</code>	zeichnet eine Gerade
<code>drawOval()</code>	zeichnet eine Ellipse
<code>drawPolygon</code>	zeichnet ein Polygon (Array)
<code>drawRect()</code>	zeichnet ein Rechteck
<code>drawRoundRect()</code>	zeichnet ein Rechteck mit abgerundeten Ecken
<code>fillArc()</code>	füllt einen Bogen mit Farbe
<code>fillOval()</code>	füllt ein Oval mit Farbe
<code>fillPolygon()</code>	füllt ein Polygon mit Farbe
<code>fillRect()</code>	füllt ein Rechteck mit Farbe
<code>getColor()</code>	gibt die aktuelle Zeichnungsfarbe aus
<code>getFont()</code>	gibt den aktuellen Zeichensatz aus
<code>getFontMetrics()</code>	gibt die aktuellen Parameter des Zeichensatzes aus
<code>setColor()</code>	legt die Zeichnungsfarbe fest
<code>setFont()</code>	setzt den aktuellen Zeichensatz

# Zeichnen in Java: Linien

## Linien zeichnen

```
public void paint (Graphics g) {  
    int width,height;  
    width = this.getWidth();  
    height = this.getHeight();  
    g.drawLine(0,0,width,height);  
}
```

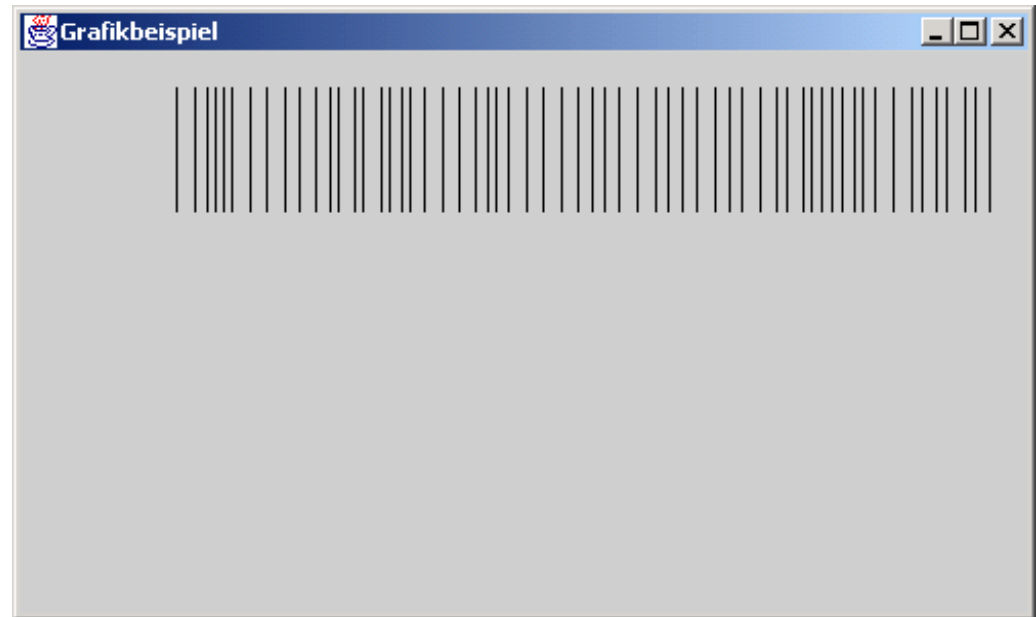


- Die Variable g bezeichnet man auch als Grafikkontext
- Vergrößern des Fensters

# Zeichnen in Java: Linien

## Linien zeichnen

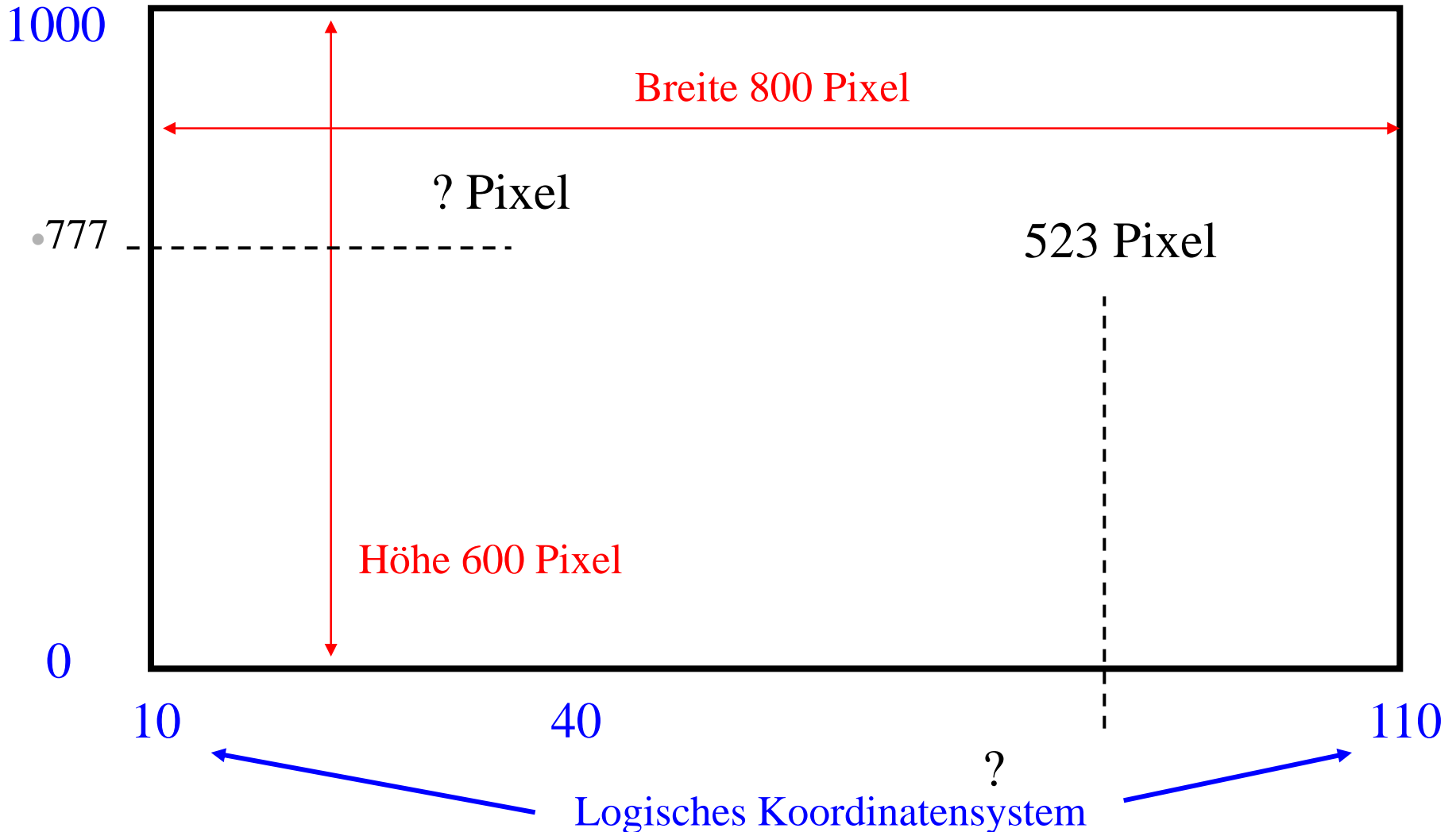
```
public void paint (Graphics g)
{
    int x = 80;
    for (int i=0; i<60; i++) {
        g.drawLine(x,40,x,100);
        x += 1+ 3*Math.random()
    }
}
```



- Die Variable g bezeichnet man auch als Grafikkontext

# Bildschirm-Darstellung

0/0: Bildschirm Nullpunkt



# Tastaturgruppen

## Eingabegerät

- Anschluss
  - DIN
  - PS/2
  - USB
- Buchstaben A bis Z
- Buchstaben a bis z
- Nummern 0 bis 9
- Nummern 0 bis 9 im Ziffernblock
- Sonderzeichen: /() } % # ^ ` ~ <sup>23</sup>
- Funktionstasten, F1 bis F24
- Strg, Shift, Alt, Numlock, Capslock, Alt GR=Alt Graph
- Enter-Taste, Enter-Taste beim Nummernblock
- Übertragen werden jeweils zwei Bytes

# Tastaturgruppen

- Buchstaben A bis Z
- Buchstaben a bis z
- Nummern 0 bis 9
- Nummern 0 bis 9, Ziffernblock
- Sonderzeichen
- Funktionstasten, F1 bis F24
- Strg, Shift, Alt, Numlock, Capslock
- Enter-Taste, Enter-Taste beim Nummernblock



# Tastatur-Events

- Low-Level-Events in Java
  - Key-Down
    - Eine Taste wurde gedrückt
    - Auch die Shift-Taste
  - Key-Pressed
    - Eine Taste **ist** gedrückt
  - Key-Up
    - Die Taste wurde losgelassen
- High-Level-Events in Java
  - Key-Typed
    - Anzeige von Uni-Character Zeichen (S\_A)

