

# **Analoge Schaltungstechnik**

Mirco Meiners              Tim Ziemann

2025-04-08



# Inhaltsverzeichnis

<b>Unsyllabus</b>	<b>1</b>
<b>I. Seminar</b>	<b>3</b>
<b>1. Einleitung und Übersicht</b>	<b>5</b>
1.1. Lernziele des Moduls . . . . .	5
1.2. Voraussetzungen des Moduls . . . . .	6
1.3. Wissenschaftliches Rechnen / Datenwissenschaft . . . . .	6
1.4. Schaltungssimulation (SPICE) . . . . .	6
1.5. Betriebssystem (OS) - Werkzeuge (Tools) . . . . .	6
1.6. Code Editoren . . . . .	7
1.7. Schreibst Du noch oder TeXst Du schon? . . . . .	7
1.8. LaTeX Editoren . . . . .	7
1.9. Literaturverwaltung und LaTeX . . . . .	7
1.10. Labor . . . . .	7
1.11. Data Science . . . . .	8
1.12. Schaltungsanalyse/-entwurf ≠ Black Magic . . . . .	8
1.13. Schöne neue Welt . . . . .	9
1.14. From Sand to Silicon (Infineon, Dresden) . . . . .	9
1.15. FinFET (Intel) . . . . .	9
1.16. TSMC Fab (Next Gen 7/5 nm) . . . . .	9
1.17. Es war einmal ... . . . . .	10
1.18. Damals und heute . . . . .	10
1.19. Systemhierarchie . . . . .	11
1.20. System Assembly . . . . .	12
1.21. Schnittstellen zur Aussenwelt . . . . .	13
1.22. Meeting mit einem System . . . . .	13
1.23. System in a Package (SiP) . . . . .	14
1.24. Backend Phasen . . . . .	14
1.25. Sie werden Experte . . . . .	14
1.26. Evolution von Produkten . . . . .	15
1.27. EDA Kompetenz . . . . .	15
1.28. Ansichten zur Hardware (I) . . . . .	16
1.29. Ansichten zur Hardware (II) . . . . .	17
1.30. Abstraktionsebenen . . . . .	18
1.31. Entwurfsablauf . . . . .	19

## *Inhaltsverzeichnis*

1.32. Verifikation . . . . .	20
1.32.1. Frontend vs. Backend (analog) . . . . .	21
1.33. Frontend vs. Backend (digital) . . . . .	22
1.34. Design-/Entwurfsmethodik . . . . .	22
1.35. Analog Design Entry . . . . .	23
1.36. Netlist . . . . .	24
1.37. Layout . . . . .	25
1.38. Digital Design Entry . . . . .	26
1.39. Hardwarebeschreibungssprachen . . . . .	27
1.40. Technology-Gates und Netlisting . . . . .	28
1.41. Standard Cell Layout . . . . .	29
1.42. Nachhaltige Elektronik ... . . . . .	29
1.43. Warum es sicht lohnt ... . . . . .	29
1.44. Und ab an den Strand ... . . . . .	30

# **Abbildungsverzeichnis**



# **Tabellenverzeichnis**



# **Unsyllabus**

Name	Description
Kursus	M 6.9 Analoge Schaltungstechnik (ANS)
Semester	SoSe25
Dozent	Prof. Dr.-Ing. M. Meiners
Seminar	Die., 9:45 Uhr bis 13 Uhr, E 507



# **Teil I.**

# **Seminar**



# 1. Einleitung und Übersicht

Im Modul **M 6.9 Analoge Schaltungen (ANS)**, im Bachelorstudiengang (B.Eng.) Elektrotechnik (ET) der Fakultät 4 der Hochschule Bremen (HSB), werden aus der Theorie des Schaltungsentwurfs gewonnene Kenntnisse im Labor experimentell erprobt ([ASLK Pro Experimente](#)) und mit **KiCAD** als eigene PCB-Designs umgesetzt. Für die erste Erprobung von Schaltungen wird das **ASLK (Analog System Lab Kit) Pro** der Firma [MikroElektronika](#) eingesetzt.

Zur Simulation werden Schaltungssimulatoren wie **LTS spice** und [ngspice](#) verwendet.

Des Weiteren kommen die [MATLAB Campus-Lizenz](#) und [Python](#) als leistungsstarke Instrumente zur Modellierung und Verhaltensanalyse von Filtern- und Verstärkern beim Schaltungsentwurf zum Einsatz.

Für die Charakterisierung der Schaltungen stehen im Labor sechs Meßplätze mit gängigen Gerätschaften wie Quellen, Signalgeneratoren, Oszilloskopen und dem integrierten Meßlabors [STEMlab](#) von RedPitaya zur Verfügung.

## 1.1. Lernziele des Moduls

- Ein tieferes Verständnis für das Verhalten von MOS-Elementen im analogen Schaltungsentwurf entwickeln und Ausblicke auf weiterführende Kurse im Master bekommen.
- Funktionsprinzipien und Charakterisierung von MOS-Elementen
- Fundamentale integrierte analoge Schaltungen
- Operationsverstärker
- Lernen, Grenzen und Tradeoffs analoger Schaltungen zu bewerten
- Entwickeln eines systematischen Entwurfsstils, auch anwendbar für andere Ingenieursdisziplinen
- Lernen, einen Schaltungssimulator für den Entwurf einzusetzen.
- Technisch-wissenschaftliche Dokumentation
- Konsolidierung der oberen Aspekte in Laborprojekten

## *1. Einleitung und Übersicht*

### **1.2. Voraussetzungen des Moduls**

- Digitaltechnik
- Grundlagen der Halbleiterbauelemente
- Netzwerk- und Systemtheorie
- Regelungstechnik

### **1.3. Wissenschaftliches Rechnen / Datenwissenschaft**

- Python
- Matlab
- Gnu Octave
- Command-line tools

### **1.4. Schaltungssimulation (SPICE)**

- LTspice Linear Technology
- ngspice (open-source)
- ELDO (Siemens EDA)
- Spectre (cadence)
- PrimeSim HSPICE (SYNOPSIS)

### **1.5. Betriebssystem (OS) - Werkzeuge (Tools)**

- Shell
  - oh-my-zsh,
  - bash-it
  - SSH (Secure Shell)
- GIT (Versionskontrolle)
- Cygwin

## 1.6. Code Editoren

- Visual Studio Code
- Notepad++ (Windows)
- Emacs
- Vim

## 1.7. Schreibst Du noch oder TeXst Du schon?

- MikTeX (Windows, MacOS, Linux)
- MacTeX (MacOS)
- TeXLive (Linux)

## 1.8. LaTeX Editoren

- IDE's
  - TeXStudio
  - TeXMaker
  - TeXWorks
- Kollaborative Frameworks
  - ShareLaTeX, Online LaTeX
  - CoCalc - Online LaTeX

