# DigitalTechnik
# Kapitel 1

Prof. Dr.-Ing. Sarah Harris
Fachgebiet Eingebettete Systeme und ihre Anwendungen (ESA)
Fachbereich Informatik

TECHNISCHE UNIVERSITÄT DARMSTADT

WS 15/16

# Wer bin ich?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Stanford University
  PhD, Electrical and Computer Engineering
  (2005)

- Harvey Mudd College
  Assistant/Associate Professorin
  (2004-2014)

- University of Nevada, Las Vegas
  Associate Professorin
  (2014 - )

# Wer bin ich?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Technische Universität Darmstadt
  2015 – 2016

- **Fachgebiet:** Eingebettete Systeme, Rechnerarchitekturen

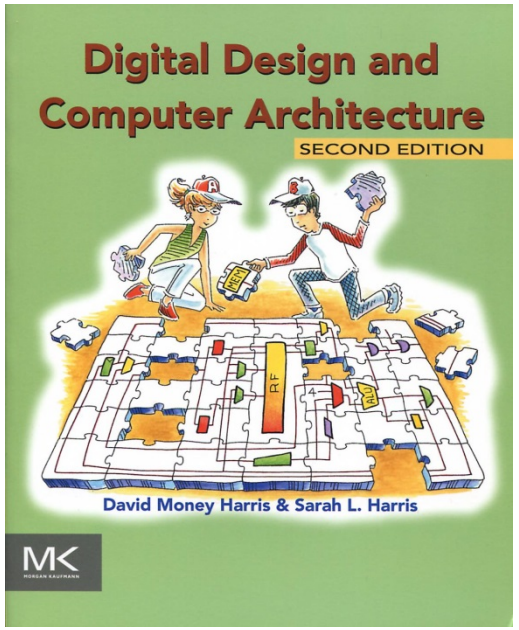
- **Kontakt Informationen:**
  Sprechstunden: Mittwochs, 13:30 Uhr – 14:30 Uhr
  Email: harris@esa.informatik.tu-darmstadt.de
  Büro: S2/02 (Piloty Gebäude), Raum E102

# Lehr- und Anschauungsmaterial

TECHNISCHE UNIVERSITÄT DARMSTADT

- Aus dem Lehrbuch

  ***Digital Design and Computer Architecture, Zweite Auflage***




- Diese Folien nach englischen Originalvorlagen erstellt (Originale sind © 2007 Elsevier)
- Buch wird an Studierende subventioniert abgegeben
  - Organisiert durch Fachschaft Informatik
- Mehr Hintergrundmaterial auf Web-Seite zu Buch
- Winter Semester: Kapitel 1 bis 5

# Organisatorisches

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Übungen:** Organisatorisches von dem Assistent morgen erklärt…

- **Webseite:** www.esa.informatik.tu-darmstadt.de
  - unter Lehre → WS 2015/2016 → DigitalTechnik
  - Vorlesungsfolien, Aufzeichnungen der Vorlesungen, u.s.w.

- **Prüfung:** 1. März 2016

# Lehrphilosophie

TECHNISCHE UNIVERSITÄT DARMSTADT

- Wenn Sie **Fragen** haben, bitte stellen Sie die mal
- Ich werde Ihnen auch Fragen stellen während der Vorlesung

TECHNISCHE UNIVERSITÄT DARMSTADT

# Die Grundlagen der Digitaltechnik

- Hintergrund
- Vorgehensweise
- Beherrschen von Komplexität
- Die digitale Abstraktion
- Zahlensysteme
- Logikgatter
- Darstellung als elektrische Spannungen
- CMOS Transistoren
- Elektrische Leistungsaufnahme

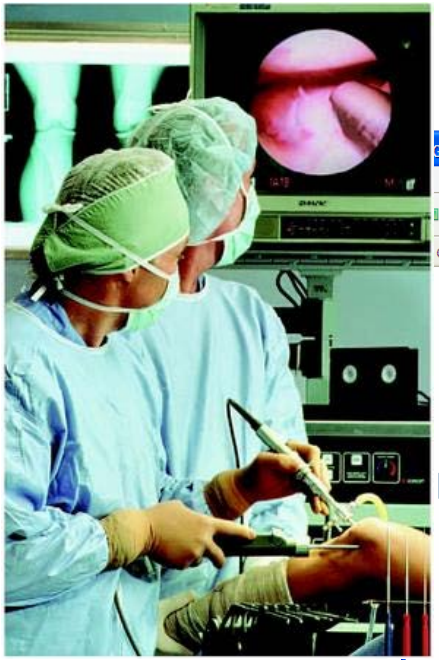
# Hintergrund

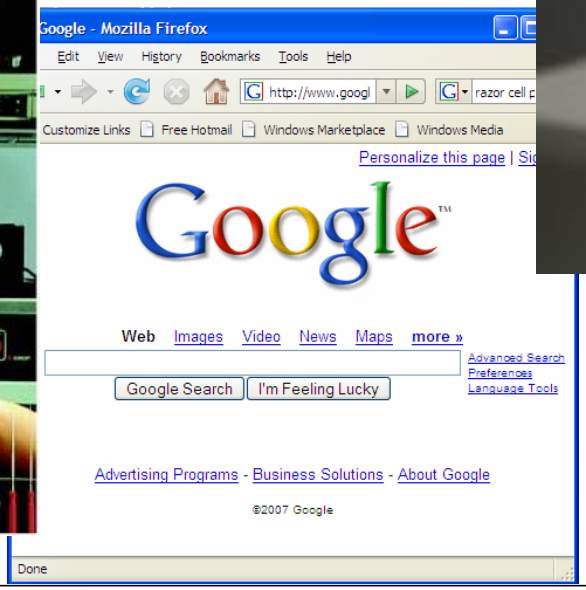


- Wie haben Digitalgeräte die Welt verändert?
- Wie haben Digitalgeräte Ihre Welt verändert?

# Hintergrund

- Mikroprozessoren haben die Welt verändert
  - Handys, Internet, Medizintechnik, Unterhaltung, …
- Umsatzwachstum in der Halbleiterindustrie von $21 Milliarden in 1985 auf $323 Milliarden in 2015

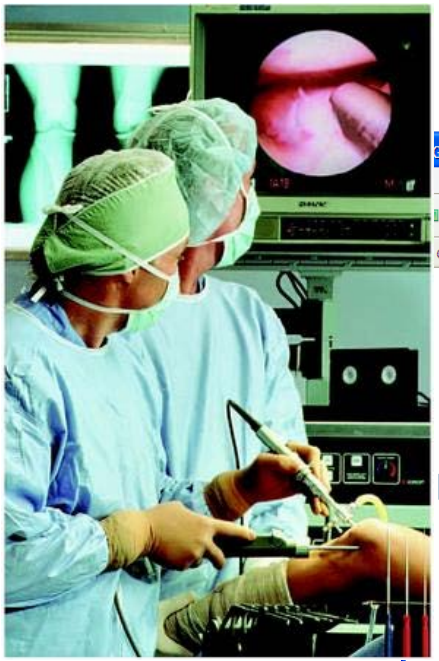


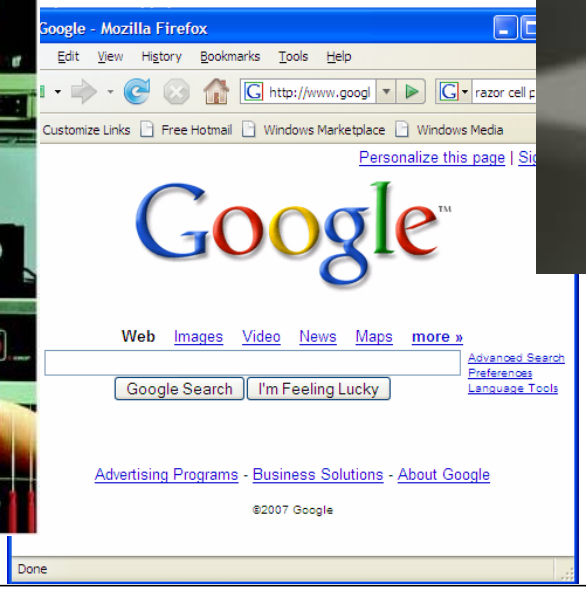



# Hintergrund

TECHNISCHE UNIVERSITÄT DARMSTADT

Als Informatiker/in werden Sie Computer von der Pike auf verstehen!






# Themen dieser Veranstaltung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Entwurf digitaler Logikschaltungen
- Systematische Fehlersuche in digitalen Logikschaltungen
- Interner Aufbau und Funktion eines Computers
- Entwurf und Realisierung eines Mikroprozessors (Sommer Semester)

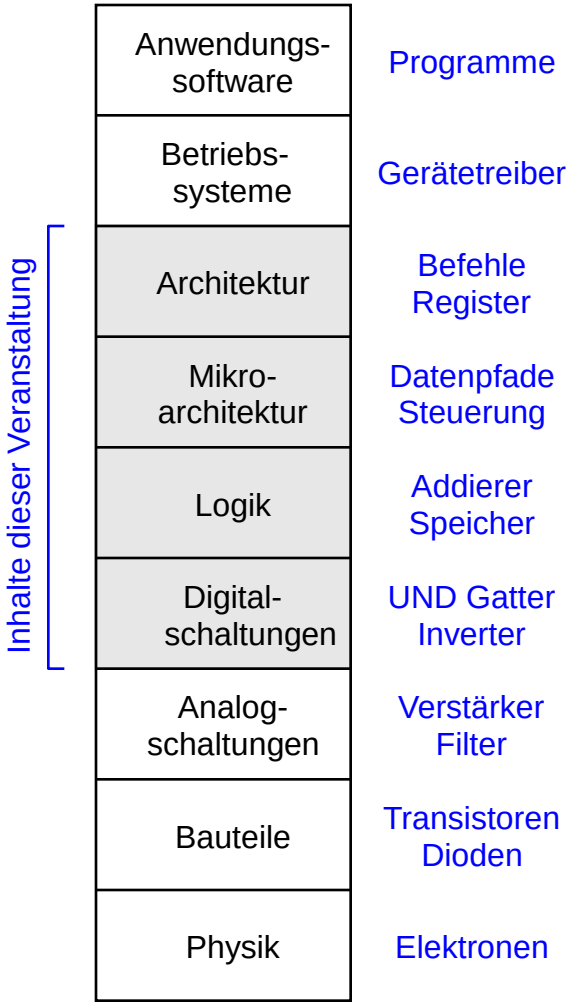
# Beherrschen von Komplexität

TECHNISCHE UNIVERSITÄT DARMSTADT

- Abstraktion
- Disziplin
- Wesentliche Techniken (die drei **Y**'s)
  - Hierarchie (*hierarch**y***)
  - Modularität (*modularit**y**)*
  - Regularität (*regularit**y**)*

# Abstraktion

- Verstecken unnötiger Details
- „unnötig“
  - Für *diese* spezielle Aufgabe unnötig!
- Für alle Aufgaben hilfreich:
  - Verstehen der anliegenden Abstraktionsebenen



# Disziplin

TECHNISCHE UNIVERSITÄT DARMSTADT

- Wissentliche Beschränkung der Realisierungsmöglichkeiten
  - Erlaubt produktivere Arbeit auf höheren Entwurfsebenen
- Beispiel: Digitale Entwurfsdisziplin
  - Arbeite mit diskreten statt mit stetigen Spannungspegeln
  - Digitalschaltungen sind einfacher zu entwerfen als analoge
    - Erlaubt den Entwurf komplexerer Schaltungen
  - Digitale Systeme ersetzen zunehmend analoge
    - Digitalkamera, digitales Fernsehen, moderne Handys, CD, DVD, …

# Beherrschen von Komplexität

TECHNISCHE UNIVERSITÄT DARMSTADT

- Abstraktion
- Disziplin
- Wesentliche Techniken (die drei **Y**'s)
  - Hierarchie (*hierarch**y***)
  - Modularität (*modularit**y**)*
  - Regularität (*regularit**y**)*