- Die Methode **keyTyped** wird immer dann aufgerufen, wenn eine Zeichentaste gedrückt wurde. Beim Drücken einer Funktionstaste wird sie dagegen nicht aufgerufen. Im Gegensatz dazu wird **keyPressed** bei jedem Tastendruck aufgerufen, unabhängig davon, ob es sich um eine Zeichentaste oder eine Funktionstaste handelt.
- Mit **keyPressed** können sogar in beschränktem Umfang die Feststelltasten wie NUM LOCK oder CAPS LOCK erkannt werden. Beide Methoden erhalten auch Tastatur-Repeats, werden also bei längerem Festhalten einer Taste wiederholt aufgerufen.
- Die Methode **keyReleased** wird aufgerufen, wenn eine gedrückte Taste losgelassen wurde, unabhängig davon, ob es sich um eine Zeichen- oder Funktionstaste handelte.
- **Methoden**
  - **getKeyCode**
  - **getKeyChar**
  - **isShiftDown**
  - **isControlDown**
  - **isMetaDown**
  - **isAltDown**

# Maus

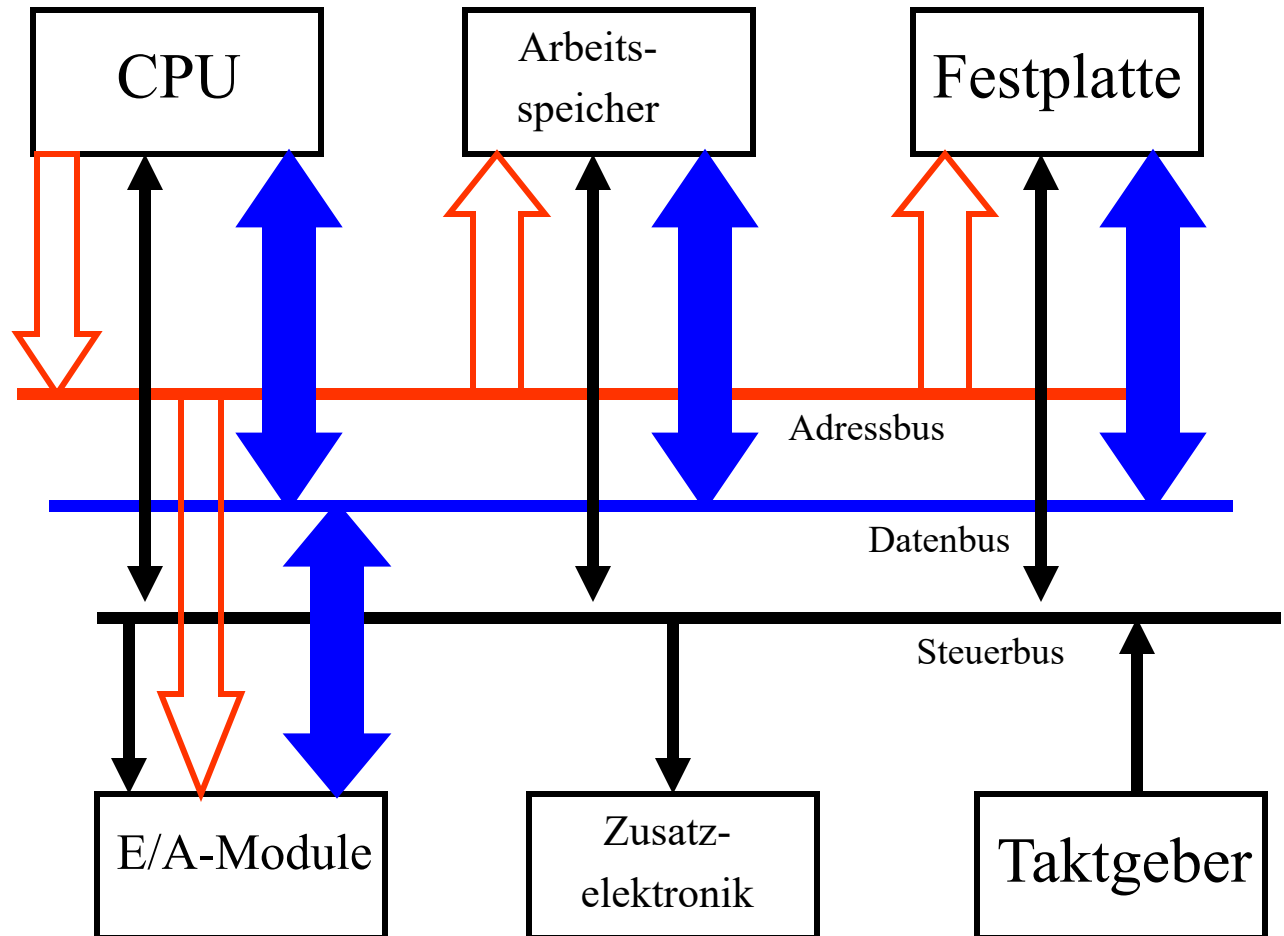


# Maus

- Externes Gerät
  - Linke Maustaste
  - Rechte Maustaste
  - Mittlere Maustaste
  - Drag & Drop
    - Mouse-Down-Event
    - Mouse-Move-Event
    - Mouse-Over-Event
    - Mouse-Up-Event
- Mouse-Down-Event
  - Mouse-Move-Event
  - Mouse-Entered
  - Mouse-Over-Event
  - Mouse-Up-Event
  - Mouse-Exit
  - Mouse-Wheel-Down-Event
  - Mouse-Wheel-Up-Event
  - Mouse-Wheel-Event
  - Single-Click
  - Double-Click
  - Umrechnung der Koordinatensysteme