## 1.9. Literaturverwaltung und LaTeX

- Citavi im Detail > Titel exportieren > Export nach BibTeX
- RefWorks - Library Guide Univ. Melbourne
- Benutzerdefinierte BibTex-Keys mit Zotero | nerdspause
- JabRef - Library Guide Univ. Melbourne
- EndNote - Library Guide Univ. Melbourne

## 1.10. Labor

- Anwendung unterschiedlicher Beschreibungsebenen
  - Systemebene (Verhaltensmodell z.B. mit Matlab/Simulink oder Python)
  - Schaltungsebene (SPICE)

## *1. Einleitung und Übersicht*

- Charakterisierung (Messungen)
- Analog System Lab Kit – [ASLK Pro](#)
- Messautomatisierung Red Pitaya [STEMlab 125-14/10](#)

## **1.11. Data Science**

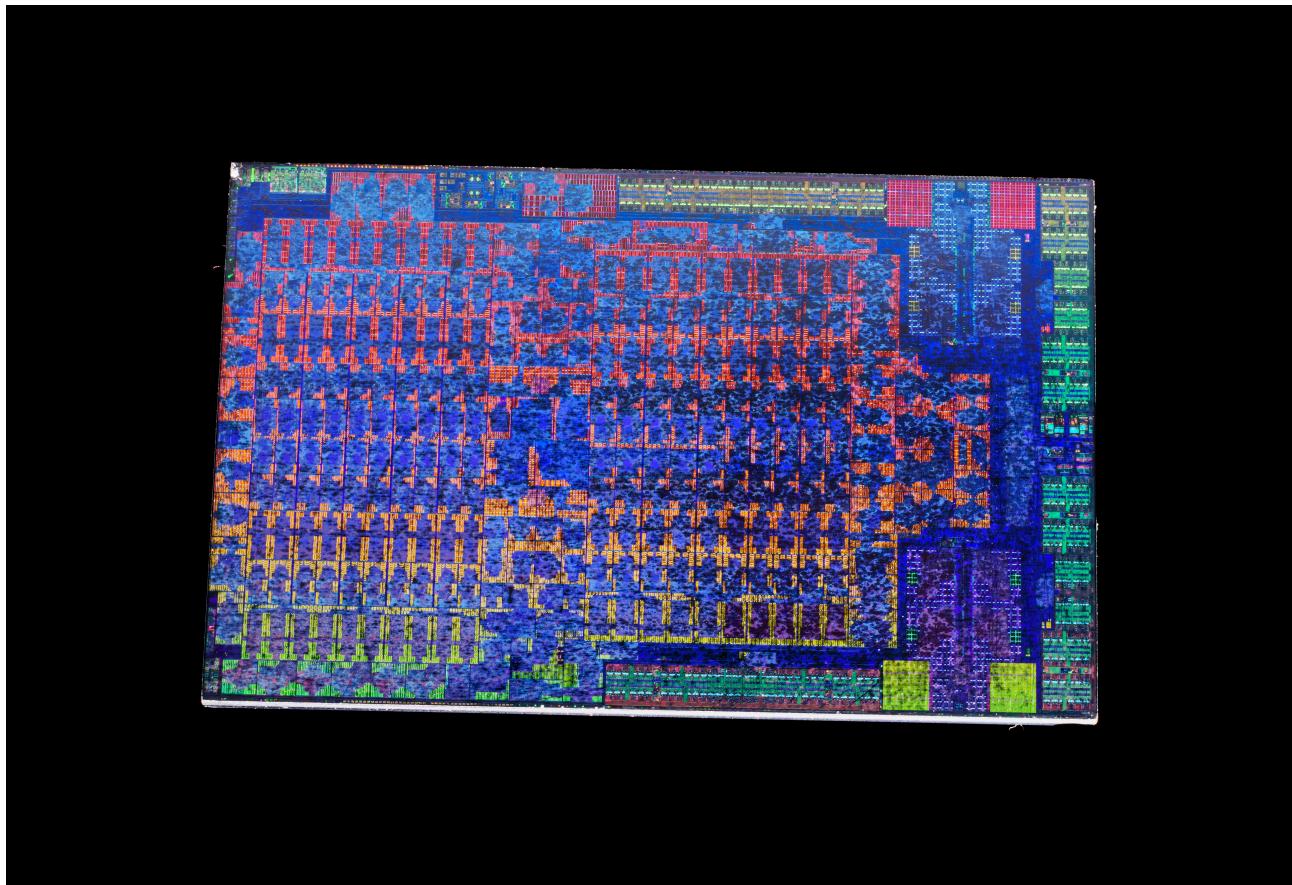
- Arbeitsordner auf dem Rechner (sandboxing, virtualenv)
- Tabellenformat (ASCII, CSV)
- Exceldatei (.xlsx) oder OpenDocument (.ods)
- Datenspeicherung in speziellen Formaten, z.B. Apache Parquet
- Datenanalyse mit [Datenbanken](#), z.B. [DuckDB](#))

## **1.12. Schaltungsanalyse/-entwurf ≠ Black Magic**

*Schaltungsanalyse.* \* ist die Fähigkeit, Schaltungen in handhabbare Teile zu zerlegen \* basierend auf einem einfachen, aber hinreichend genauen Modell \* “Just-in-time” Modellierung: Verwende kein komplexes Modell, so lange es nicht benötigt wird! \* eine Schaltung ⇒ eine Lösung

*Schaltungsdesign/-entwurf.* \* ist die Fähigkeit der Schaltungssynthese auf Basis von Erfahrung und intensiver Analyse. \* eine Spezifikation ⇒ viele Lösungen \* Entwurfspraktiken werden am besten durch's "Selbermachen" ausgebildet – "Machen ist wie wollen nur krasser."