# Wesentliche Techniken (Die Drei-Y's)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Hierarchie (Hierarch*y*)
  - Aufteilen eines Systems in Module und Untermodule
- Modularität (Modularit*y*)
  - Wohldefinierte Schnittstellen und Funktionen
- Regularität (Regularit*y*)
  - Bevorzuge einheitliche Lösungen für einfachere Wiederverwendbarkeit

# Beispiel: Steinschlossgewehr

TECHNISCHE UNIVERSITÄT DARMSTADT

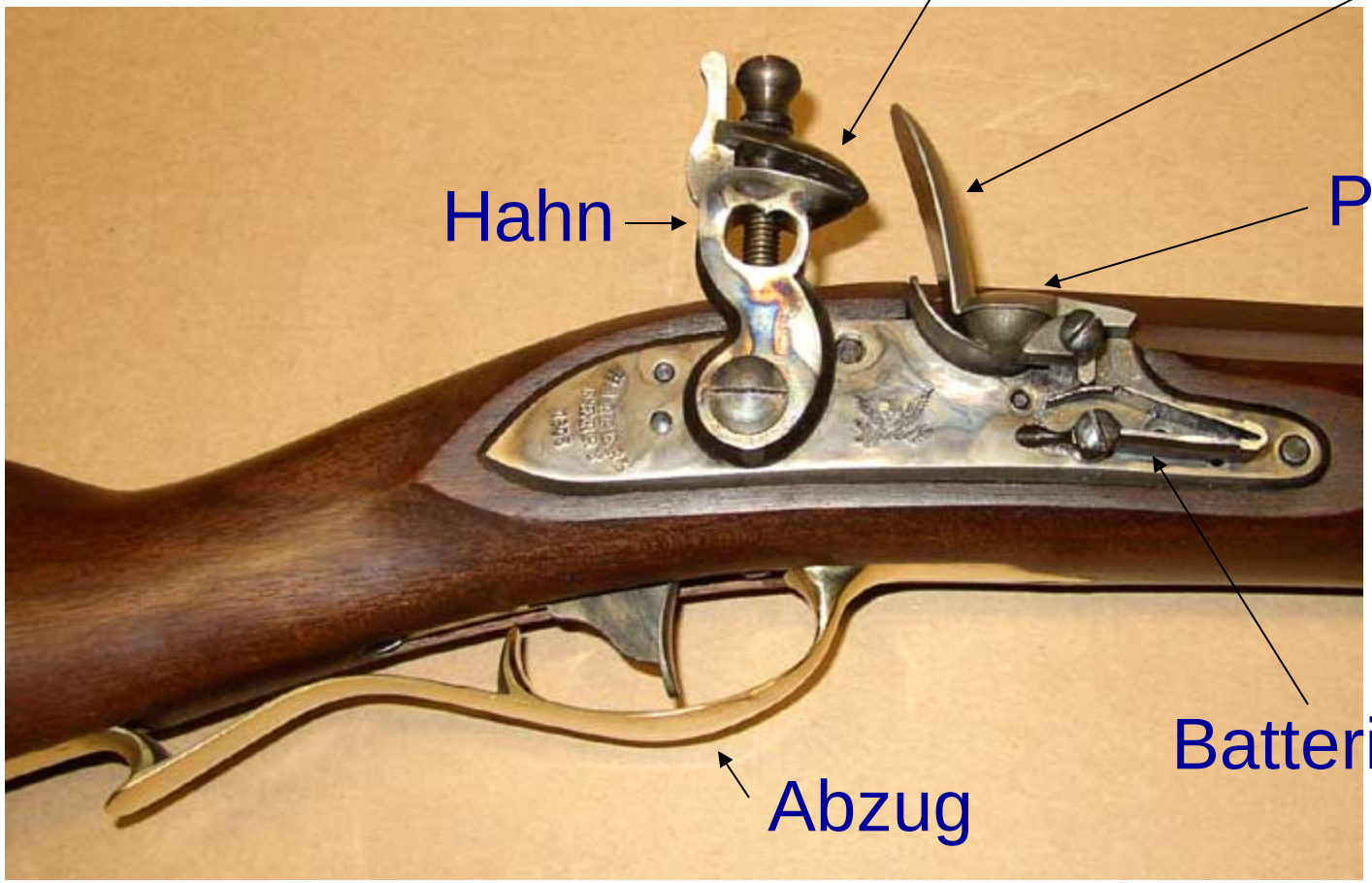
- Frühes Beispiel für Anwendungen der Drei-Y's
- Komplexer Gebrauchsgegenstand
- Entwicklung begann im 16. Jahrhundert
  - Aber noch sehr unzuverlässig
- Höhere Stückzahlen ab dem 17. Jahrhundert
  - Aber alles Einzelanfertigungen von Büchsenmachern
- Bis zum 19. Jahrhundert zunehmende Vereinheitlichung

# Hierarchie: Zerlegung in Module

# Hierarchie: Zerlegung in Untermodule

TECHNISCHE UNIVERSITÄT DARMSTADT

- Untermodule des Schlosses



# Modularität: Schaft und Lauf

TECHNISCHE UNIVERSITÄT DARMSTADT

- Funktion des Schafts
  - Schloss und Lauf stabil zusammenfügen
- Funktion des Laufes
  - Projektil während Beschleunigung zu führen und mit Drall zu versehen
- Im Idealfall sind Funktionen unabhängig und beeinflussen sich nicht
- Schnittstelle zwischen Schaft und Lauf
  - Gemeinsame Haltevorrichtung

# Regularität: Austauschbare Teile

TECHNISCHE UNIVERSITÄT DARMSTADT

- Gleiche Schlösser in unterschiedlichen Schäften
  - Passender Ausschnitt in Schaft
- Unterschiedliche Läufe in gleichen Schäften
  - Passende Länge und Haltemechanismus
- Voraussetzung für industrielle Massenproduktion

# Beherrschen von Komplexität

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Abstraktion**
- Disziplin
- Wesentliche Techniken (die drei **Y**'s)
  - Hierarchie (*hierarch**y***)
  - Modularität (*modularit**y**)*
  - Regularität (*regularit**y**)*

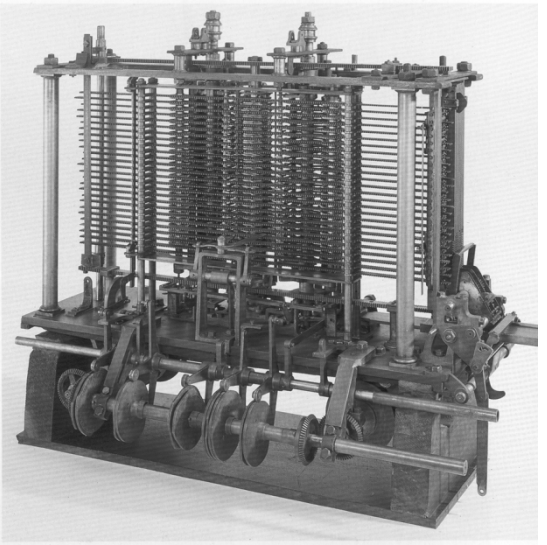
# Digitale Abstraktion

TECHNISCHE UNIVERSITÄT DARMSTADT

- Die meisten physikalischen Größen haben stetige Werte
  - Elektrische Spannung auf einem Leiter
  - Frequenz einer Schwingung
  - Position einer Masse
- Berücksichtigen alle Werte der Größe (unendlich viele)
- Digitale Abstraktion: Berücksichtigt nur endlich viele Werte
  - Untermenge aus einem stetigen Wertebereich

# Analytische Maschine

- *Analytical engine*
- Entworfen durch Charles Babbage von 1834 – 1871
- Erster Digitalrechner
- Aufgebaut als mechanischer Rechner
  - Zahnstangen und –räder
  - Stellungen repräsentieren Ziffern 0-9
    - Genau 10 Stellungen je Zahnrad
- Babbage verstarb vor Fertigstellung
- Entwurf hätte aber funktioniert



# Digitale Disziplin: Binärwerte

TECHNISCHE UNIVERSITÄT DARMSTADT

- Digitale Disziplin
  - In der Regel Beschränkung auf nur zwei unterschiedliche Werte
    - Binärsystem
  - Können unterschiedlich heißen
    - 1, WAHR, TRUE, HIGH, …
    - 0, FALSCH, FALSE, LOW, …
- Unterschiedlichste Darstellungen der beiden Werte möglich
  - Spannungspegel, Zahnradstellungen, Flüssigkeitsstände, Quantenzustände, …
- Digitalschaltungen verwenden üblicherweise unterschiedliche Spannungspegel
- *Bit* (*B*inary dig*it):* Maßeinheit für Information
  - 1 b = Eine Ja/Nein-Entscheidung

# Zahlensysteme

TECHNISCHE UNIVERSITÄT DARMSTADT

- Dezimalzahlen
- Binärzahlen
- Hexadezimal

# Zahlensysteme

TECHNISCHE UNIVERSITÄT DARMSTADT

- Dezimalzahlen

1000's column
100's column
10's column
1's column

$5374_{10} =$

- Binärzahlen

8's column
4's column
2's column
1's column

$1101_2 =$

# Zweierpotenzen

TECHNISCHE UNIVERSITÄT DARMSTADT

- $2^0$ =
- $2^1$ =
- $2^2$ =
- $2^3$ =
- $2^4$ =
- $2^5$ =
- $2^6$ =
- $2^7$ =
- $2^8$ =
- $2^9$ =
- $2^{10}$ =
- $2^{11}$ =
- $2^{12}$ =
- $2^{13}$ =
- $2^{14}$ =
- $2^{15}$ =

# Zahlenkonvertierung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Binär nach dezimal umrechnen:
  - Wandele $10011_2$ ins Dezimalsystem um

- Dezimal nach binär umrechnen
  - Wandele $47_{10}$ ins Binärsystem um

# Dezimal nach Binär Umrechnen

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Auf zwei Arten möglich**
  - **Art 1:** Jeweils nach größter noch passender Zweierpotenz suchen
  - **Art 2:** Durch immer größer werdende Zweierpotenzen dividieren

# Dezimal nach Binär Umrechnen

TECHNISCHE UNIVERSITÄT DARMSTADT

**Art 1:** Jeweils nach größter noch passender Zweierpotenz suchen

$53_{10}$

**Method 2:** Durch immer größer werdende Zweierpotenzen dividieren

$53_{10}$ =

# Dezimal nach Binär Umrechnen

TECHNISCHE UNIVERSITÄT DARMSTADT

**Noch ein Beispiel:** $75_{10}$ ins Binär umrechnen

# Binärzahlen und Wertebereiche

TECHNISCHE UNIVERSITÄT DARMSTADT

- ***N*-stellige Dezimalzahl**
  - Wie viele verschiedene Werte? **$10^N$**
  - Wertebereich? **$[0, 10^N - 1]$**
  - **Beispiel:** 3-stellige Dezimalzahl:
    - $10^3$ = 1000 mögliche Werte
    - Wertebereich: [0, 999]
- ***N*-bit Binärzahl**
  - Wie viele verschiedene Werte? **$2^N$**
  - Wertebereich? **$[0, 2^N - 1]$**
  - **Beispiel:** 3-bit Binärzahl
    - $2^3$ = 8 mögliche Werte
    - Wertebereich : [0, 7] = $[000_2 , 111_2]$

# Hexadezimale Zahlen

TECHNISCHE UNIVERSITÄT DARMSTADT

| Hex-Ziffer | Entspricht Dezimal | Entspricht Binär |
|---|---|---|
| 0 | | |
| 1 | | |
| 2 | | |
| 3 | | |
| 4 | | |
| 5 | | |
| 6 | | |
| 7 | | |
| 8 | | |
| 9 | | |
| A | | |
| B | | |
| C | | |
| D | | |
| E | | |
| F | | |