# Modifizierte Architektur

- Busse
  - Adressbus
  - Datenbus
  - Steuerbus
- Mikro-  
prozessor

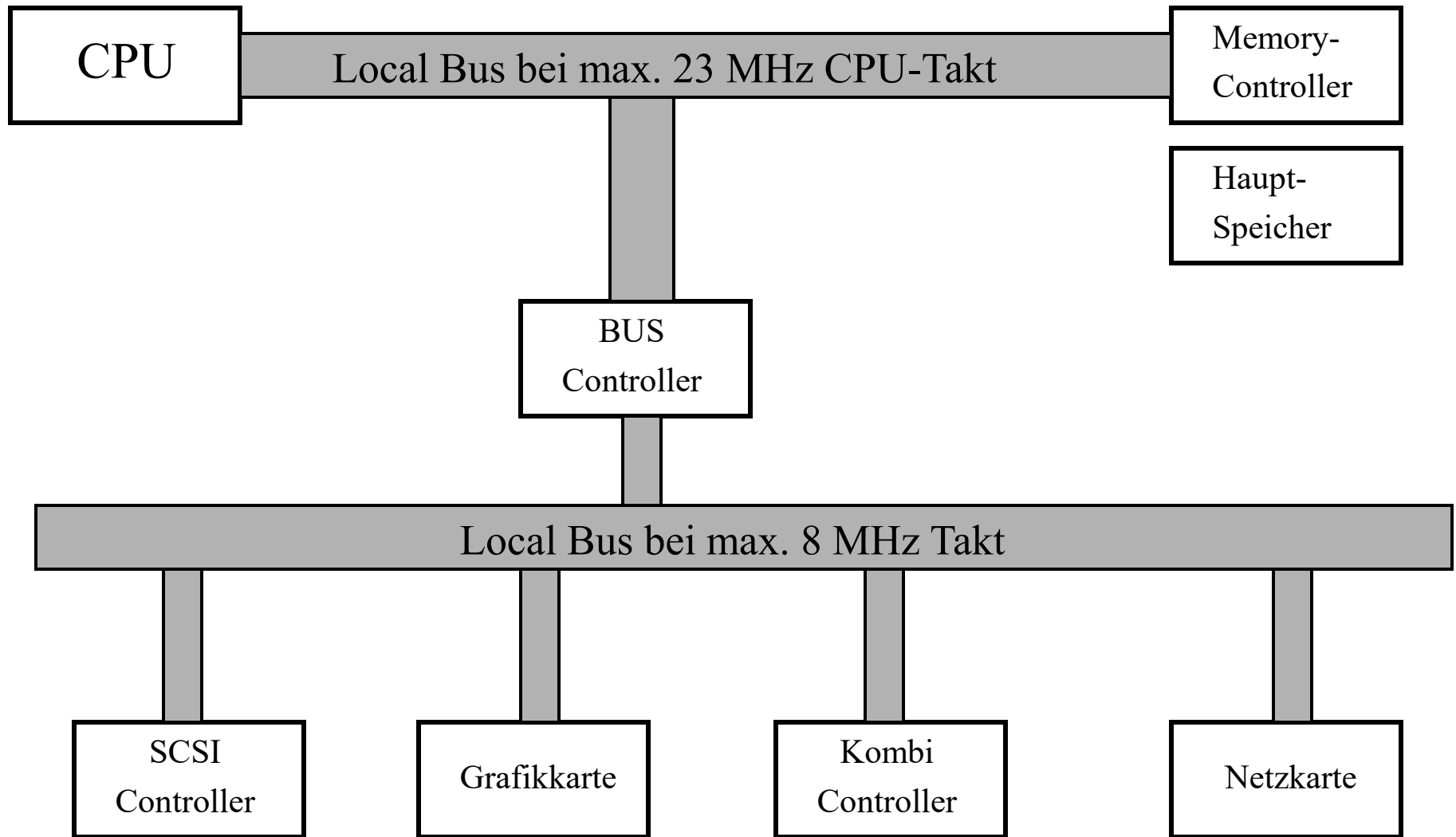


# Übersicht der Busse

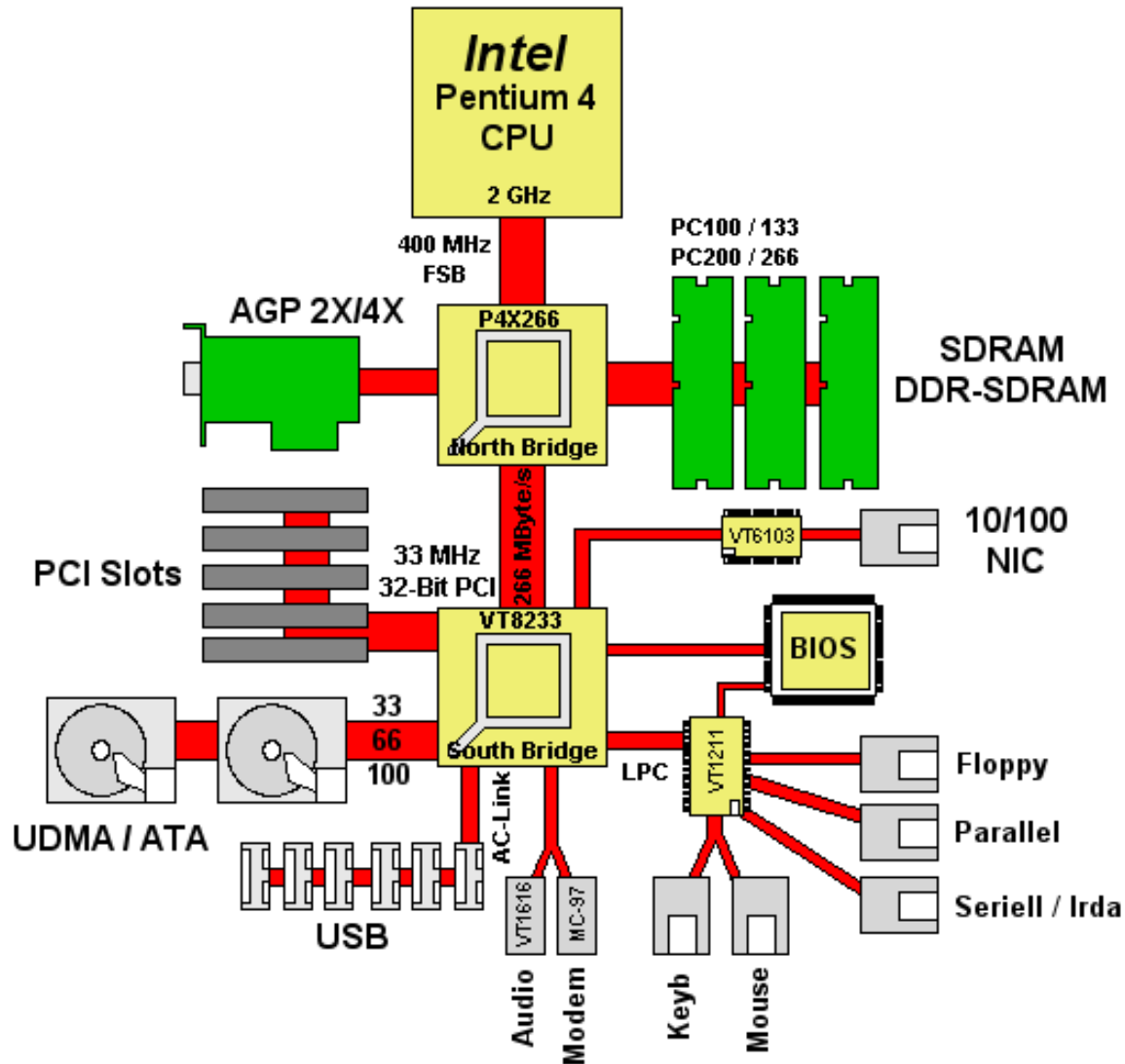
## Systembusse (für Controller)

- ISA Bus
- Mikrokanal Bus (IBM)
- EISA Bus
- Vesa Local Bus (VL Bus)
- PCI Bus
- PCI-X
- NuBus (Apple)
- VME Bus
- AGP-Bus (Steckplatz)
- PCI-Express-Bus

# Der ISA Bus



# Motherboard: PCI-BUS



Quelle: <http://www.hardwaregrundlagen.de>



# Aufgaben eines Motherboards

- Die Northbridge sollte die Daten so schnell transportieren können wie sie die CPU liefert.
- Der Arbeitsspeicher sollte die Daten so schnell aufnehmen können wie sie von der CPU kommen.
- Die Interne Verbindung (Datenautobahn) zwischen North- und Southbridge sollte so groß sein wie nur möglich, um einen Flaschenhals zu vermeiden. Hier krankt es bei den meisten Mainboards erheblich.

Quelle: <http://www.hardwaregrundlagen.de>)

# Eigenschaften des PCI Bus

## **Der Peripheral Component Interconnect Bus (PCI Bus):**

- ist Prozessorunabhängig (z.B. Intel, Alpha, Motorola's PowerPC)
- führt die Konfiguration selbständig durch
- erlaubt PCI-Bus Komponenten das Busmastering, d.h. Datentransport ohne Unterstützung des Prozessors
- arbeitet asynchron; ist geeignet für Multiprozessorsysteme
- hat die Basisdaten: 33 bis 1200 MHz Taktfrequenz, 32bit Pfadbreite für Daten und Adressen, max. Transferrate von 117 MByte/s bis 6 GByte/s

## Northbridges und Southbridge

Die heutigen Mainboards haben entweder einen Haupt-Kontroll-Chip für das Bussystem oder lassen diese Aufgaben von zwei Chips erledigen. Die letztere Variante kommt sehr häufig vor und damit die beiden Chips klar zu unterscheiden sind wurden sie von den Amerikanern in Northbridge und Southbridge getauft.

### Northbridge:

Dieser Chip ist für den High-Speed Bereich von CPU, Arbeitsspeicher sowie der AGP-Grafikkarte verantwortlich. Hier laufen die meisten und schnellsten Daten hin und her. Außerdem ist die Entwicklung dieser Baugruppen sehr viel weiter als die Entwicklung der anderen Komponenten.

### Southbridge:

Dieser Chip regelt den Rest des Datenverkehrs. Hierzu zählen die Laufwerke, sämtliche Schnittstellen sowie die Steckplätze für die restlichen Erweiterungskarten. (Quelle: <http://www.hardwaregrundlagen.de>)

# Motherboard: Asus P5 GD2-X





- Intel LGA775 Pentium 4 CPU
- Intel 915P Chipsatz
- Dual-Channel DDR2 533
- PCI Express Architektur
- DDR2 600 Native Support
- ASUS Hyper Path2

## **Internal I/O**

2 x USB 2.0 connectors supports additional 4 USB 2.0 ports 1 x COM connector 1 x GAME/MIDI connector CPU / 2\*Chassis / Power Fan connectors Front panel High Definition Audio connector Chassis Intrusion connector CD audio-in connector 24-pin ATX Power connector 4-pin ATX 12V Power connector

## **Lan:**

- Marvell® 88E8053 PCIe Gigabit LAN Controller, features AI NET2

## **Audio:**

- C-Media High Definition Audio 8-channel CODEC S/PDIF out on back I/O port

## **Slots:**

- 1 x PCI Express x16 slot for discrete graphics card
- 3 x PCI Express x1
- 3 x PCI

# Busse für Peripheriegeräte

## **E/A-Busse (von Controllern gesteuert)**

- serielle und parallele Schnittstelle
- IDE Festplattenbus (PATA)
- EIDE Festplattenbus (PATA)
- SATA Festplattenbus
- SCSI Bus (8 Bit), SCSI 2 (bis 16 Bit), 32 Bit
- PCMCIA
- USB 1/2 (bis 2128 Geräte),
- IrDA
- Firewire IEEE 1394, 400 Mbps