### 1.13. Schöne neue Welt



### 1.14. From Sand to Silicon (Infineon, Dresden)

[https://www.youtube.com/embed/bor0qLifjz4?list=PLO\\_wT97BGA6xC6hNy9VGtt1bKwVuQXI5B](https://www.youtube.com/embed/bor0qLifjz4?list=PLO_wT97BGA6xC6hNy9VGtt1bKwVuQXI5B)

### 1.15. FinFET (Intel)

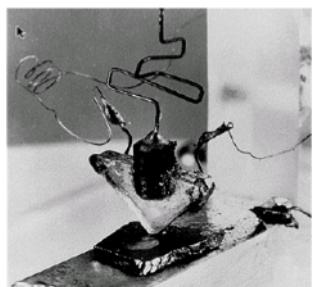
[https://www.youtube.com/embed/\\_VMYPLXnd7E](https://www.youtube.com/embed/_VMYPLXnd7E)

### 1.16. TSMC Fab (Next Gen 7/5 nm)

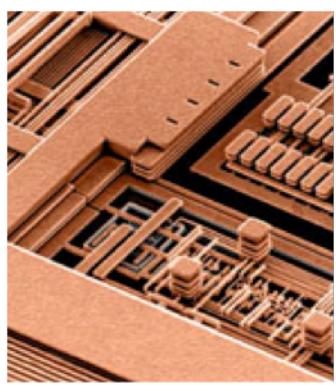
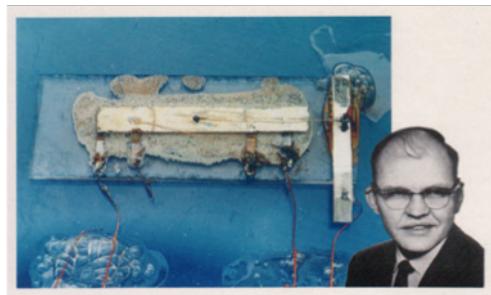
<https://www.youtube.com/embed/Hb1WDxSoSec>

1. Einleitung und Übersicht

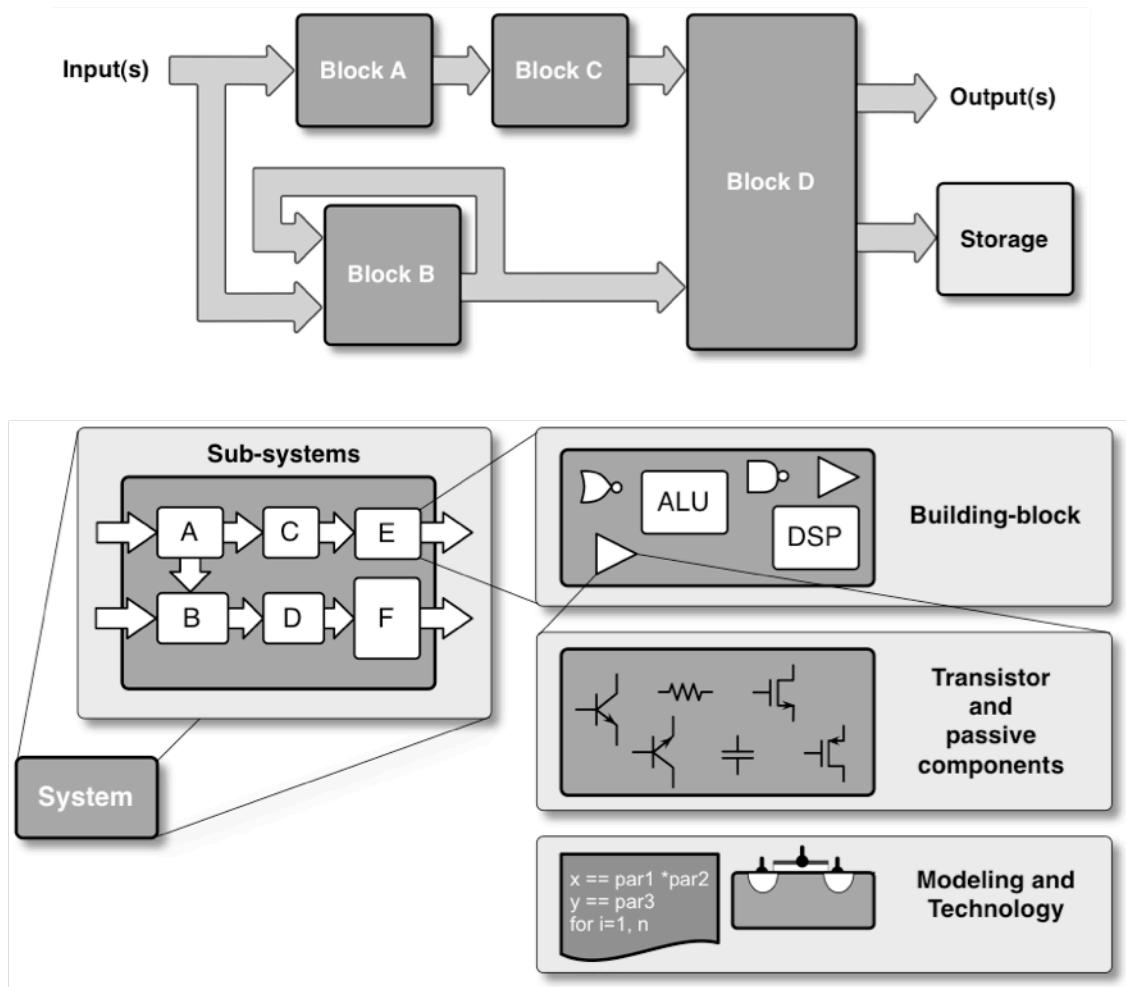
**1.17. Es war einmal ...**



**1.18. Damals und heute**



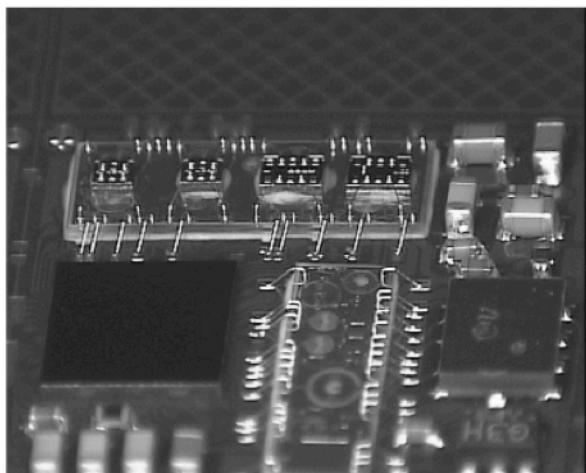
## 1.19. Systemhierarchie



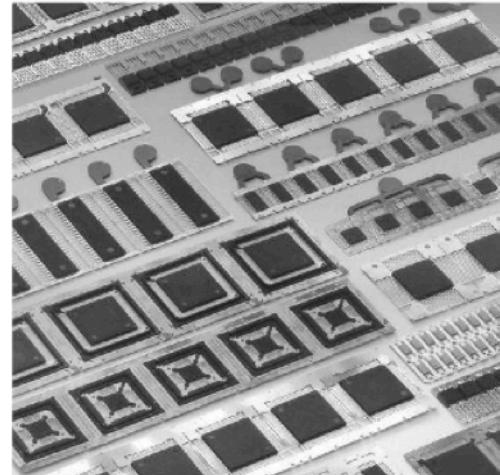
- Nutzen Sie Hierarchien zur Beschreibung komplexer Systeme
- Teile und herrsche