# Hexadezimale Zahlen

TECHNISCHE UNIVERSITÄT DARMSTADT

| Hex-Ziffer | Entspricht Dezimal | Entspricht Binär |
|---|---|---|
| 0 | 0 | 0000 |
| 1 | 1 | 0001 |
| 2 | 2 | 0010 |
| 3 | 3 | 0011 |
| 4 | 4 | 0100 |
| 5 | 5 | 0101 |
| 6 | 6 | 0110 |
| 7 | 7 | 0111 |
| 8 | 8 | 1000 |
| 9 | 9 | 1001 |
| A | 10 | 1010 |
| B | 11 | 1011 |
| C | 12 | 1100 |
| D | 13 | 1101 |
| E | 14 | 1110 |
| F | 15 | 1111 |

# Hexadezimalzahlen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Schreibweise zur Basis 16
- Kürzere Darstellung für lange Binärzahlen

# Umwandeln von Hexadezimaldarstellung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Umwandeln von hexadezimal nach binär:
  - Wandele $4AF_{16}$ (auch geschrieben als 0x4AF) nach binär

- Umwandeln von hexadezimal nach dezimal:
  - Wandele 0x4AF nach dezimal

# Zahlensysteme

TECHNISCHE UNIVERSITÄT DARMSTADT

- Dezimal Basis 10
- Binär Basis 2
- Hexadezimal Basis 16

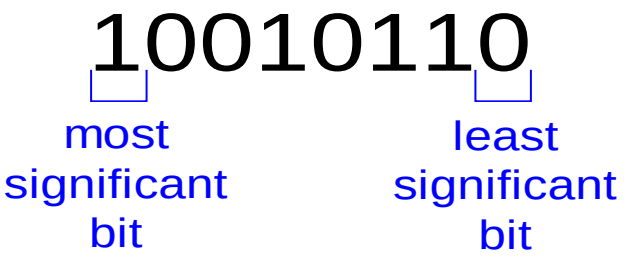
Sonstige:

- Oktal
- Andere Basen

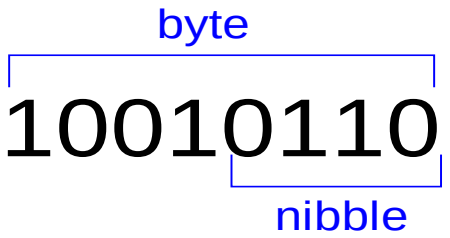
# Bits, Bytes, Nibbles…

TECHNISCHE UNIVERSITÄT DARMSTADT

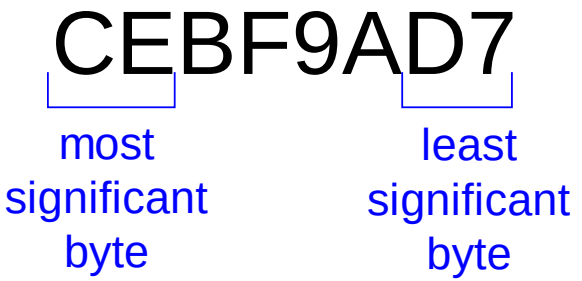
- Bits (Einheit *b*)
  - Höchstwertiges Bit (*msb*)
  - Niedrigstwertiges Bit (*lsb*)




- Bytes (Einheit *B*) & Nibbles




- Bytes
  - Höchstwertiges Byte (*MSB*)
  - Niedrigstwertiges Byte (*LSB*)



# Zweierpotenzen und Präfixe

TECHNISCHE UNIVERSITÄT DARMSTADT

- $2^{10}$ = 1 Kilo (K) ≈ 1000 (1024)
- $2^{20}$ = 1 Mega (M) ≈ 1 Million (1,048,576)
- $2^{30}$ = 1 Giga (G) ≈ 1 Milliarde (1,073,741,824)

- Beispiele
  - 4 GB: Maximal adressierbare Speichergröße für 32b-Prozessoren
  - 16M x 32b: erste GDDR5-Speicherchips für Grafikkarten

- Vorsicht Falle:
  - Deutsch $10^9$=1 Milliarde
  - US English $10^9$=1 *billion*

# Zweierpotenzen schnell schätzen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Was ist der Wert von $2^{24}$?

- Wie viele verschiedene Werte kann eine 32b Variable annehmen?

# Addition

TECHNISCHE UNIVERSITÄT DARMSTADT

- Dezimal

$$\begin{array}{r} 3734 \\ +\ 5168 \\ \hline \end{array}$$

- Binär

$$\begin{array}{r} 1011 \\ +\ 0011 \\ \hline \end{array}$$

# Beispiele für Addition von Binärzahlen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Addiere die 4-bit Binärzahlen

$$\begin{array}{r} 1001 \\ +\ 0101 \\ \hline \end{array}$$

- Addiere die 4-bit Binärzahlen

$$\begin{array}{r} 1011 \\ +\ 0110 \\ \hline \end{array}$$

# Überlauf

TECHNISCHE UNIVERSITÄT DARMSTADT

- Digitale Systeme arbeiten mit einer festen Anzahl an Bits
  - In der Regel, es gibt aber durchaus Ausnahmen!
- Eine Addition läuft über, wenn ihr Ergebnis nicht mehr in die verfügbare Anzahl von Bits hineinpasst
- Beispiel: 11+6, gerechnet mit 4b Breite

# Darstellung von Anzahlen



- Wir haben von postiven Anzahlen geredet.
- Wozu mit den negativen Zahlen?

# Vorzeichenbehaftete Binärzahlen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Darstellung als Vorzeichen und Betrag
- Zweierkomplement

# Darstellung als Vorzeichen und Betrag

TECHNISCHE UNIVERSITÄT DARMSTADT

- 1 Vorzeichenbit, *N*-1 Bits für Betrag

$$A: \{a_{N-1}, a_{N-2}, \ldots a_2, a_1, a_0\}$$

- Vorzeichenbit ist höchstwertiges Bit (msb)
  - Positive Zahl: Vorzeichenbit = 0
  - Negative Zahl: Vorzeichenbit = 1

$$A = (-1)^{a_{N-1}} \sum_{i=0}^{N-2} a_i \, 2^i$$

- **Beispiel:** 4-bit Vorzeichen/Betrag-Darstellung von ± 6:

  +6 = **0110**

  **-** 6 = **1110**

- Wertebereich einer Zahl in Vorzeichen/Betrag-Darstellung :

  **$[-(2^{N-1}-1), 2^{N-1}-1]$**

# Darstellung als Vorzeichen/Betrag: Probleme

TECHNISCHE UNIVERSITÄT DARMSTADT

- Addition schlägt fehl
  - Beispiel: -6 + 6:

```
  1110
+ 0110
------
 10100 (falsch!)
```

- Zwei Darstellungen für Null (± 0):

| | |
|---|---|
| 1000 | (-0) |
| 0000 | (+0) |

# Zahlendarstellung im Zweierkomplement

TECHNISCHE UNIVERSITÄT DARMSTADT

- Behebt Probleme der Vorzeichen/Betrag-Darstellung
  - Addition liefert wieder korrekte Ergebnisse
  - Nur eine Darstellung für Null

# Zahlendarstellung im Zweierkomplement

TECHNISCHE UNIVERSITÄT DARMSTADT

- Wie vorzeichenlose Binärdarstellung, aber …
  - msb hat nun einen Wert von $-2^{N-1}$

$$A = a_{N-1}(-2^{N-1}) \sum_{i=0}^{N-2} a_i \, 2^i$$

- Größte positive 4b Zahl : **0111 = $2^2 + 2^1 + 2^0 = 7$**
- Kleinste negative 4b Zahl : **1000 = $-2^3 = -8$**
- msb gibt immer noch das Vorzeichen an
  - 1=negativ, 0=positiv

- Wertebereich einer *N*-bit Zweierkomplementzahl:
  **$[-(2^{N-1}), 2^{N-1}-1]$**

# Darstellung im Zweierkomplement

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Annahme:** Umzuwandelnde Zahlen liegen im Wertebereich
  - ***N*** bit breites Zweierkomplement
  - Stelle Wert ***k*** im Zweierkomplement ***z*** dar

- **Positive Zahlen:** ***k*** >= 0

- **Negative Zahlen:** ***k*** < 0

# Zweierkomplement arithmetisch bilden

TECHNISCHE UNIVERSITÄT DARMSTADT

- In **beide** Richtungen anwendbar
  - Vorzeichenwechsel: *k* ➔ *-k*
- **Algorithmus**
  1. Alle Bits invertieren (0➔1, 1➔0)
  2. Dann 1 addieren
- **Beispiel:** Vorzeichenwechsel von $3_{10} = 00011_2$
- **Beispiel:** Vorzeichenwechsel von $-3_{10} = 11101_2$

# Weitere Beispiele Zweierkomplement

TECHNISCHE UNIVERSITÄT DARMSTADT

- Bestimme Zweierkomplement von $6_{10} = 0110_2$

- Was ist der Dezimalwert der Zweierkomplementzahl $1001_2$?

# Addition im Zweierkomplement

TECHNISCHE UNIVERSITÄT DARMSTADT

- Addiere 6 + (-6)

$$\begin{array}{r} 0110 \\ +\ 1010 \\ \hline \end{array}$$

- Addiere -2 + 3

$$\begin{array}{r} 1110 \\ +\ 0011 \\ \hline \end{array}$$

# Erweitern von Zahlen auf höhere Bitbreite

TECHNISCHE UNIVERSITÄT DARMSTADT

- Verknüpfen von Zahlen unterschiedlicher Bitbreite?

# Erweitern durch Auffüllen mit Vorzeichenbit

TECHNISCHE UNIVERSITÄT DARMSTADT

- Vorzeichenbit nach links kopieren bis gewünschte Breite erreicht
- Zahlenwert bleibt unverändert
  - Auch bei negativen Zahlen!

- **Beispiel 1:**
  - 4-bit Darstellung von 3 = **0**011
  - 8-bit aufgefüllt durch Vorzeichen:

- **Beispiel 2:**
  - 4-bit Darstellung von -5 = **1**011
  - 8-bit aufgefüllt durch Vorzeichen :

# Erweitern durch Auffüllen mit Nullbits

TECHNISCHE UNIVERSITÄT DARMSTADT

- Nullen nach links anhängen bis gewünschte Breite erreicht
- Zerstört Wert von negativen Zahlen
  - Positive Zahlen bleiben unverändert

- **Beispiel 1:**
  - 4-bit Wert = $0011 = 3_{10}$
  - 8-bit durch Auffüllen mit Nullbits:

- **Beispiel 2:**
  - 4-bit Wert = $1011 = -5_{10}$
  - 8-bit durch Auffüllen mit Nullbits :

# Vergleich der Zahlensysteme

TECHNISCHE UNIVERSITÄT DARMSTADT

| Zahlensystem | Wertebereich |
|---|---|
| Vorzeichenlos | $[0, 2^N\text{-}1]$ |
| Vorzeichen/Betrag | $[\text{-}(2^{N\text{-}1}\text{-}1), 2^{N\text{-}1}\text{-}1]$ |
| Zweierkomplement | $[\text{-}2^{N\text{-}1}, 2^{N\text{-}1}\text{-}1]$ |