# Computer-Architekturen

## ■ John von Neumann Architektur

- Single BUS
- Daten und Programmspeicher sind zusammen

## ■ Harvard Architektur

- Multi BUS Systeme (Parallelität)
- Daten und Programmspeicher sind getrennt

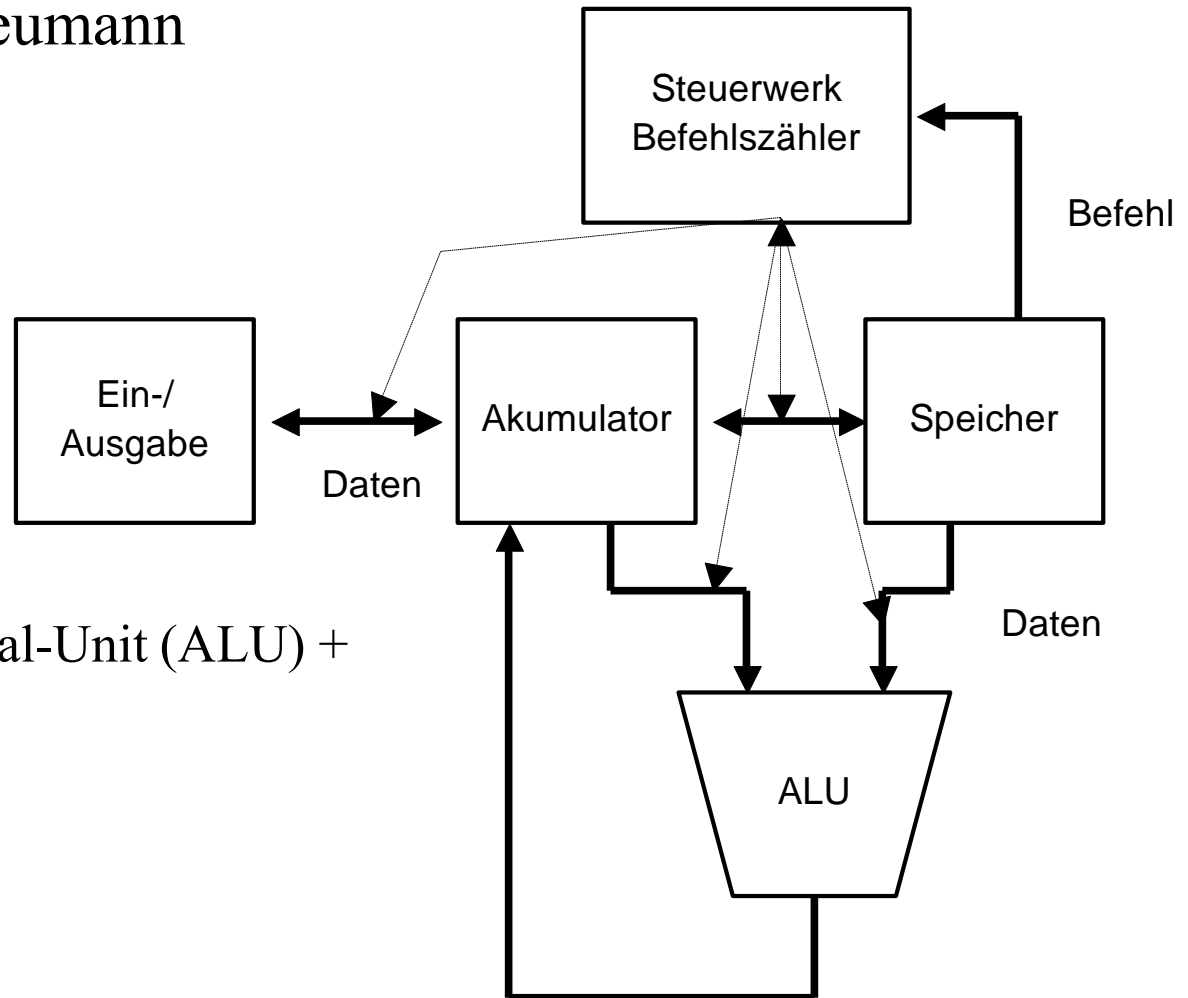
# Von-Neumann Architektur

nach John von Neumann

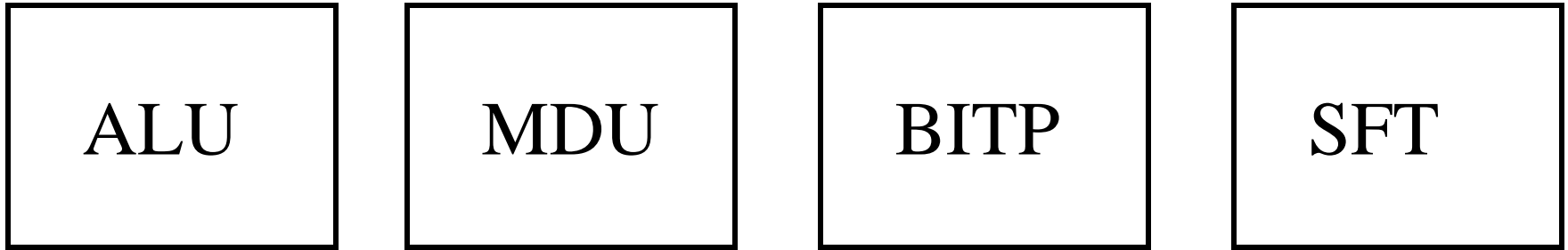
**Prozessor** =

Rechenwerk +  
Arithmetic-Logical-Unit (ALU) +  
Steuerwerk

**Speicher  
Module**



# Rechenwerk:



**ALU:** Arithmetisch-Logische-Einheit

**MDU:** Multiplikation / Division

**BITP:** Bitprozessor

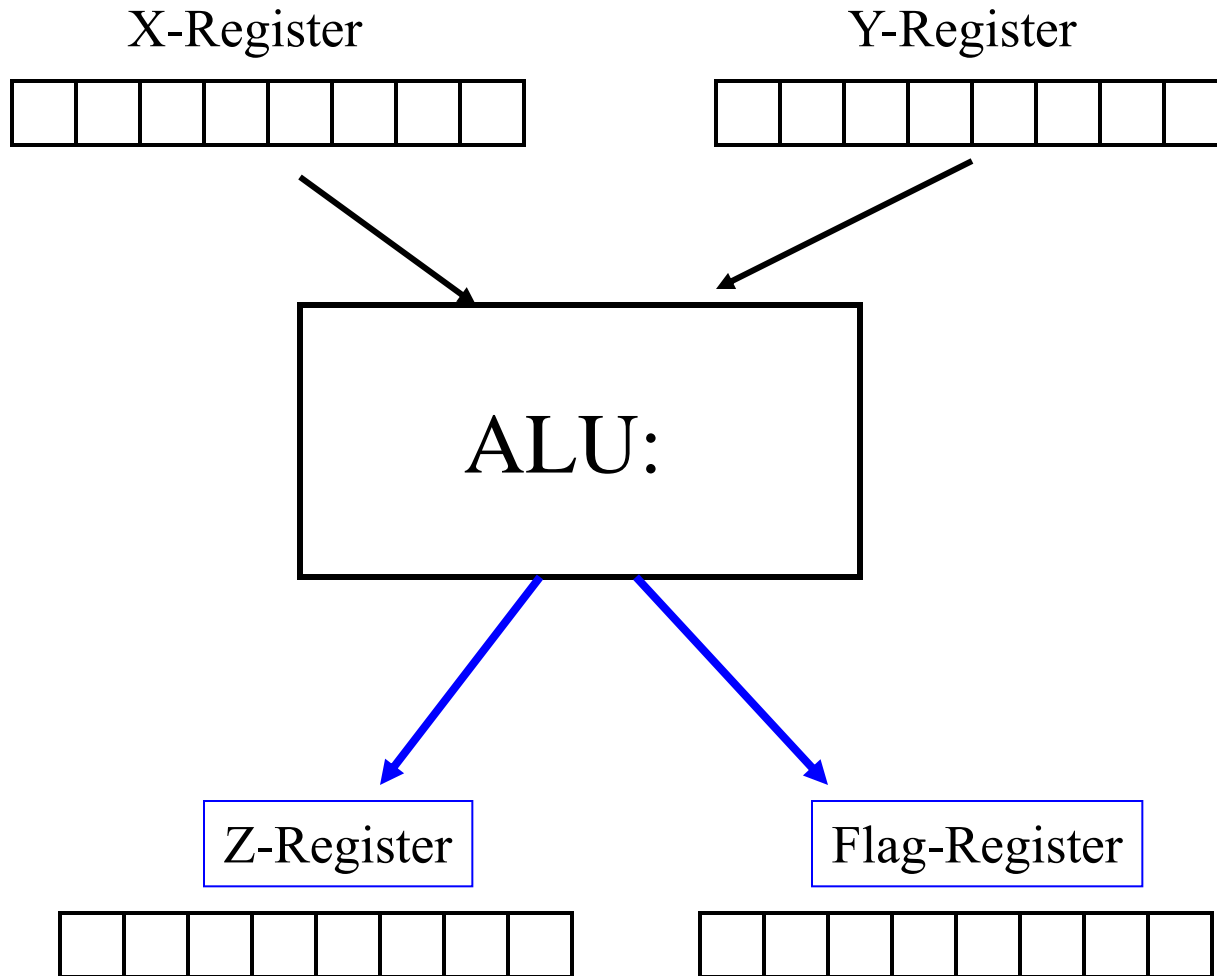
**SFT:** Barrel Shifter

(+) Spezialisierte Module

(+) Parallele Bearbeitung

(+) Optimierte Befehle in Zusammenarbeit mit Compiler (MMX)