1. Einleitung und Übersicht

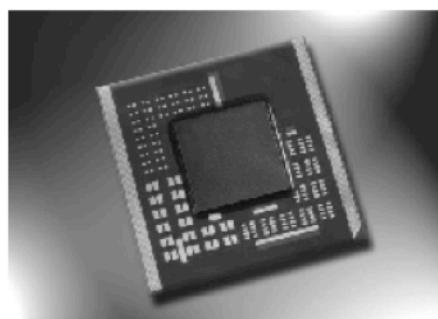
## 1.20. System Assembly



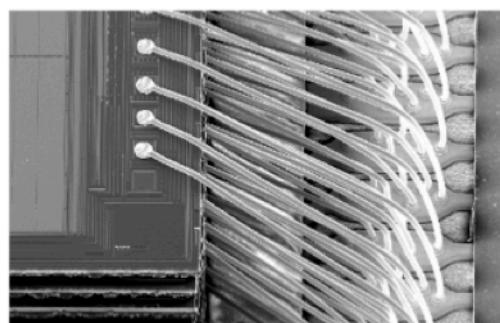
(a)



(b)

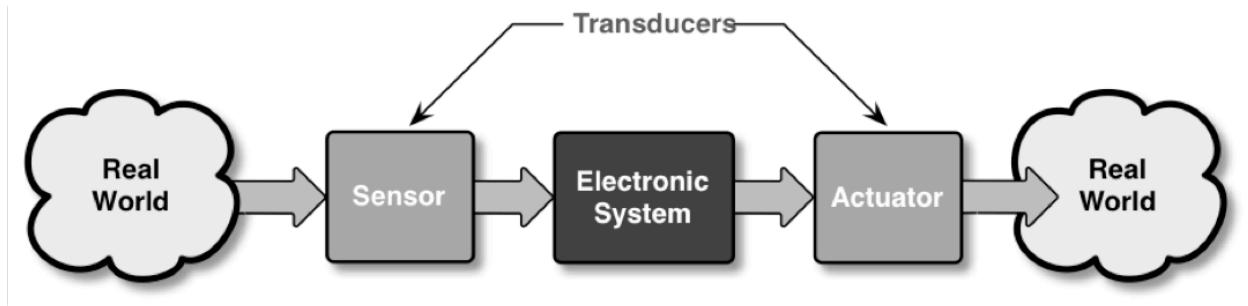


(c)



(d)

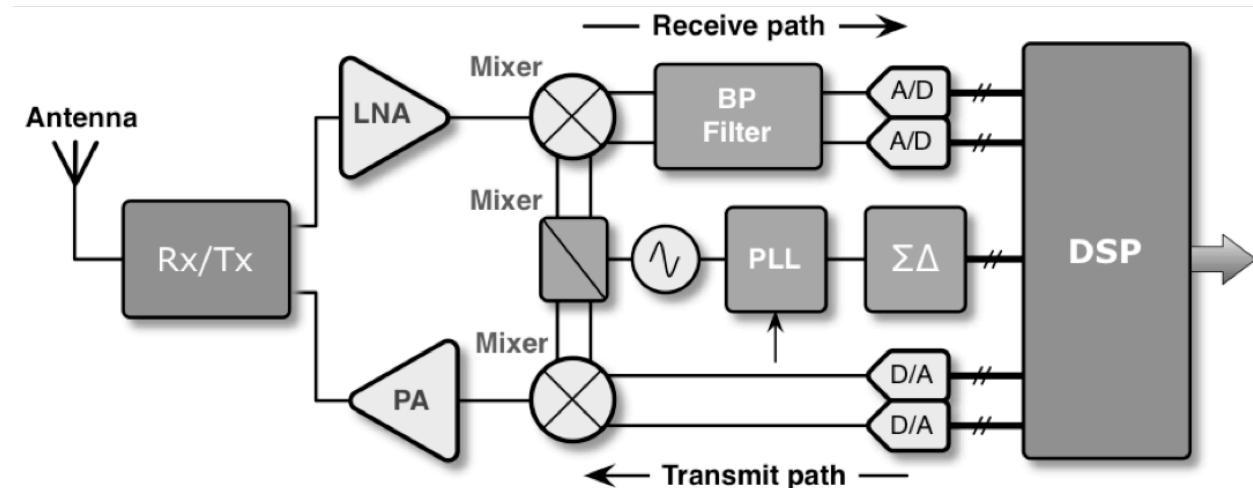
## 1.21. Schnittstellen zur Aussenwelt



Entire system involving signals of real world.

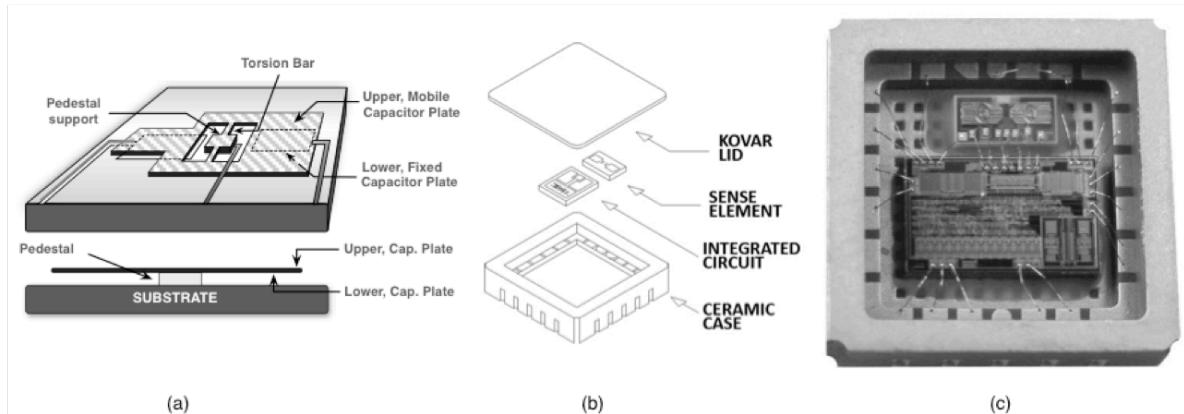
## 1.22. Meeting mit einem System

Block diagram of a wireless communication system



## 1. Einleitung und Übersicht

### 1.23. System in a Package (SiP)



- (a) Micro structure of an accelerometer.
- (b) Assembling diagram of the system-on- package.
- (c) Microphotograph. (*Courtesy of Silicon Designs, Inc.*).