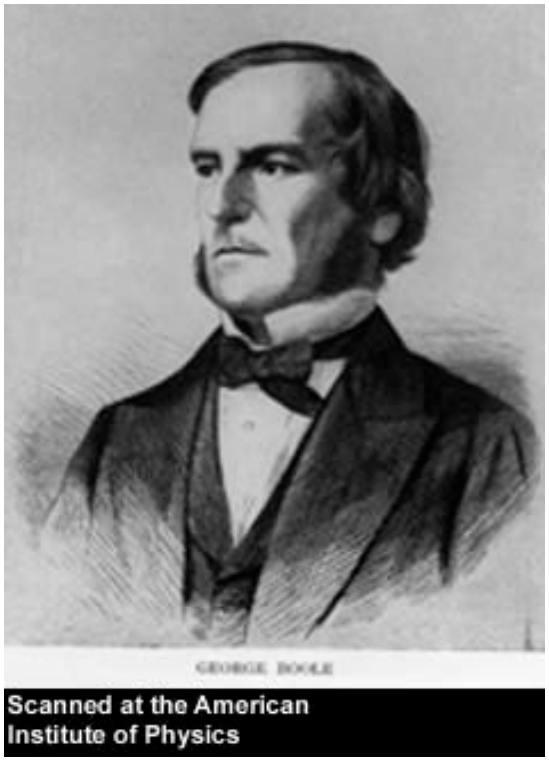
Beispiel 4-bit breite Darstellung:

| -8 | -7 | -6 | -5 | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Vorzeichenlos | | | | | | | | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
| Zweierkomplement | | | | | | | | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | | | | | | | | |
| | | | | | | | | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 | | | | | | | | |
| Vorzeichen/Betrag | | | | | | | | 0000<br>1000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | | | | | | | | |
| | | | | | | | | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 | | | | | | | | | |

# George Boole, 1815 - 1864

TECHNISCHE UNIVERSITÄT DARMSTADT

- In einfachen Verhältnissen geboren
- Brachte sich selbst Mathematik bei
- Später Professor am Queen's College in Irland
- Verfasste *An Investigation of the Laws of Thought*
  - 1854
- Einführung binärer Variablen
- Einführung der drei grundlegenden Logikoperationen
  - UND (*AND*)
  - ODER (*OR*)
  - NICHT (*NOT*)
- Verknüpfen binäre Werte mit binärem Ergebnis



# Logische Gatter

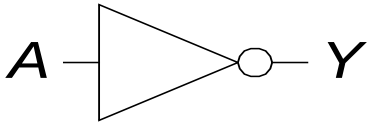
TECHNISCHE UNIVERSITÄT DARMSTADT

- **Berechnen logische Funktionen:**
  - Inversion (NICHT), UND, ODER, …
  - NOT, AND, OR, NAND, NOR, …
- **Ein Eingang:**
  - NOT Gatter, Puffer (*buffer*)
- **Zwei Eingänge:**
  - AND, OR, XOR, NAND, NOR, XNOR
- **Viele Eingänge**

# Logikgatter mit einem Eingang

TECHNISCHE UNIVERSITÄT DARMSTADT

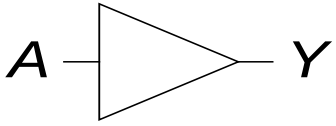
## NOT




$Y = \overline{A}$

| A | Y |
|---|---|
| 0 | |
| 1 | |

## BUF



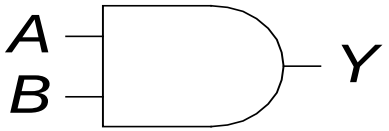

$Y = A$

| A | Y |
|---|---|
| 0 | |
| 1 | |

# Logikgatter mit zwei Eingängen

TECHNISCHE UNIVERSITÄT DARMSTADT

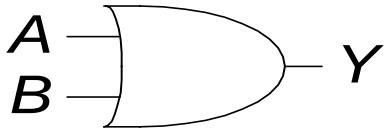
## AND




$Y = AB$

| A | B | Y |
|---|---|---|
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |

## OR



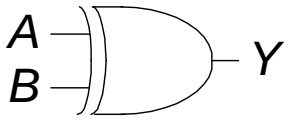

$Y = A + B$

| A | B | Y |
|---|---|---|
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |

# Weitere Logikgatter mit zwei Eingängen

TECHNISCHE UNIVERSITÄT DARMSTADT

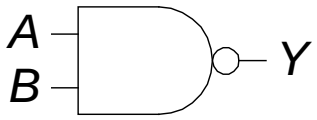
## XOR




$Y = A \oplus B$

| A | B | Y |
|---|---|---|
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |

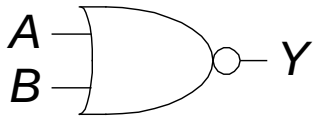
## NAND




$Y = \overline{AB}$

| A | B | Y |
|---|---|---|
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |

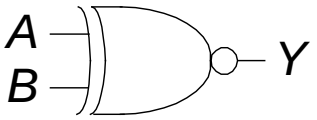
## NOR




$Y = \overline{A + B}$

| A | B | Y |
|---|---|---|
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |

## XNOR



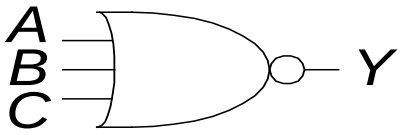

$Y = \overline{A \oplus B}$

| A | B | Y |
|---|---|---|
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |

# Logikgatter mit mehr als zwei Eingängen

TECHNISCHE UNIVERSITÄT DARMSTADT

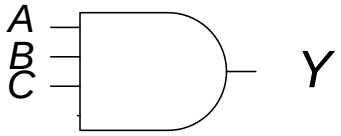
## NOR3




$$Y = \overline{A+B+C}$$

| A | B | C | Y |
|---|---|---|---|
| 0 | 0 | 0 | |
| 0 | 0 | 1 | |
| 0 | 1 | 0 | |
| 0 | 1 | 1 | |
| 1 | 0 | 0 | |
| 1 | 0 | 1 | |
| 1 | 1 | 0 | |
| 1 | 1 | 1 | |

## AND3




$$Y = ABC$$

| A | B | C | Y |
|---|---|---|---|
| 0 | 0 | 0 | |
| 0 | 0 | 1 | |
| 0 | 1 | 0 | |
| 0 | 1 | 1 | |
| 1 | 0 | 0 | |
| 1 | 0 | 1 | |
| 1 | 1 | 0 | |
| 1 | 1 | 1 | |

# XOR mit mehreren Eingängen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Paritätsfunktion
  - Erkennt gerade oder ungerade Anzahl von Eingängen mit Wert 1
- XOR
  - Ungerade Paritätsfunktion
  - Liefert 1 am Ausgang, wenn ungerade Anzahl von Eingängen den Wert 1 haben

# Darstellung von Binärwerten durch Spannungen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Definiere Spannungspegel für die Werte 0 und 1
  - Logikpegel *(logic levels)*

- **Beispiel:**
  - 0 Volt (Erde, ground) entspricht Binärwert 0
  - 5 Volt (Versorgungsspannung, $V_{DD}$) entspricht Binärwert 1

- **Probleme**
  - Wofür steht 4,99 V? Den Wert 0 oder 1?
  - Wofür steht 3,2V?

- Reale Schaltungen haben keine ganz exakten Spannungspegel
  - Teils sogar Umgebungsabhängig (Temperatur, Einstreuen, …)
  - Solche Spannungsschwankungen werden Rauschen genannt

# Was ist Rauschen?

# Darstellung von Binärwerten durch Spannungen

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Lösung**
  - Statt einzelner Spannungspegel für 0 und 1 …
  - … verwende Bereiche von Spannungspegeln für 0 und 1

- Steigere Robustheit durch unterschiedliche Bereiche für
  - Eingänge
  - Ausgänge

# Statische Entwurfsdisziplin

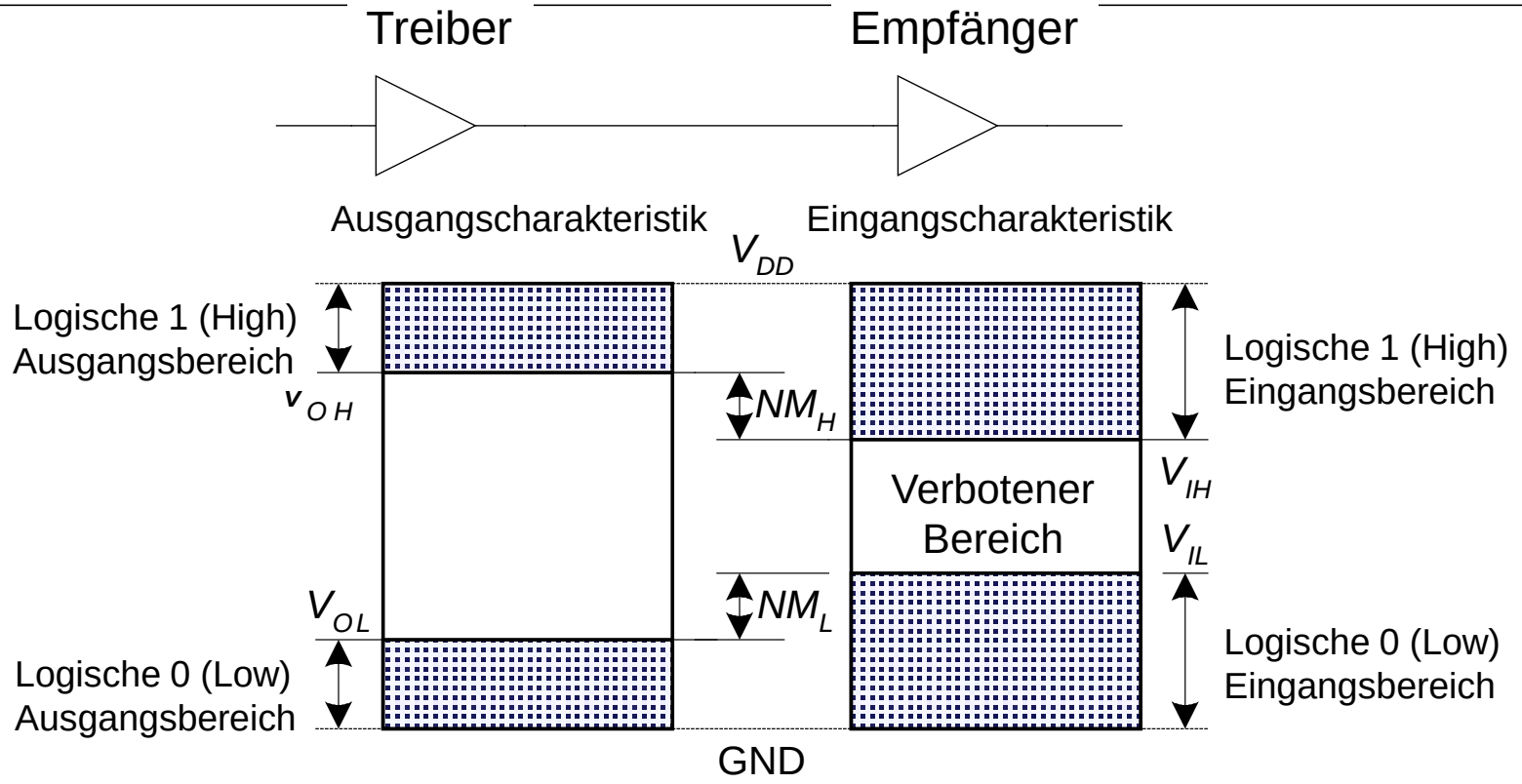
TECHNISCHE UNIVERSITÄT DARMSTADT

- Jedes Schaltungselement muss bei Eingabe gültiger Logikpegel auch am Ausgang einen gültigen Logikpegel liefern

- Verwende nur einen Satz Spannungsbereiche für Logikpegel in gesamter Schaltung
  - Wird manchmal bewusst missachtet
    - Optimierung von Platz, Geschwindigkeit, Energiebedarf, Kosten, …
  - … bedarf aber großer Vorsicht