# Arithmetisch-Logische Einheit:



# Arithmetisch-Logische Einheit:

Realisierung der Elementaroperationen:

ADD, SUB, AND, OR, XOR, NOT, EQ

Registerbreite: 8, 16, 32, 64

## Ausnahmefälle (Flag):

Overflow: Ergebnis zu groß für Register Z, Zahl multi. mit 10

Underflow: Ergebnis zu klein für Register Z, Zahl div. mit 10

Sign: Ergebnis ist negativ

Zero: Ergebnis ist null

## ■ Operationsprinzipien

- Der Prozessor (die Central Processing Unit, CPU) arbeitet taktgesteuert.
- Die internen Signale sind binär codiert.
- Der Inhalt einer Speicherzelle wird über eine Adresse angesprochen.
- Programme und Daten werden sequentiell bearbeitet (single instruction single data, SISD).
- Es wird eine feste Wortlänge zur Verarbeitung verwendet.

## ■ Bitmuster im Speicher repräsentieren 3 Arten:

- Daten, Befehle, Adressen

# Die Elemente des Prozessors

- Das **Steuerwerk** enthält den Befehlszähler (Adresse des nächsten Befehls), das Befehlsregister und das Adressregister (Adresse des nächsten Datums)
- Das **Rechenwerk** (ALU) enthält mehrere Datenregister und Zusatzregister (Flagregister). Die Verknüpfung erfolgt hier gleichzeitig innerhalb eines Takts mit allen Bits je nach Datenbreite
- Einsatz von **Mikrocode** zur Steuerung der internen Abläufe: eigener Speicher (ROM), Adressen der Register, Steuerwerk zur Abfolge von Mikrobefehlen in mehreren Takten, Ziel: maximale Parallelisierung bei minimalem Aufwand an Logikschaltungen

# Beispiel: Java Virtual Machine

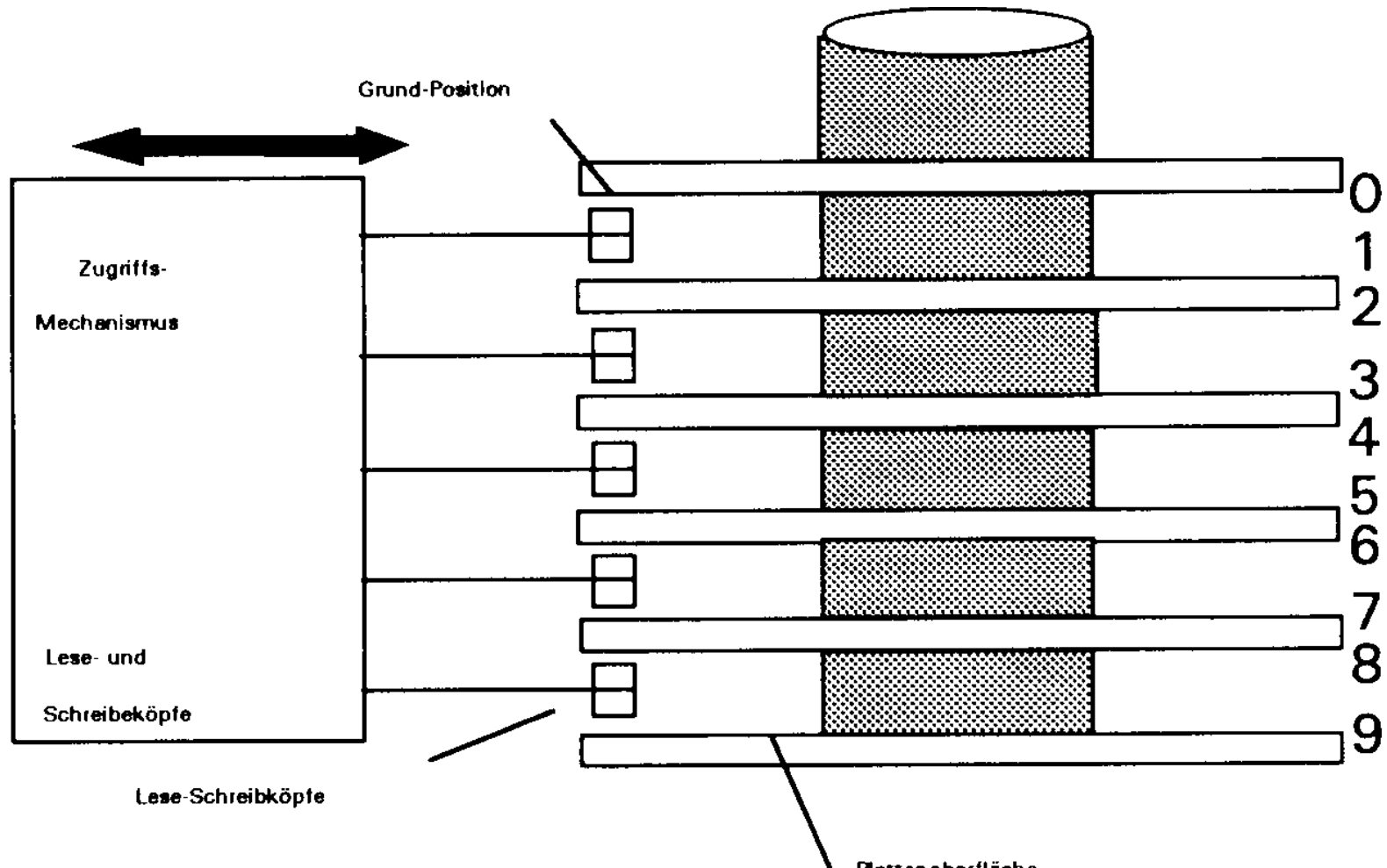
- Steuerwerk wird durch ein Programm eines realen Prozessors verwirklicht (emuliert Java VM)
- Als Mikrocode wird der Code eines realen Prozesses ausgeführt
- ermöglicht komplexe Befehle zur Adressierung von Objekten und Klassen
- kann auf eigener Hardware implementiert werden (Java Chip)



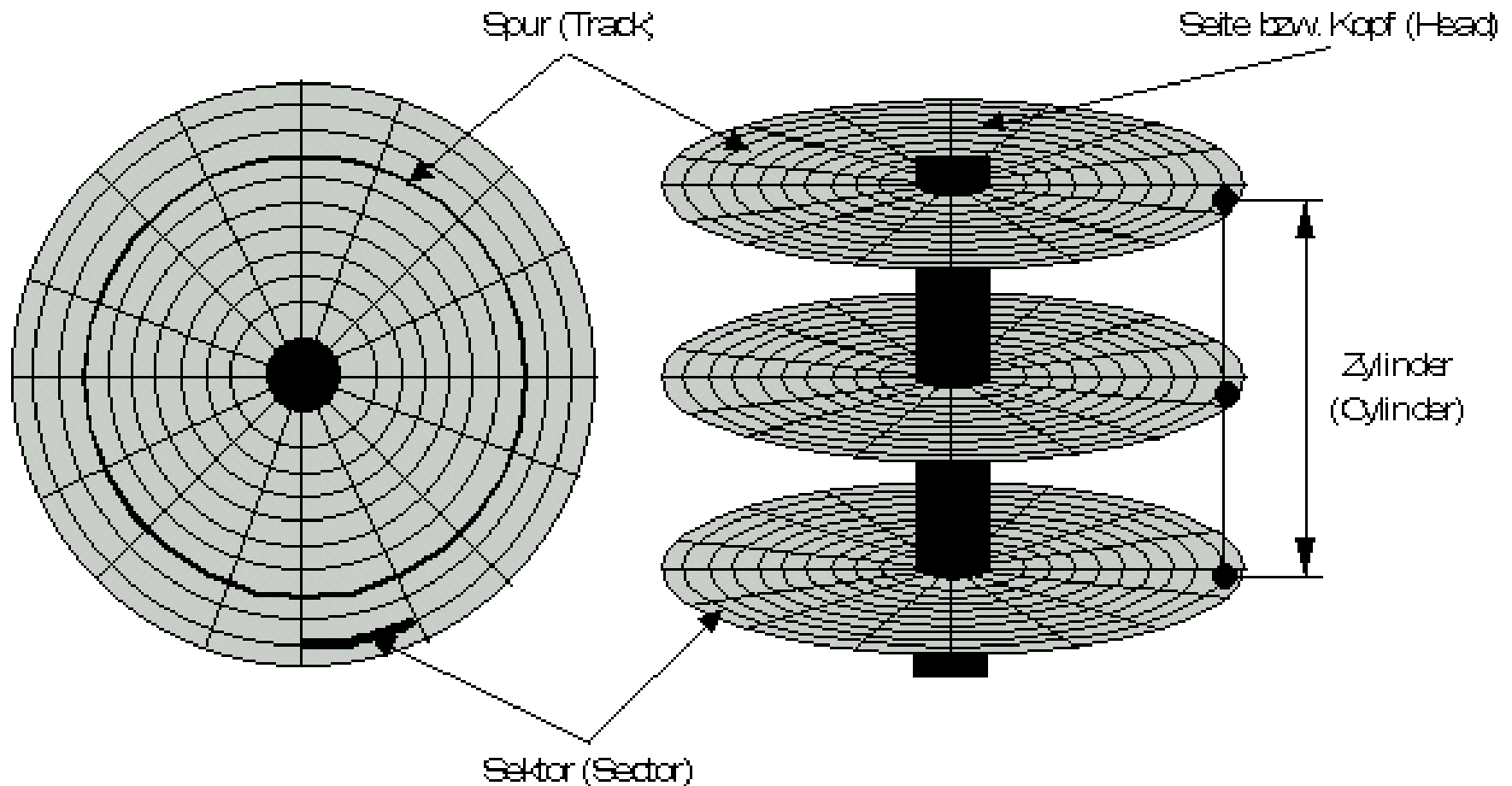
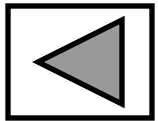
# Hauptspeicher