### 1.24. Backend Phasen

- Packaging
- Zuverlässigkeit = Qualität auf Zeit
- Testing auf Wafer Level, known good die (KGD)
  - Burn-in und Accelerated Aging (thermischer Stress, Arrhenius Gesetz)
  - Automatic Test Equipment (ATE)
    - \* System Probe
    - \* Interconnect Test
    - \* Build-in Self-Test (BIST)
- Statistische Datenanalyse und Yield Prognosen
  - Ausfallrate FIT (failure in time)
  - Badewannenkurve

### 1.25. Sie werden Experte

Leistungsmerkmale. \* Hintergrundwissen \* Systemverständnis, Architektur, Herstellungsverfahren, Implementation

- Unterbewusste Kompetenz
  - Abgespeicherte Erfahrungen aus Erfolgsgeschichten und Misserfolgen
- Spezialwissen
  - Berufsspezifisches Wissen
- Teamwork Haltung
  - Kommunikationsfähigkeit, Berichtswesen und technische Präsentation
- Kreativität
- Tool-Kenntnisse

## 1.26. Evolution von Produkten

- Angetrieben durch Technologieverbesserung
  - Kosten (größere Chips, geringere Größe der Merkmale, bessere Ausbeute)
  - Leistung (neue Bauteile, höhere Geschwindigkeit, weniger Stromverbrauch)
- Angetrieben durch Verbesserung der Entwurfsmethodik
  - Architektur (Leistung, Funktionen)
  - CAD (Entwicklungskosten, Time-to-Market)
- Komplexität der Designs verdoppelt sich jedes Jahr (Moore's Gesetz)
- Rolle von CAD
  - Verbesserung der Produktivität von Konstruktionsprozessen
  - Reduzierung der Komplexität für den Konstrukteur
  - Sicherstellung des ordnungsgemäßen Betriebs der Geräte

## 1.27. EDA Kompetenz

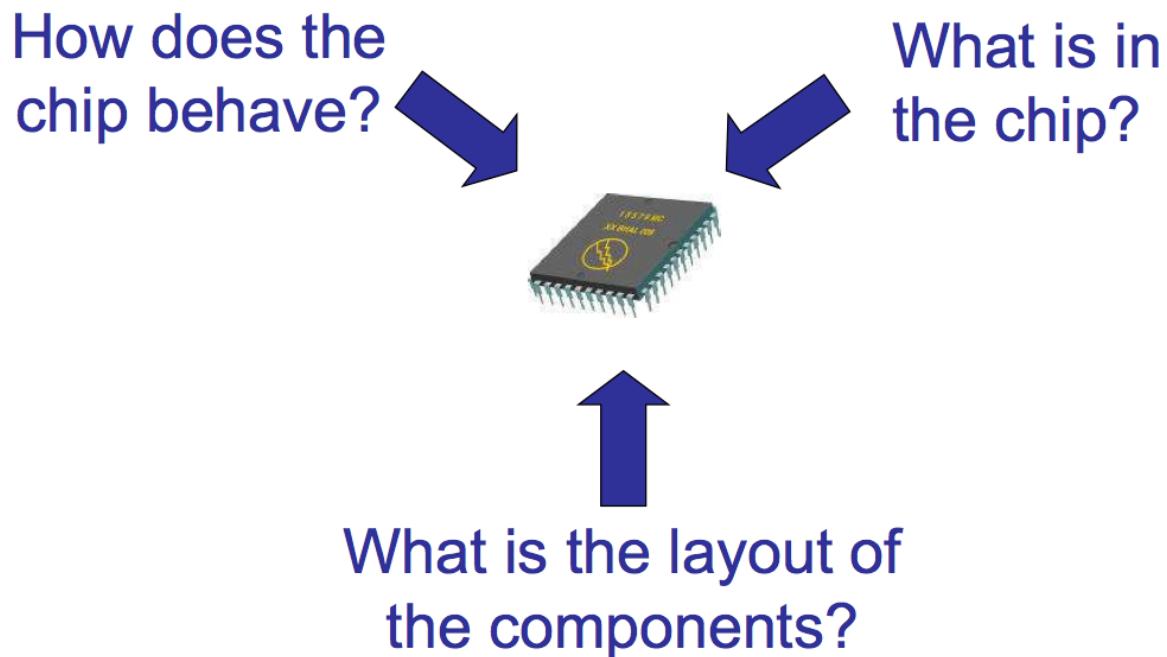
- EDA-Anbieter (Tool-Entwickler)
  - Identifikation von Entwurfsaufgaben, Bedarf an Werkzeugen
  - Entwicklung von Strategien und Algorithmen
  - Implementierung von Software-Werkzeugen
  - Verifikation der Stabilität und Funktionalität der Software-Tools
- IC-Hersteller
  - Entscheidungsplanung, welches Tool die Produktivität steigern könnte
  - EDA-Tool-Manager, Installation und Wartung
  - Experten für Softwareeinsatz, Anwendung in Produktdesign und -entwicklung