# Logikpegel

# Störabstand (*noise margin*)



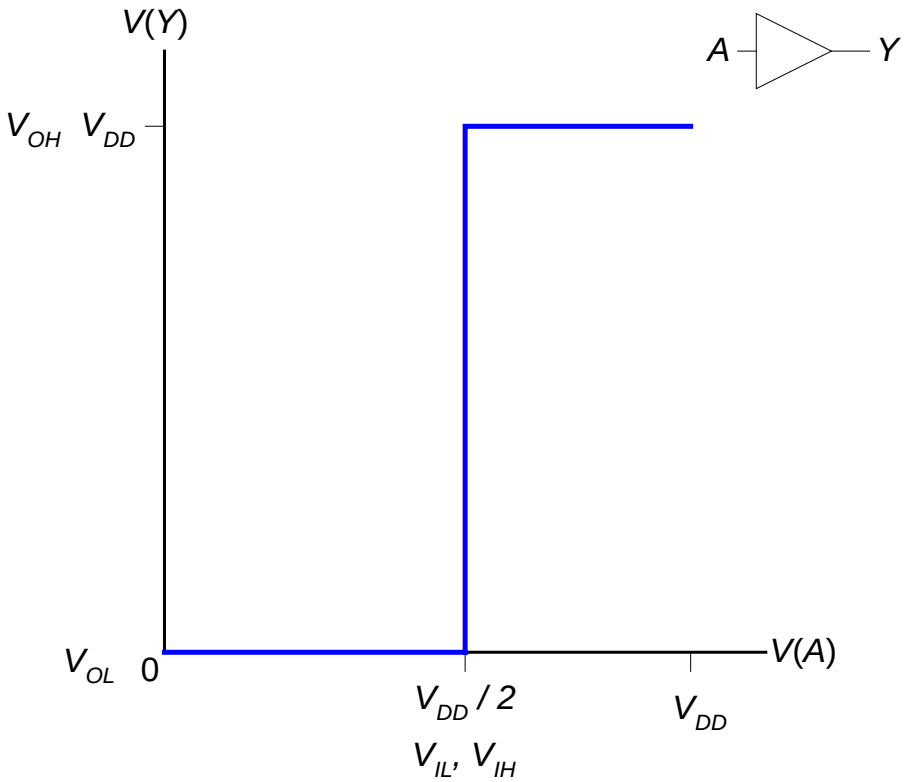

***Hoch Störabstand:*** $NM_H = V_{OH} - V_{IH}$

***Niedriger Störabstand:*** $NM_L = V_{IL} - V_{OL}$

# Gleichstrom-Transferkurve (*DC transfer characteristics*)

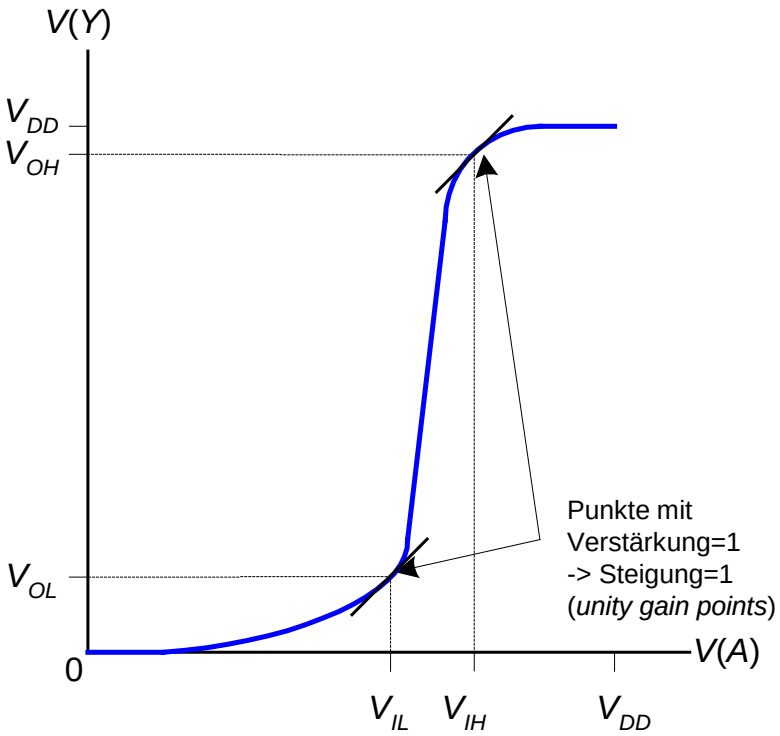
TECHNISCHE UNIVERSITÄT DARMSTADT

## Idealer Buffer:



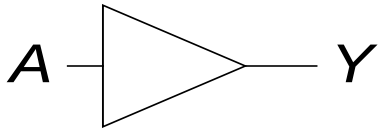

$NM_H = NM_L = V_{DD}/2$

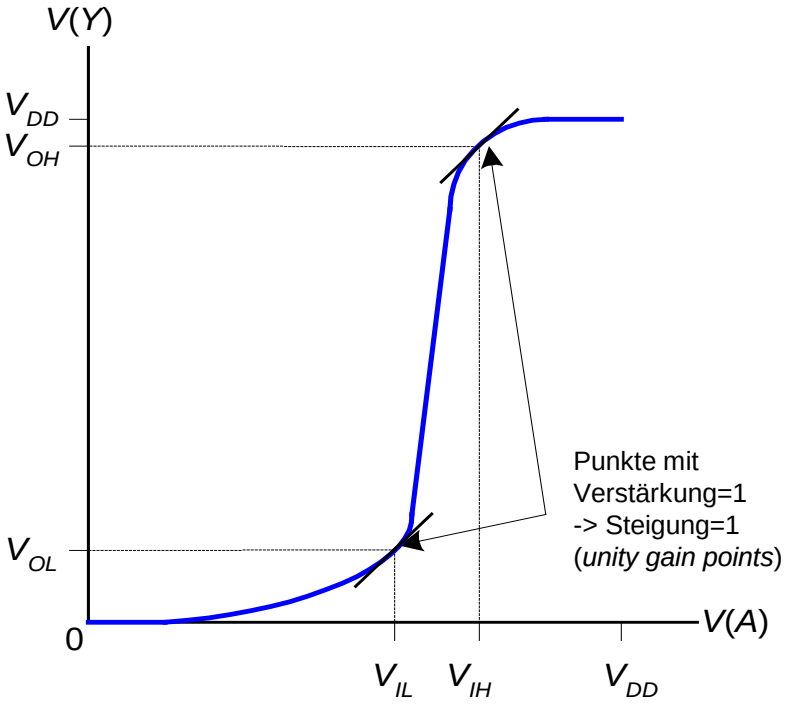
## Realer Buffer:

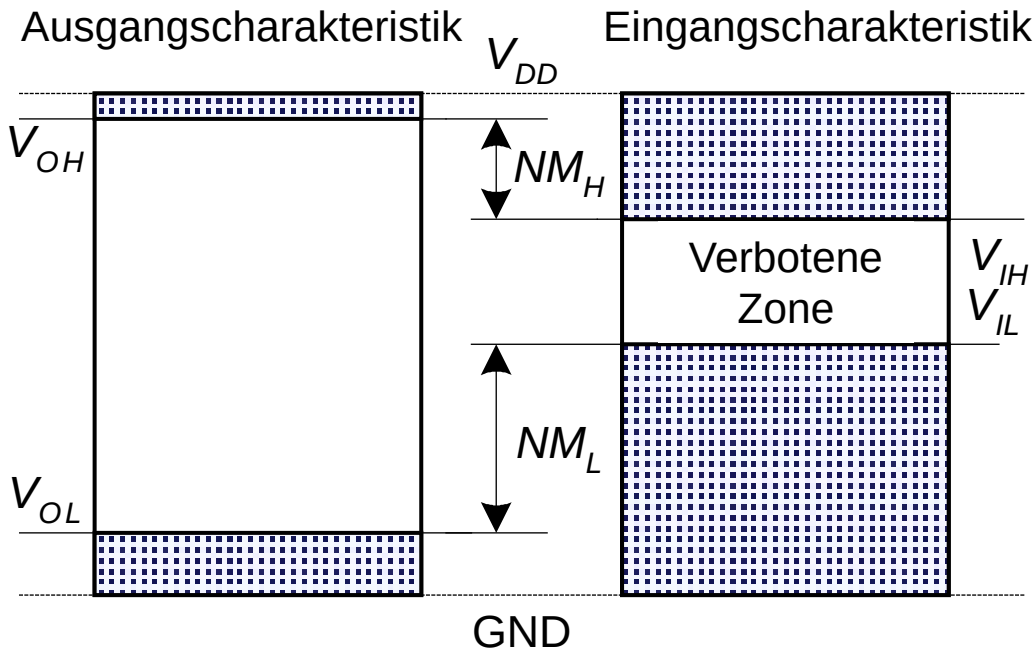



$NM_H , NM_L < V_{DD}/2$

# Gleichstrom-Transferkurve



# Absenken der Versorgungsspannung $V_{DD}$

TECHNISCHE UNIVERSITÄT DARMSTADT

- Versorgungsspannung in den 70er-80er Jahren: $V_{DD} = 5$ V
- Verbesserte Chip-Fertigungstechnologie erforderten Absenkung von $V_{DD}$
  - Hohe Spannungen würden nun sehr kleine Transistoren beschädigen
  - Energiebedarf reduzieren
- 3.3 V, 2.5 V, 1.8 V, 1.5 V, 1.2 V, 1.0 V, …
- Vorsicht beim Verbinden von Chips mit unterschiedlichen Versorgungsspannungen!

# Beispiele für Logikfamilien

TECHNISCHE UNIVERSITÄT DARMSTADT

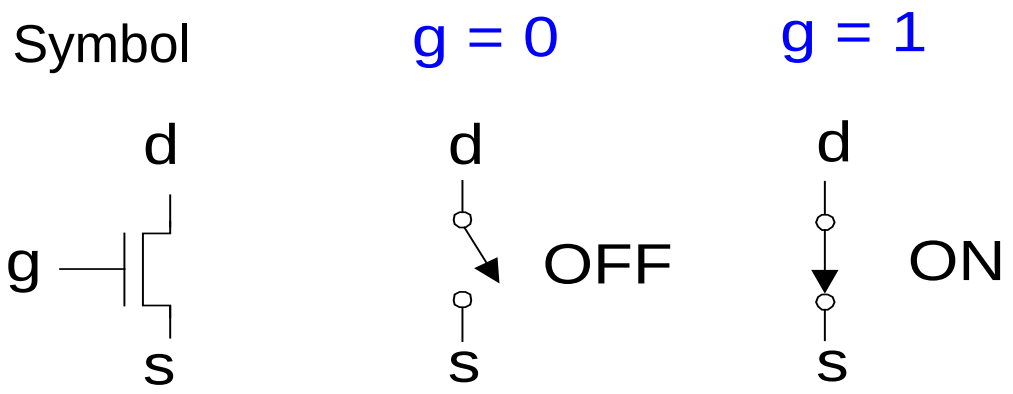
- Bausteine mit kompatiblen Spannungspegeln

| Logikfamilie | $V_{DD}$ | $V_{IL}$ | $V_{IH}$ | $V_{OL}$ | $V_{OH}$ |
|---|---|---|---|---|---|
| **TTL** | 5 (4.75 - 5.25) | 0.8 | 2.0 | 0.4 | 2.4 |
| **CMOS** | 5 (4.5 - 6) | 1.35 | 3.15 | 0.33 | 3.84 |
| **LVTTL** | 3.3 (3 - 3.6) | 0.8 | 2.0 | 0.4 | 2.4 |
| **LVCMOS** | 3.3 (3 - 3.6) | 0.9 | 1.8 | 0.36 | 2.7 |

# Transistoren

TECHNISCHE UNIVERSITÄT DARMSTADT

- Logikgatter werden üblicherweise aus Transistoren aufgebaut
  - Heute überwiegend Feldeffekttransistoren (FET)
  - Weiteres bezieht sich implizit auf FETs, nicht Bipolartransistoren
- Transistoren sind spannungsgesteuerte Schalter
  - Zwei Anschlüsse werden abhängig von Spannung an einem dritten geschaltet
    - Verbunden oder getrennt
  - **Beispiel:** Verbindung zwischen d,s verbunden wenn g=1, getrennt wenn g=0