- Beispiele der Speichergröße (RAM)
  - 128 KB
  - 4 GB
  - 2 TB, nur 64-bit Betriebssystem
- Cache
  - Level1: Cache (Register)
  - Level2: Cache
  - Level3: Cache
- Speicheradressierung abhängig von der BUS-Breite

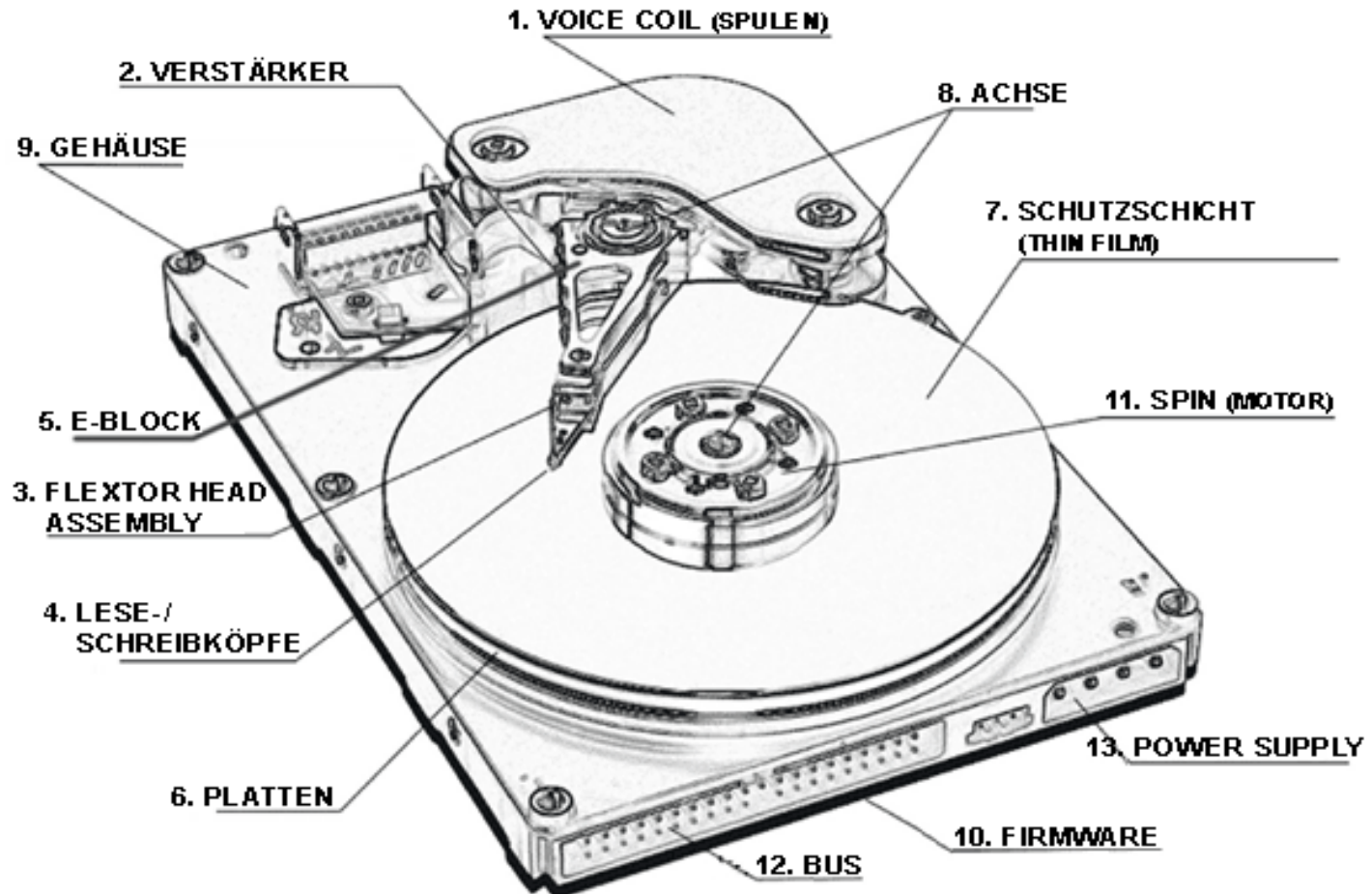
# Massenspeicher: Festplatte



# Aufbau einer Festplatte



# Aufbau einer Festplatte



# Festplatte

## ■ Einteilung in

- Sektoren (512 oder 4096 Byte)
- Spuren
- Kopf
- Zylinder

## ■ Betriebssystem

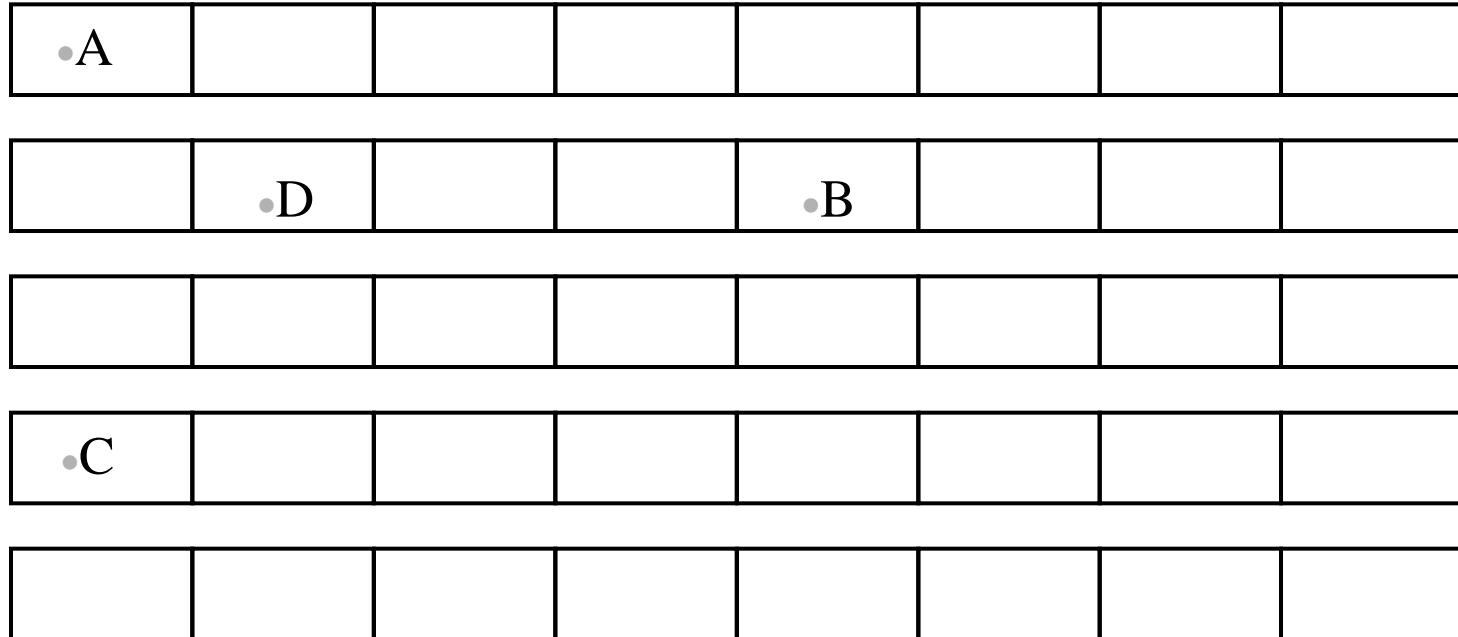
- Verwendet ein Cluster, n-faches eines Sektors