## 1. Einleitung und Übersicht

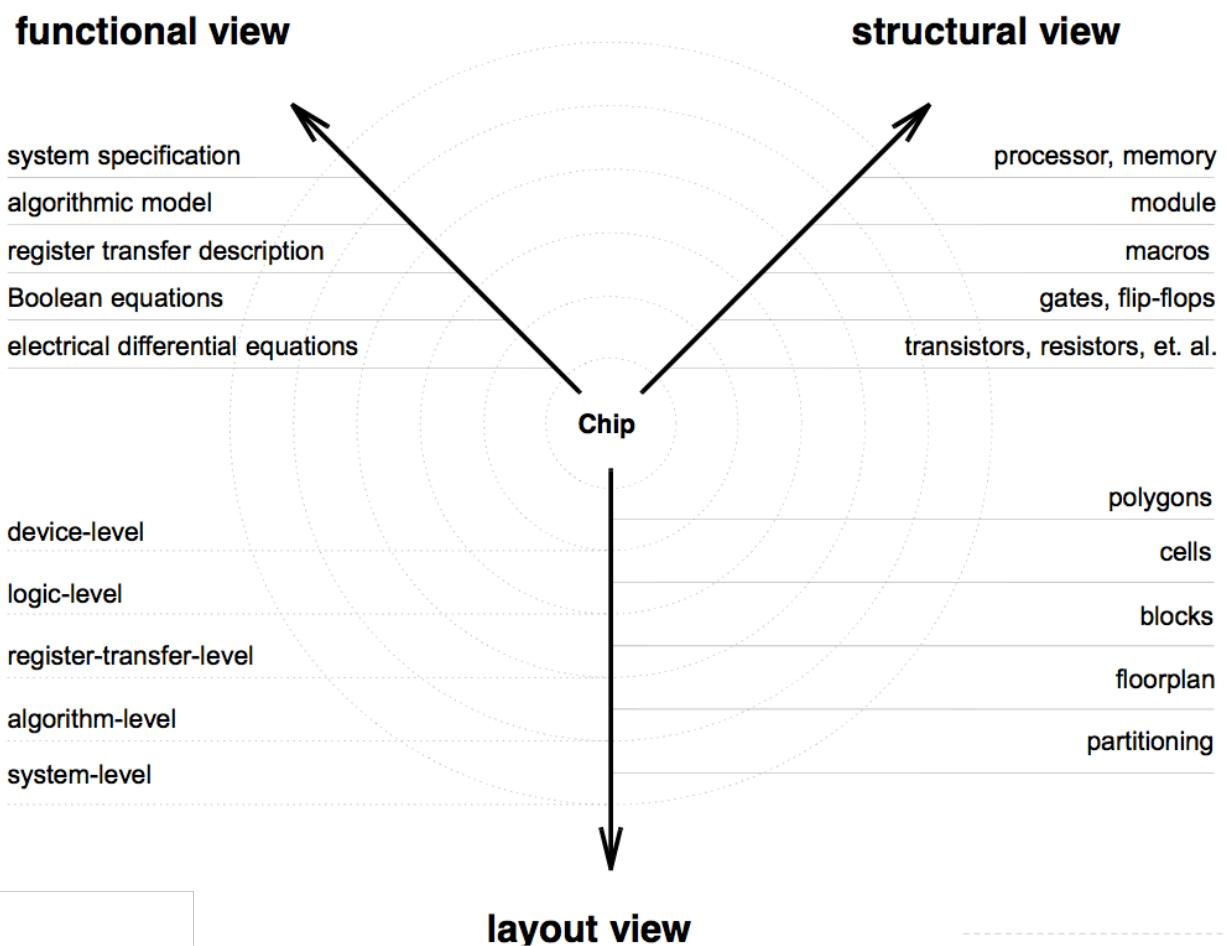
- Dozenten und Studenten

- Jobchancen
- Notwendigkeit, auf dem Laufenden zu bleiben

## 1.28. Ansichten zur Hardware (I)

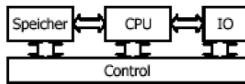
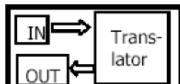
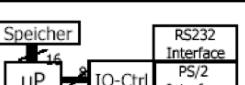
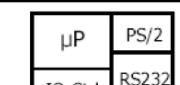
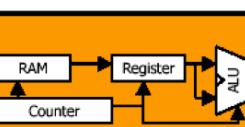
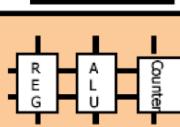
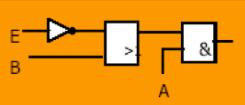
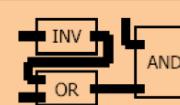
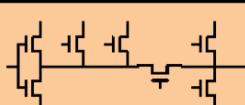