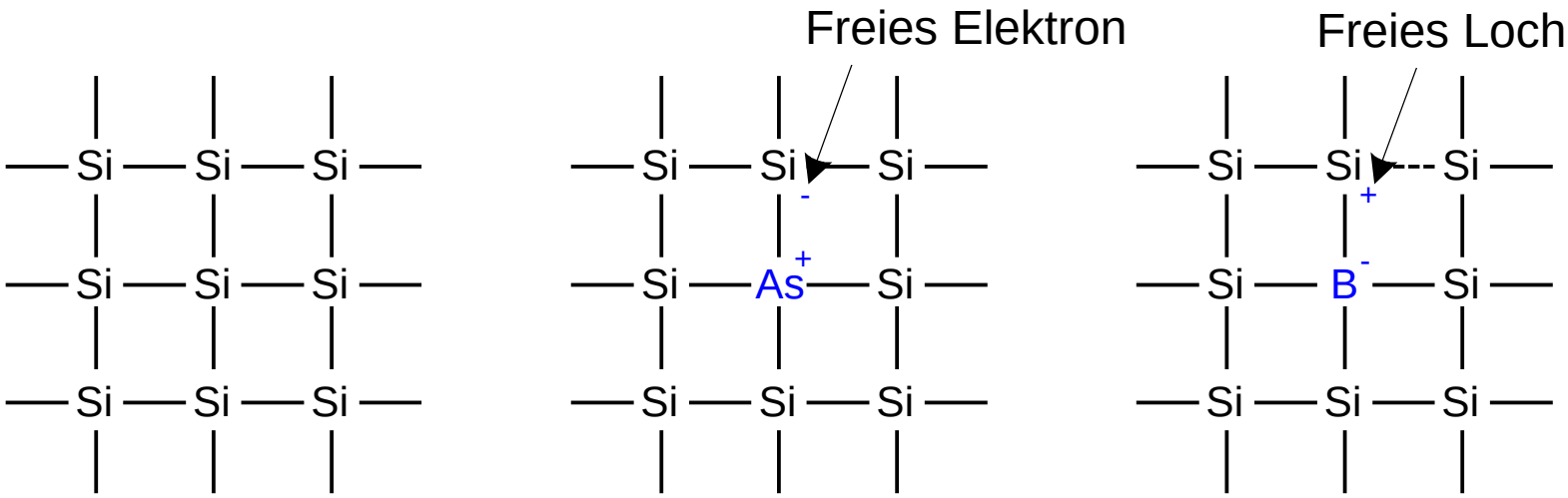
# Robert Noyce, 1927 - 1990

TECHNISCHE UNIVERSITÄT DARMSTADT

- Spitzname “Bürgermeister von Silicon Valley”
- Mitgründer von Fairchild Semiconductor in 1957
- Mitgründer von Intel in 1968
- Miterfinder der integrierten Schaltung

# Silizium

- Transistoren werden üblicherweise aus Silizium (Si, Gruppe IV) gefertigt
- Reines Silizium ist ein schlechter Leiter (keine freien Ladungsträger)
- Dotiertes Silizium ist ein guter Leiter (freie Ladungsträger)
  - n-Typ (freie ***n***egative Ladungsträger, Elektronen, dotiert mit Arsen, Gruppe V)
  - p-Typ (freie ***p***ositive Ladungsträger, Löcher, dotiert mit Bor, Gruppe III)

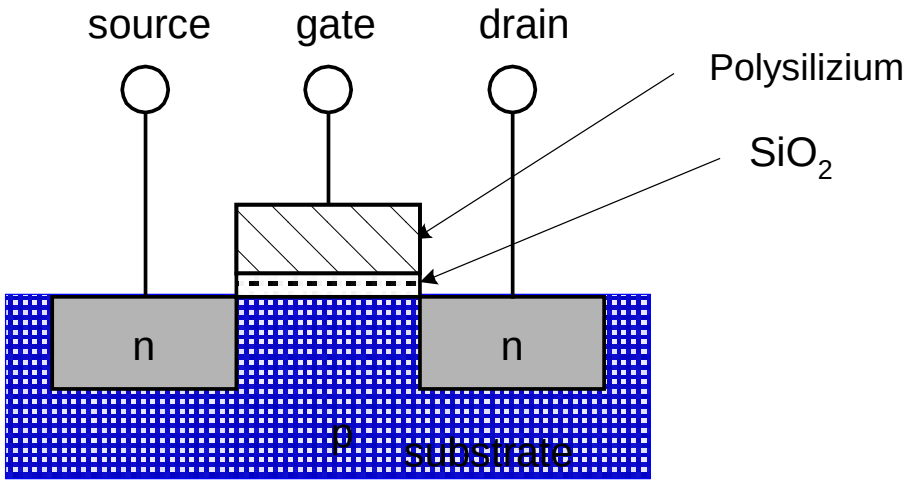


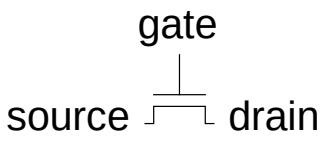

**Silizium-Atomverbund** **n-Typ** **p-Typ**

# MOS Feldeffekttransistoren (MOSFETs)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Metalloxid-Silizium (MOS) Transistoren
  - Polysilizium (früher **Metallschicht**) Gate
  - **Oxid** (Siliziumdioxid = Glas) als Isolator
  - Dotiertes Silizium




nMOS

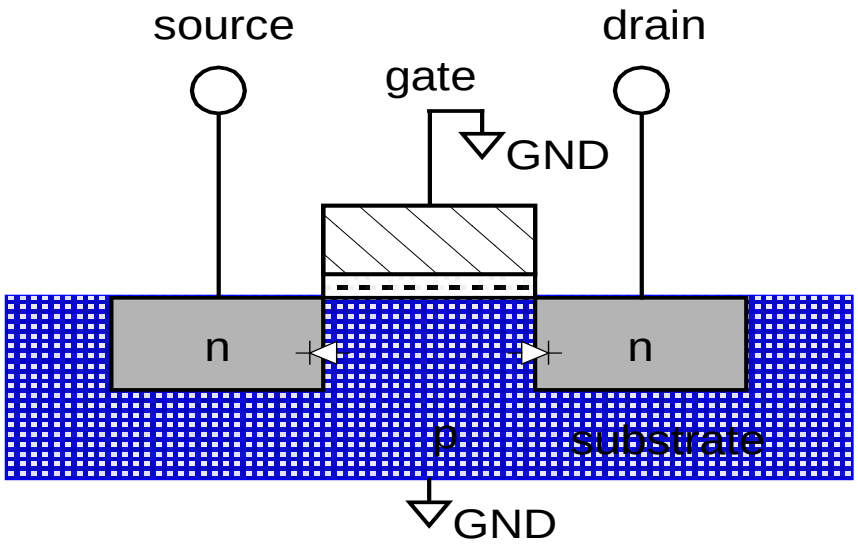
# Transistor: nMOS

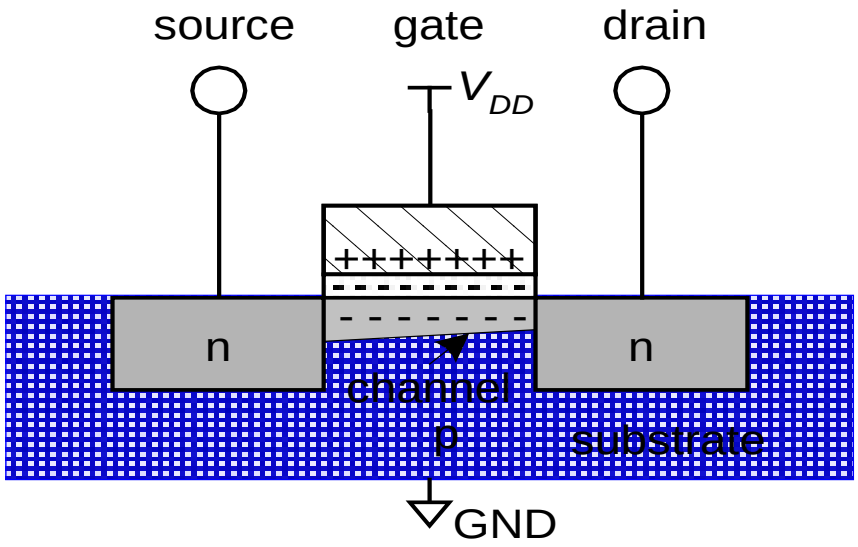
**Gate = 0**, ausgeschaltet

- keine Verbindung zwischen Source und Drain

**Gate = 1**, eingeschaltet

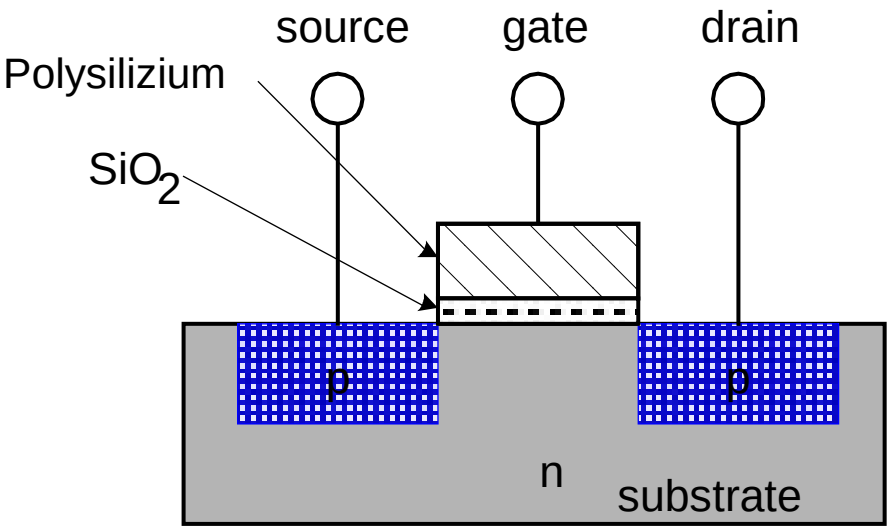
- leitfähiger Kanal zwischen Source und Drain)

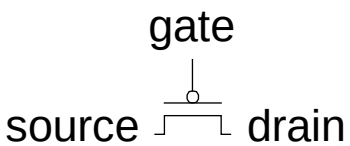


# Transistor: pMOS

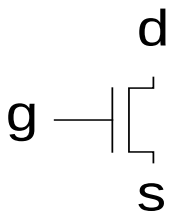
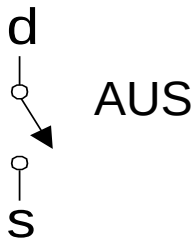
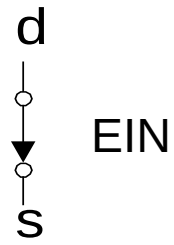
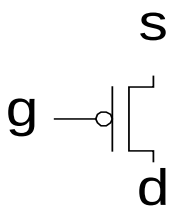
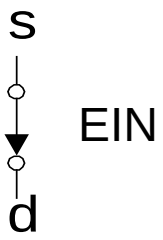
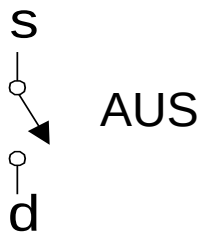
- Verhalten von pMOS Transistor ist genau **umgekehrt**
  - **EIN** wenn **Gate = 0**
  - **AUS** wenn **Gate = 1**