## ■ Adressierung im Betriebssystem von 0 bis n-1

## ■ Adressierung in der Festplatte mit Sektor#, Spur#, Kopf#

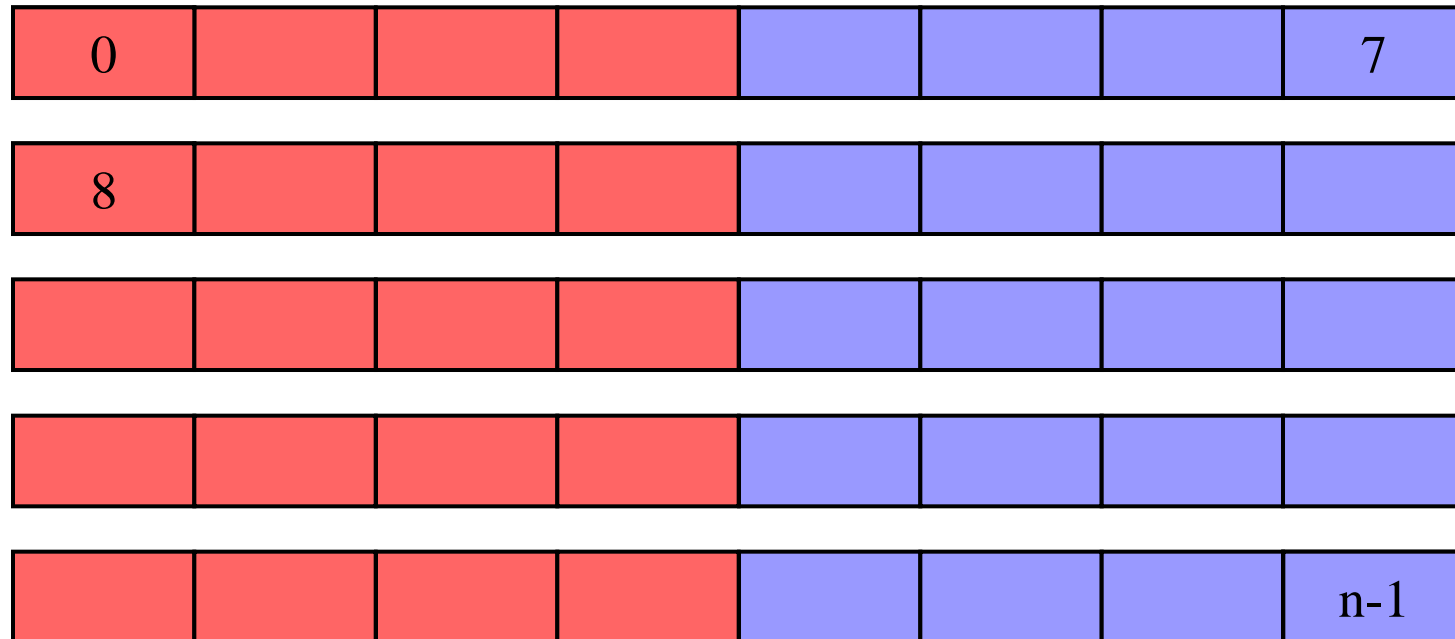
# Festplatte: Clusterdefinition des Betriebssystems

Anzeige: Sektoren der Festplatte, alle 512 Byte groß



# Festplatte: Clusterdefinition des Betriebssystems

Anzeige: Sektoren der Festplatte, alle 512 Byte groß

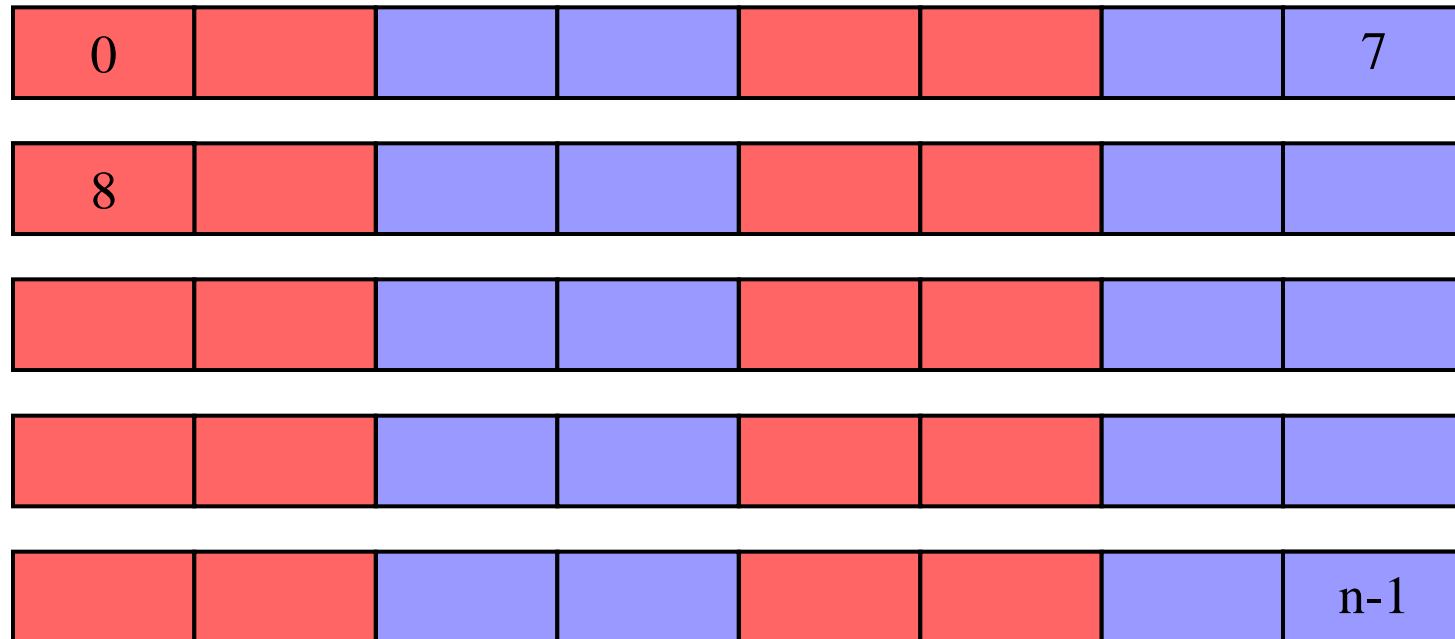


Ein BS-Cluster ist ein Vielfaches eines Sektors:  $2^x$

Bild zeigt Belegung wenn Clustergröße 2048 Bytes sind  $\Rightarrow$  4 Sektoren/Cluster

# Festplatte: Clusterdefinition des Betriebssystems

Anzeige: Sektoren der Festplatte, alle 512 Byte groß



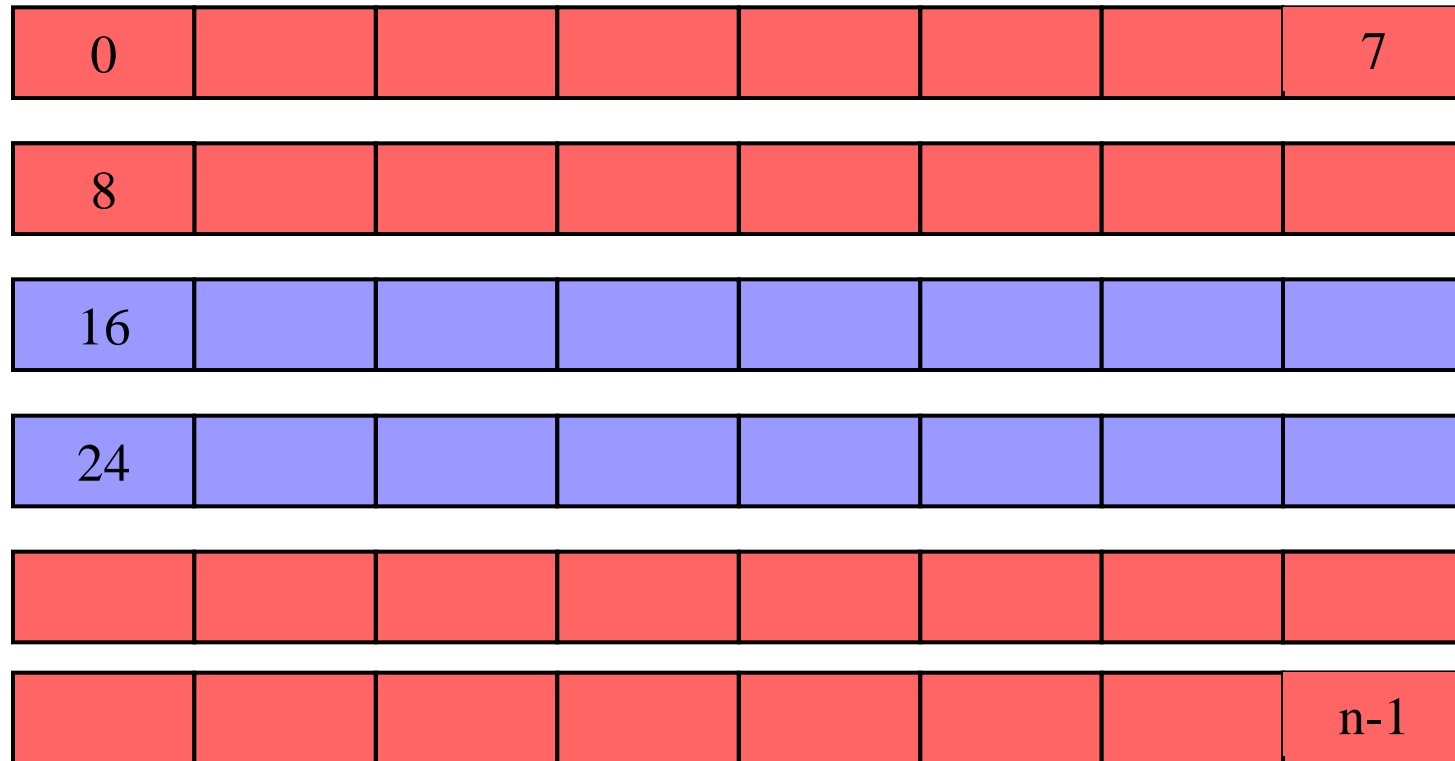
Ein BS-Cluster ist ein Vielfaches eines Sektors:  $2^x$