## 1.29. Ansichten zur Hardware (II)

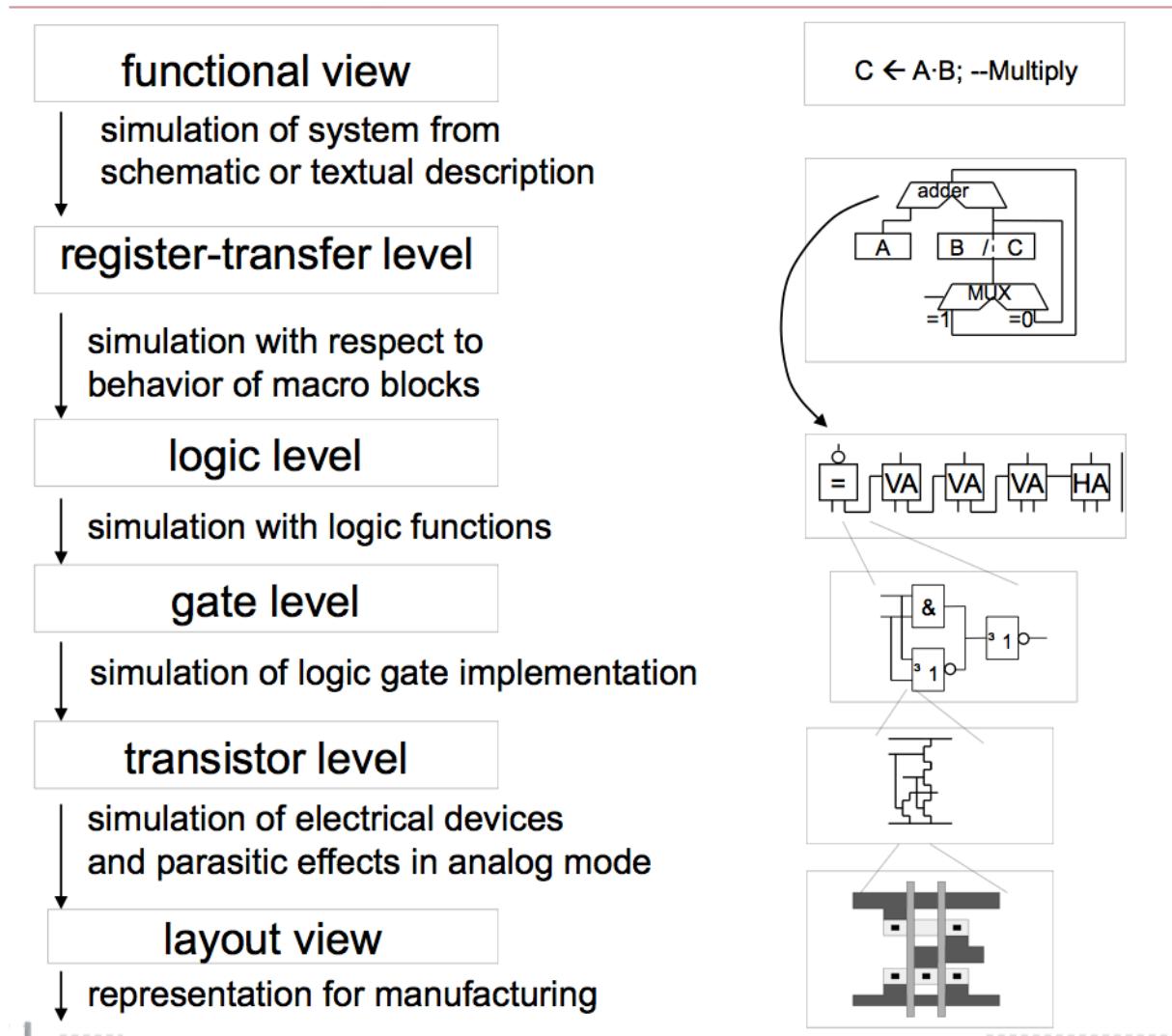


## 1. Einleitung und Übersicht

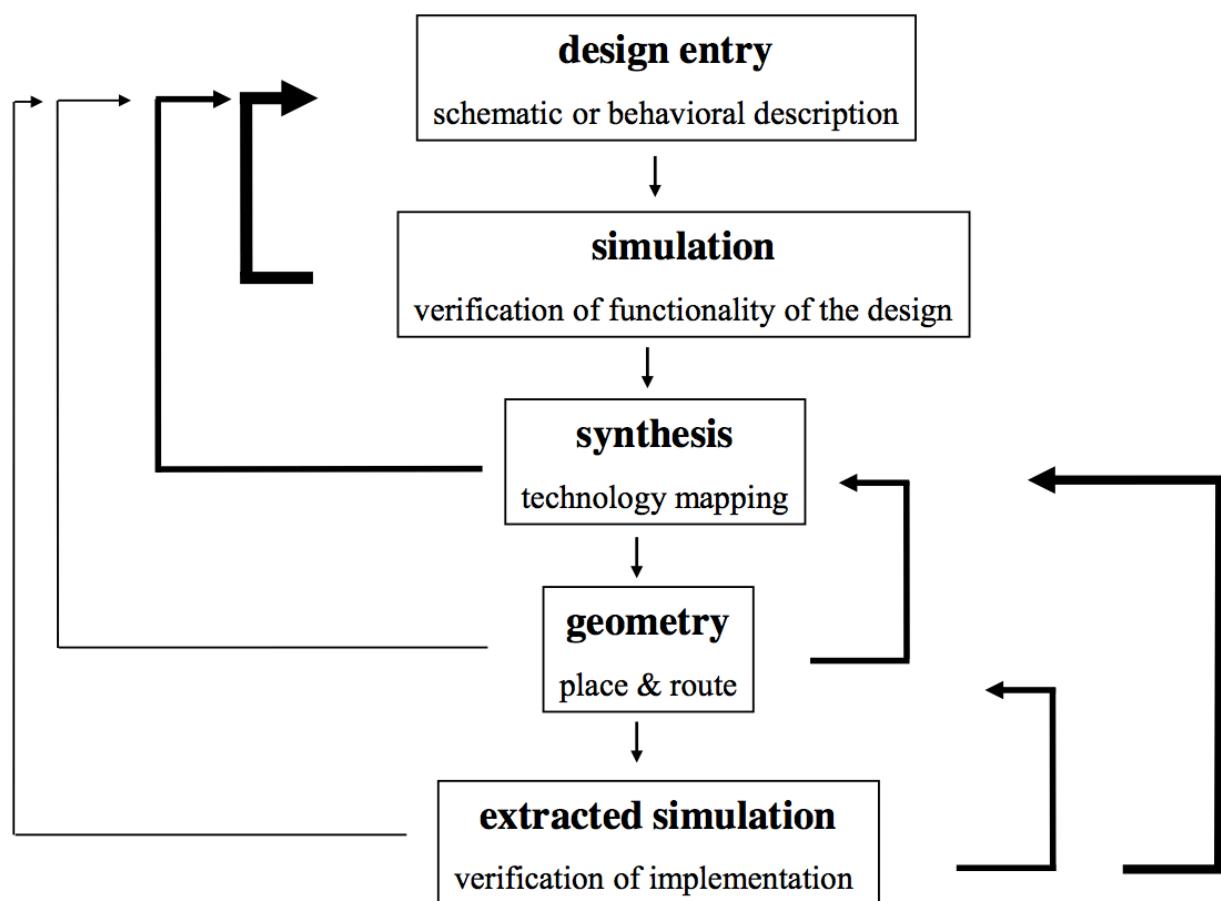
### 1.30. Abstraktionsebenen

	function	structure	layout
system level	Inputs : Keyboard Output: Display Funktion: .....		
algorithmic level	while input Read „Schilling“ Calculate Euro Display „Euro“		
register transfer level	if A='1' then B:= B+1 else B:= B end if		
logic level	D = NOT E C = (D OR B) AND A		
device level	$\frac{dU}{dt} = R \frac{dI}{dt} + \frac{I}{C} + L \frac{d^2I}{dt^2}$		