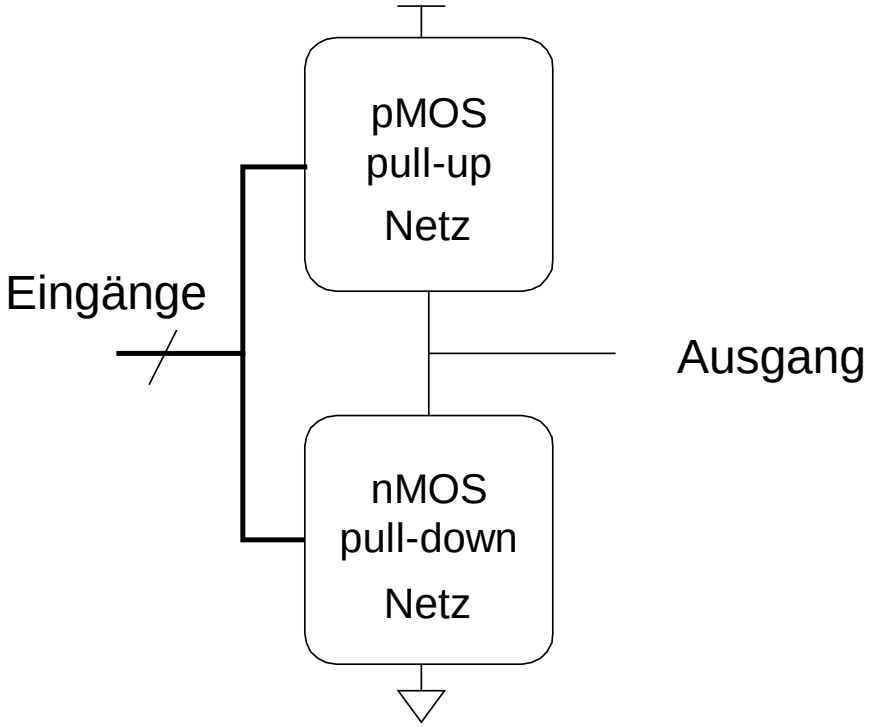
# Übersicht über Funktion von Transistoren

TECHNISCHE UNIVERSITÄT DARMSTADT

| | | g = 0 | g = 1 |
|---|---|---|---|
| nMOS |  |  |  |
| pMOS |  |  |  |

# Kombinieren von komplementären Transistoren (CMOS)

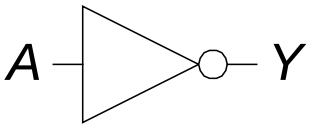
- nMOS Transistoren leiten 0'en gut zwischen S und D weiter
  - 1'en werden abgeschwächt ➔ S an GND anschließen
- pMOS Transistoren leiten 1'en gut zwischen S und D weiter
  - 0'en werden abgeschwächt ➔ S an $V_{DD}$ anschließen



# CMOS Gatter: NOT

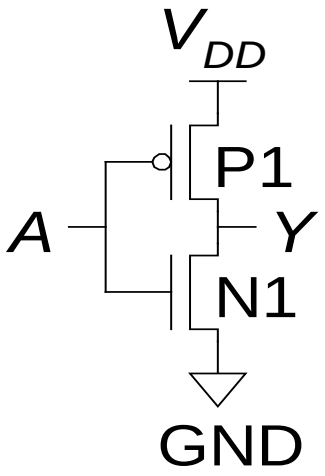
TECHNISCHE UNIVERSITÄT DARMSTADT

## NOT




$Y = \overline{A}$

| A | Y |
|---|---|
| 0 | 1 |
| 1 | 0 |

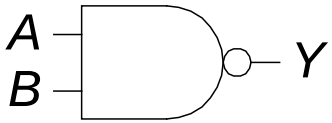



| A | P1 | N1 | Y |
|---|---|---|---|
| 0 | | | |
| 1 | | | |

# CMOS Gates: NAND Gate

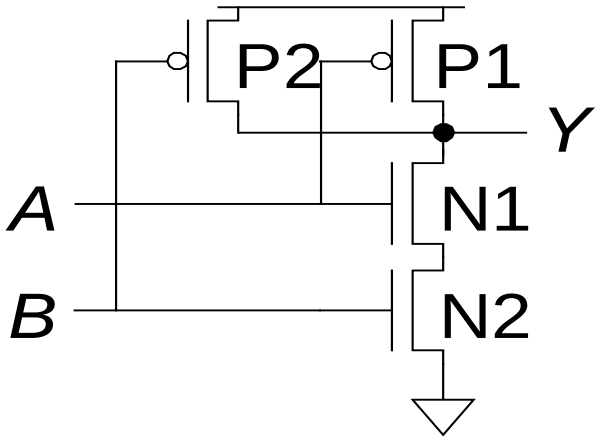
TECHNISCHE UNIVERSITÄT DARMSTADT

## NAND



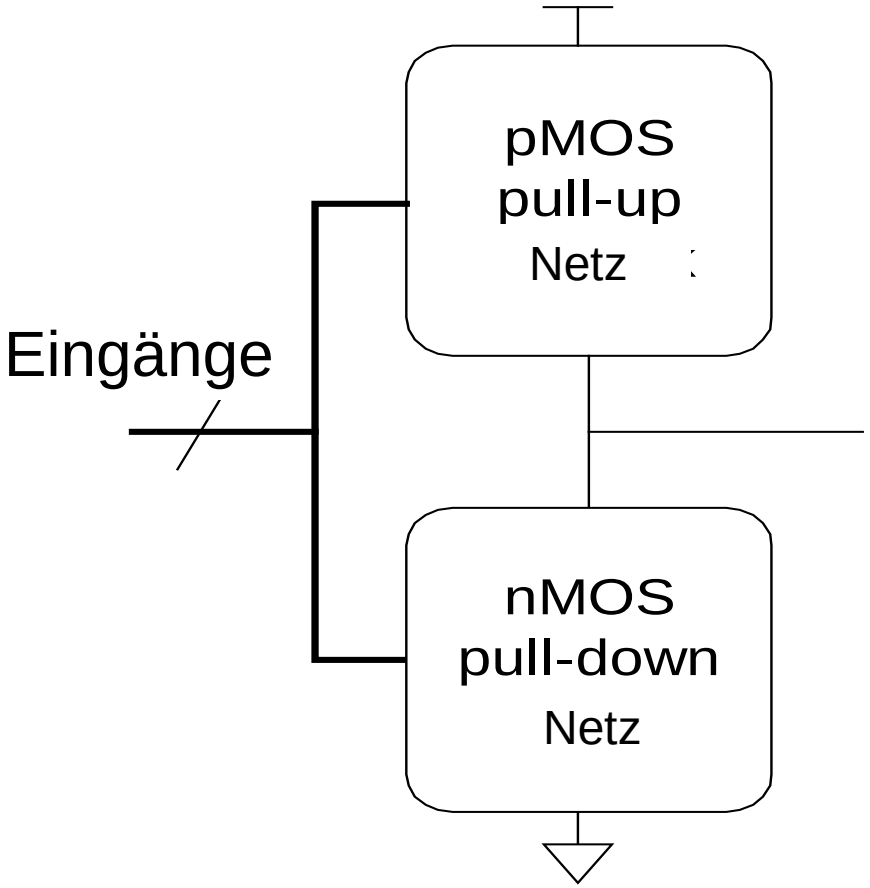

$Y = \overline{AB}$

| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |




| A | B | P1 | P2 | N1 | N2 | Y |
|---|---|---|---|---|---|---|
| 0 | 0 | | | | | |
| 0 | 1 | | | | | |
| 1 | 0 | | | | | |
| 1 | 1 | | | | | |

# Struktur eines CMOS Gatters




Wenn p-Transistoren in **Reihenschaltung** dann n-Transistoren in **Parallelschaltung**.

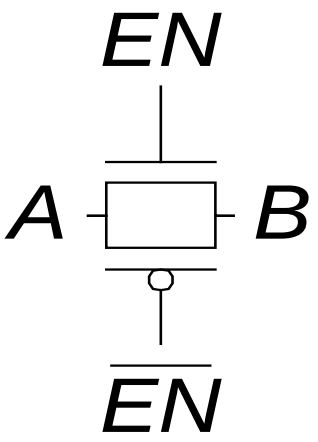
Wenn n-Transistoren in **Reihenschaltung** dann p-Transistoren in **Parallelschaltung**.

# Aufbau eines NOR-Gatters mit drei Eingängen

# Aufbau eines AND-Gatters mit zwei Eingängen

# Transmissionsgatter (*transmission gates*)

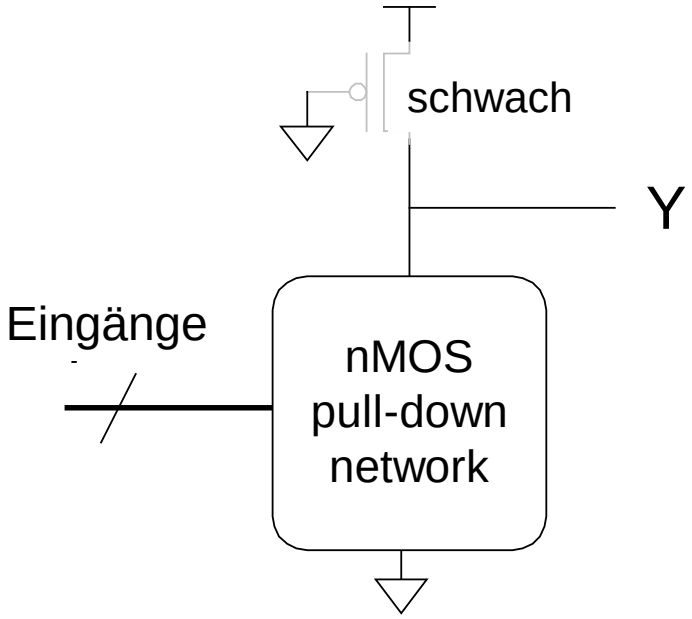
- nMOS leiten 1'en schlecht weiter
- pMOS leiten 0'en schlecht weiter
- Transmissionsgatter ist ein besserer Schalter
  - Leitet 0 und 1 gut weiter
- Wenn $EN = 1$, Schalter ist EIN:
  - $\overline{EN} = 0$
  - $A$ ist verbunden mit $B$
- Wenn $EN = 0$, Schalter ist AUS:
  - $\overline{EN} = 1$
  - $A$ ist nicht verbunden mit $B$



# Tricks: Pseudo-nMOS Gatter

TECHNISCHE UNIVERSITÄT DARMSTADT

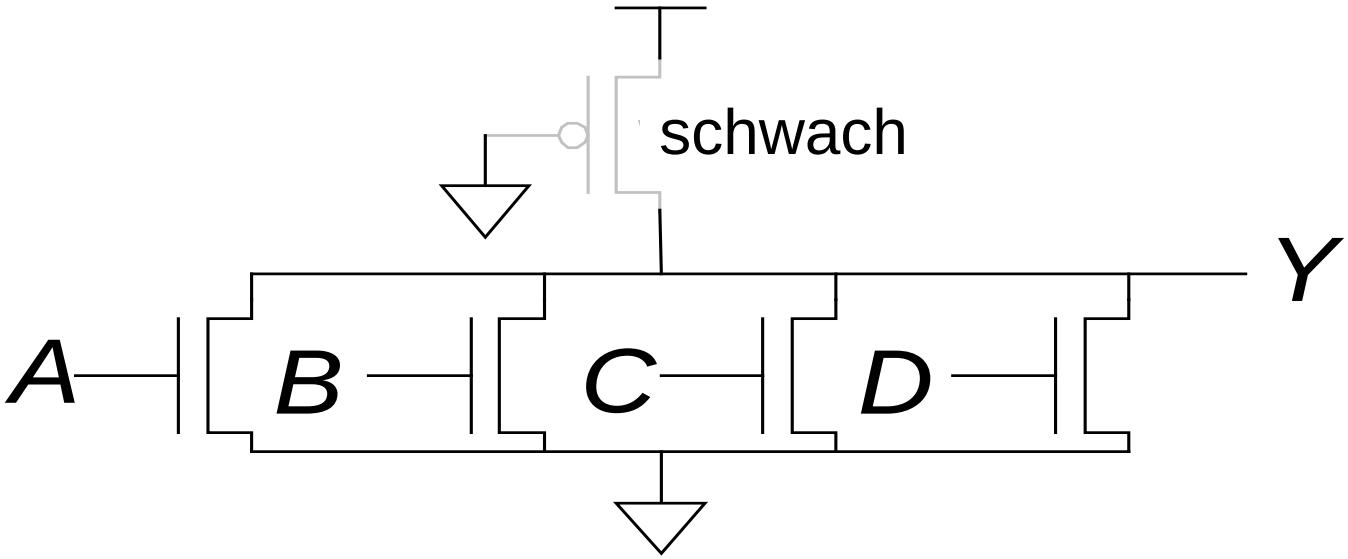
- Pseudo-nMOS Gatter ersetzen das Pull-Up Netz
- Durch schwachen immer eingeschalteten pMOS Transistor
  - Schwach heißt: Seine 1 kann durch das Pull-Down Netz neutralisiert werden
- Nützlich um lange Reihen von Transistoren zu vermeiden: breite NORs