Bild zeigt Belegung wenn Clustergröße 1024 Bytes sind  $\Rightarrow$  2 Sektoren/Cluster



# Festplatte: Clusterdefinition des Betriebssystems

Anzeige: Sektoren der Festplatte, alle 512 Byte groß



Ein BS-Cluster ist ein Vielfaches eines Sektors:  $2^x$

Bild zeigt Belegung wenn Clustergröße 8192 Bytes sind  $\Rightarrow$  16 Sektoren/Cluster

# Festplatte

## ■ Typen

- IDE (PATA)
- EIDE (PATA)
- SCSI, SCSI-2, SCSI-3
- SATA, SATA-2
- Lan 1394

## ■ Festplatte aus Speicher

- Solid State Disks, Firma Samsung
- SATA-Festplatte aus Flash-Bausteinen
- Datendurchsatz 40 MB/s, ca. 2,5" Festplatte
- Zugriffszeit 0,2 ms
- Energieverbrauch: 0,5 Watt
- Problem beim Schreiben: Das Suchen freier Sektoren dauert lange
- 240 GB: 182,00 € (Corsair),      128 GB: 102,00 € (Intel)
- 480 GB: 370,00 € (OSZ)      960 GB: 549,00 € (Crucial)
- 600 GB: 889,00 € (Intel)

## RAIDDrive II Plus

Preis: 5000,00



- RAIDDrive II Plus is able to support a max read speed of up to 3.2 GB/sec and a max write speed as fast as 3.2 GB/sec with a turbocharged cache system. The RAIDDrive II Plus houses eight discrete SATA SSDs and comes in a custom form factor measuring 231.5 x 94.0 x 20.6 mm.
- **Features**
- PCIe Gen.2 x8 lane host interface
- Max Speed: Read up to 3.2 GB/s, Write up to 3.2 GB/s
- Capacities up to 2 TB
- Burst Speed: Read/Write (MB/sec)                      3200/3200
- RAID Cache Size:                      1GB
- Number of Port:                      8 port
- [http://www.supertalent.com/datasheets/6\\_100010.pdf](http://www.supertalent.com/datasheets/6_100010.pdf)

## M.2-SSD

Preis: 1049,00



- Samsung 960 PRO 2 TB, Solid State Drive
- Die Samsung 960 PRO 2 TB unterstützt das NVMe™ Protokoll sowie die PCIe® Gen 3 x4 Schnittstelle und bietet damit höhere Bandbreiten und niedrigere Latenzen als herkömmliche SATA SSDs.
- Die sequenzielle Lese- und Schreibgeschwindigkeiten liegen bei jeweils bis zu 3.500/2.100 MB/s und die Random Lese-/Schreibgeschwindigkeiten bei bis zu 440/360K IOPS.
- Diese Leistung der Extraklasse wird unter anderem durch ein optimiertes Package-Design ermöglicht.

# CD / DVD

- CD, maximal 800 MB
- DVD, 4,7 GB bis maximal 7,92 GB
- HD-DVD , maximal 50 GB (nicht mehr produziert)
- Blu-Ray-Disk,
  - 27 GB (Single)
  - 50 GB (Double-Disk)
  - 400 GB (Multi-Disk)
- HVD, ca. 300 GB (in Planung)

# Wireless-Lan (WLAN)

## ■ Übertragungsraten

<b>EE 802.11</b>	2 Mbps maximal
<b>IEEE 802.11a</b>	54 Mbps maximal (108 Mbps bei 40 MHz Bandbreite proprietär)
<b>IEEE 802.11b</b>	11 Mbps maximal (22 Mbps bei 40 MHz Bandbreite proprietär, 44 Mbps bei 60 MHz Bandbreite proprietär)
<b>IEEE 802.11g</b>	54 Mbps maximal (g+ =108 Mbps proprietär, bis 125 Mbps möglich)
<b>IEEE 802.11h</b>	54 Mbps maximal (108 Mbps bei 40 MHz Bandbreite)
<b>IEEE 802.11n</b>	300 Mbps maximal (Verwendung von MIMO-Technik; Entwurf am 20. Januar 2006 verabschiedet; Draft 2.0 am 19. März 2007 als neuer Entwurf verabschiedet)

# Wireless-Lan (WLAN)

## ■ Sicherheit

Wired Equivalent Privacy (WEP)	Ein Sicherheitsstandard, der den RC4-Algorithmus enthält. Die darin enthaltene Verschlüsselung mit einem nur 40 Bit, (64 Bit genannt) bzw. 104 Bit (128 Bit genannt), bei einigen Herstellern auch 232 Bit (256 Bit genannt). Kann recht einfach entschlüsselt werden
WEPplus, Wi-Fi Protected Access (WPA) TKIP bei WPA	Etwas besser
Advanced Encryption Standard (AES) bei WPA2	Kann zur Zeit nicht entschlüsselt werden
MAC-Adresse	Eine Authentifizierung ist auch über die MAC-Adresse der drahtlosen Netzwerkadapter möglich. Die WLAN-IP-Adressen werden im Router eingetragen, sehr leicht knackbar

# Grundlegende Sicherheitsmaßnahmen im WLAN

Dazu gehören einige Einstellungen am Router bzw. AP:

- Aktivierung der Verschlüsselung mit einer sicheren Verschlüsselungsmethode, d. h. mindestens WPA
- Vergabe eines sicheren Netzwerkschlüssels
- Ersetzen der werkseitig voreingestellten Router- bzw. AP-Passwörter
- Änderung des werkseitig voreingestellten SSID-Namens
- Deaktivierung der Fernkonfiguration des Routers, soweit vorhanden (insbesondere bei privaten Haushalten)

[http://www.netgear.de/Support/Basiswissen/wireless\\_lan\\_grundlagen.html](http://www.netgear.de/Support/Basiswissen/wireless_lan_grundlagen.html)



# Logische Verknüpfungen

<b>ODER</b>	<b>0</b>	<b>1</b>
<b>0</b>	0	1
<b>1</b>	1	1

<b>UND</b>	<b>0</b>	<b>1</b>
<b>0</b>	0	0
<b>1</b>	0	1

# Logische Verknüpfungen

	<b>0</b>	<b>1</b>
<b>NOT</b>	1	0

<b>XOR</b>	<b>0</b>	<b>1</b>
<b>0</b>	0	1
<b>1</b>	1	0

XOR ergibt eine 1, wenn die Bits unterschiedlich sind!

# Beispiele:

Diese logische Verknüpfungen werden auch auf Bitfolgen angewandt.

0001 ODER 1000 = 1001

0001 OR 1000 = 1001

0001  $\vee$  1000 = 1001

0001 | 1000 = 1001

1011 UND 1000 = 1000

1011 AND 1000 = 1000

1011  $\wedge$  1000 = 1000

1011 & 1000 = 1000

# Beispiele:

Diese logische Verknüpfungen werden auch auf Bitfolgen angewandt.

NOT	1010	=	0101
NICHT	1010	=	0101
¬	1010	=	0101
!	1010	=	0101

1011	XOR	1000	=	0011
1010	XOR	1111	=	????
0101	XOR	1111	=	????