# Beispiel für Pseudo-nMOS Gatter

TECHNISCHE UNIVERSITÄT DARMSTADT

Pseudo-nMOS NOR4

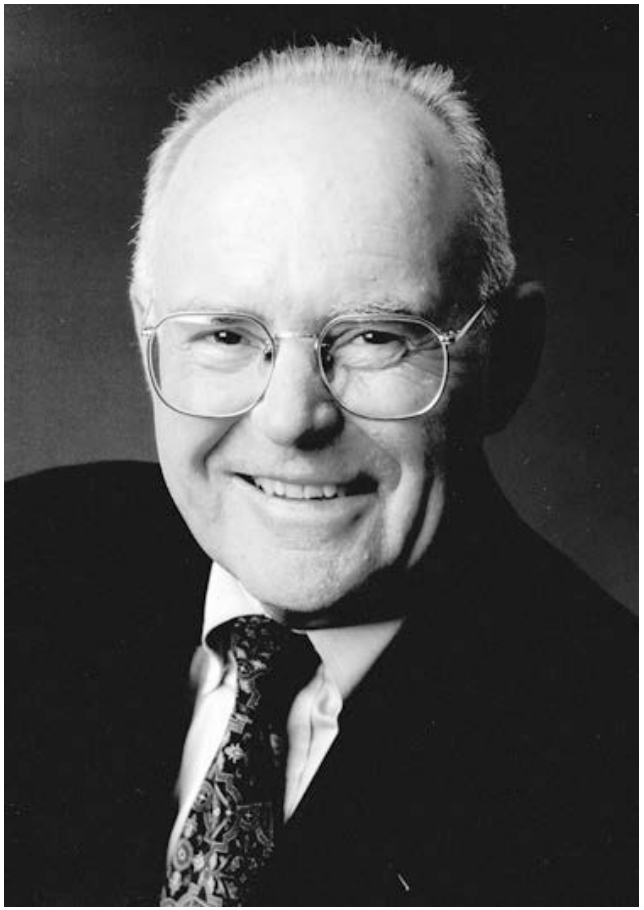



Verbraucht aber mehr Energie: Schwacher Dauerkurzschluss bei Y=0

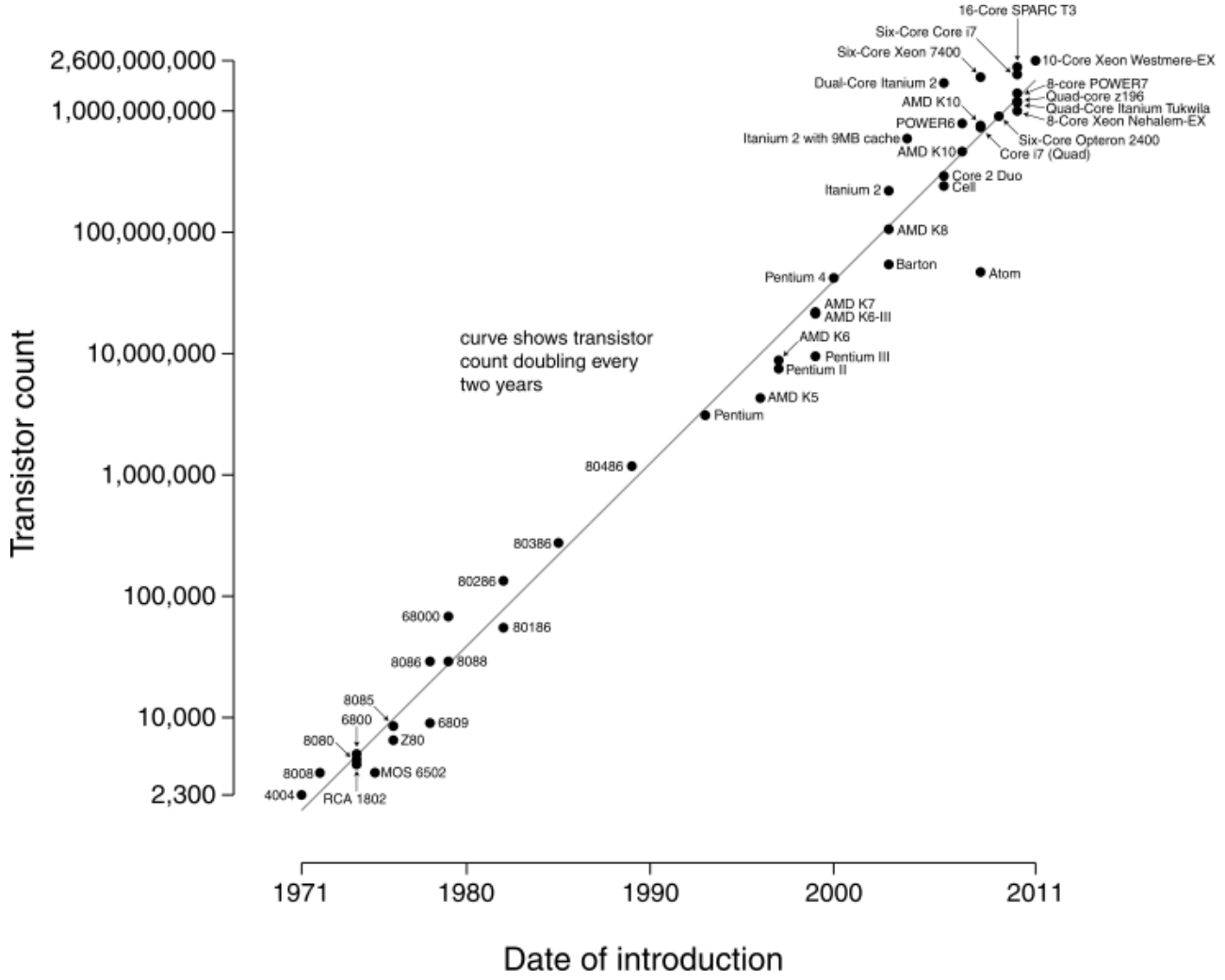
# Gordon Moore, 1929 -

TECHNISCHE UNIVERSITÄT DARMSTADT

- Gründete Intel in 1968 zusammen mit Robert Noyce
- **Moores Gesetz:** Die Anzahl von Transistoren auf Chips verdoppelt sich
  - Jedes Jahr (1965)
  - Alle zwei Jahre (angepasst 1975)

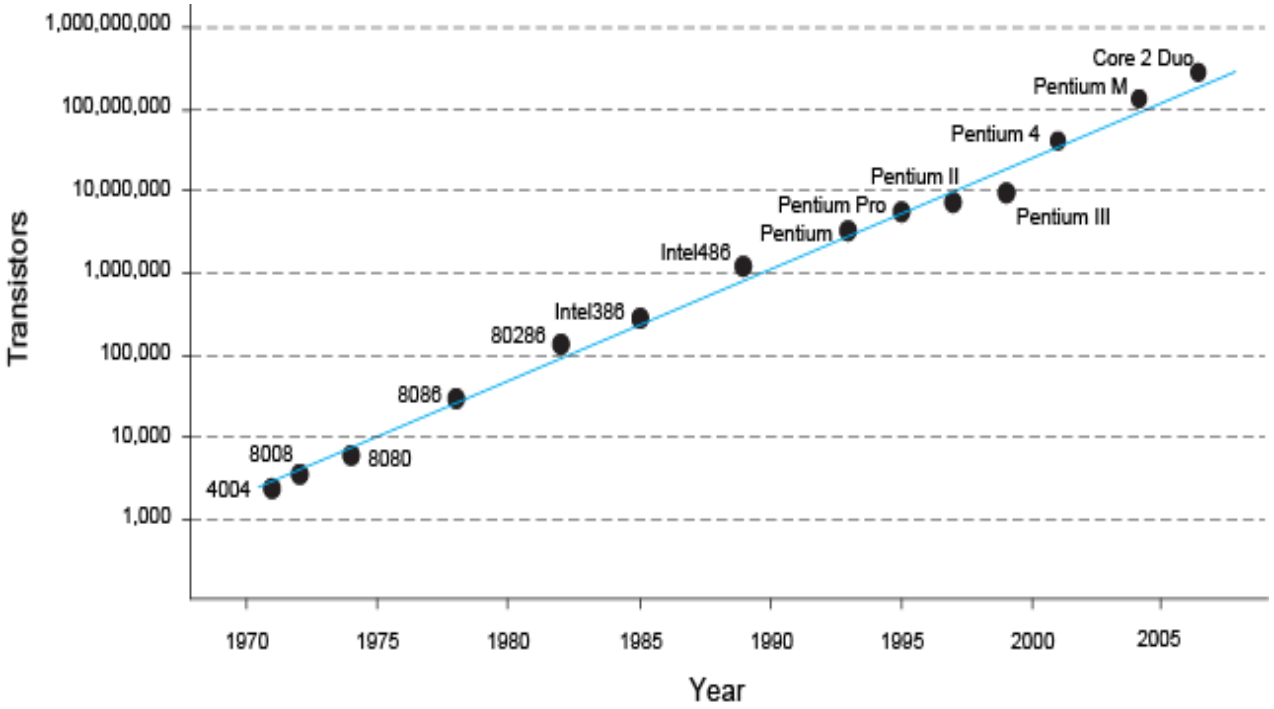


# Moores Gesetz




Quelle: Wikipedia

# Moores Gesetz




- *"Wenn sich das Auto wie die Computer entwickelt hätte, würde ein Rolls-Royce heute $100 kosten, 250 µl Benzin auf 100 km verbrauchen und einmal im Jahr explodieren …"*

  *– Robert X. Cringely (Infoworld)*

# Leistungsaufnahme

TECHNISCHE UNIVERSITÄT DARMSTADT

- Leistung = Energieverbrauch pro Zeiteinheit
- Zwei Arten der Leistungsaufnahme:
  - Dynamische Leistungsaufnahme
  - Statische Leistungsaufnahme

# Dynamische Leistungsaufnahme

TECHNISCHE UNIVERSITÄT DARMSTADT

- Leistung um Gates der Transistoren umzuladen
  - Wirken als Kondensator
- Leistung um einen Kondensator der Kapazität $C$ auf $V_{DD}$ zu laden:
  - $CV_{DD}^2$
- Schaltung wird mit Frequenz $f$ betrieben
  - Transistoren schalten $f$-mal pro Sekunde
  - Aber nicht alle Transistoren schalten jeden Takt um 0-1-0 um
  - Annahme: Jeden Takt nur Laden oder Entladen
    - Halbe Leistungsaufnahme (realistischer wäre 0,1)
- Die dynamische Leistungsaufnahme ist also:

$$P_{dynamic} = \tfrac{1}{2} C V_{DD}^2 f$$

# Statische Leistungsaufnahme

TECHNISCHE UNIVERSITÄT DARMSTADT

- Leistungsbedarf wenn kein Gatter schaltet
- Wird verursacht durch den Leckstrom $I_{DD}$
  - Immer kleinere Transistoren schalten nicht mehr vollständig ab
  - Pseudo-nMOS, …
- Statische Leistungsaufnahme ist also

$$P_{static} = I_{DD} V_{DD}$$

# Beispielrechnung Leistungsaufnahme

TECHNISCHE UNIVERSITÄT DARMSTADT

- Abschätzen der Leistungsaufnahme für ein Netbook
- Parameter
  - Versorgungsspannung $V_{DD}$ = 1,2 V
  - Transistorkapazität $C$ = 20 nF
  - Taktfrequenz $f$ = 1 GHz
  - Leckstrom $I_{DD}$ = 20 mA