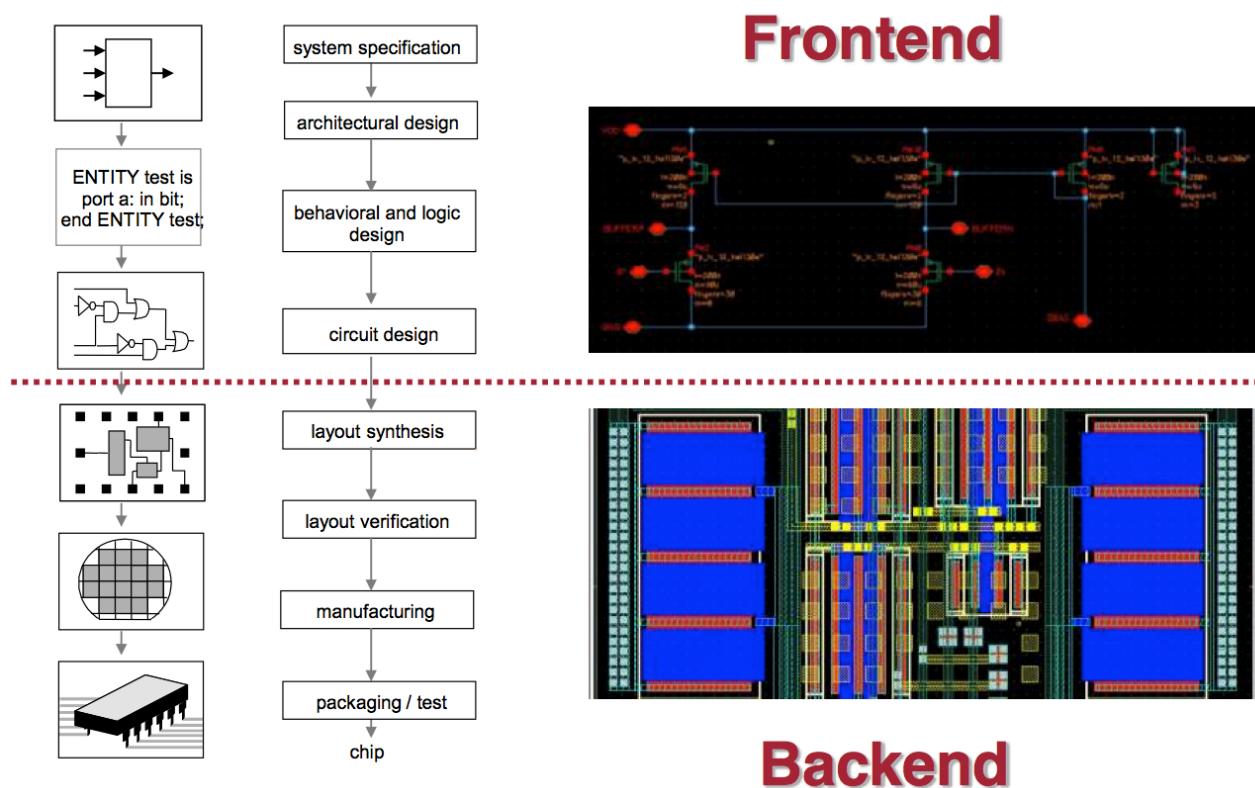
## 1.31. Entwurfsablauf



### 1.32. Verifikation

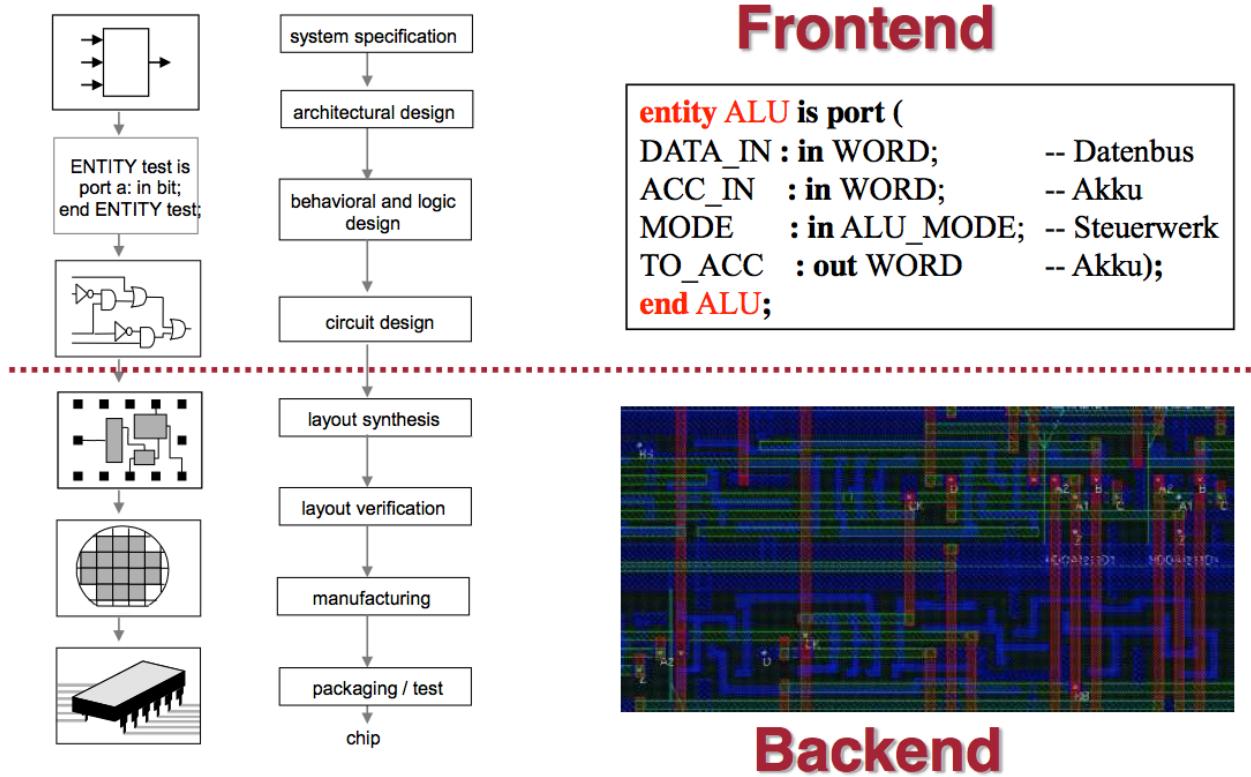


### 1.32.1. Frontend vs. Backend (analog)



## 1. Einleitung und Übersicht

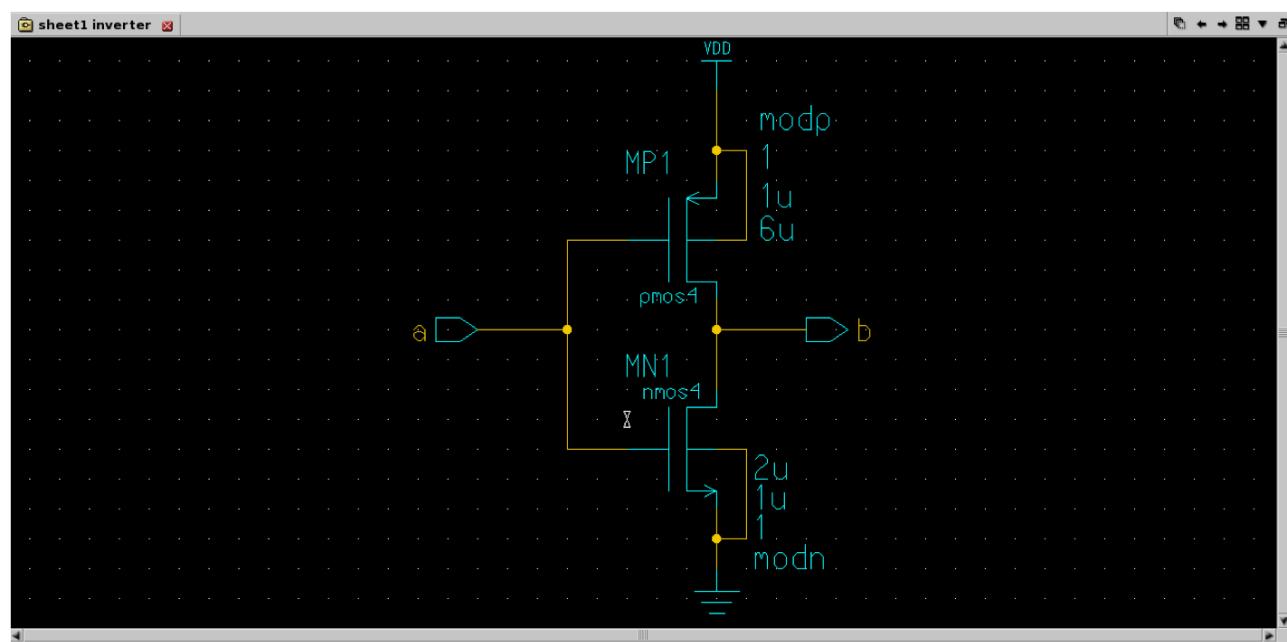
### 1.33. Frontend vs. Backend (digital)



### 1.34. Design-/Entwurfsmethodik

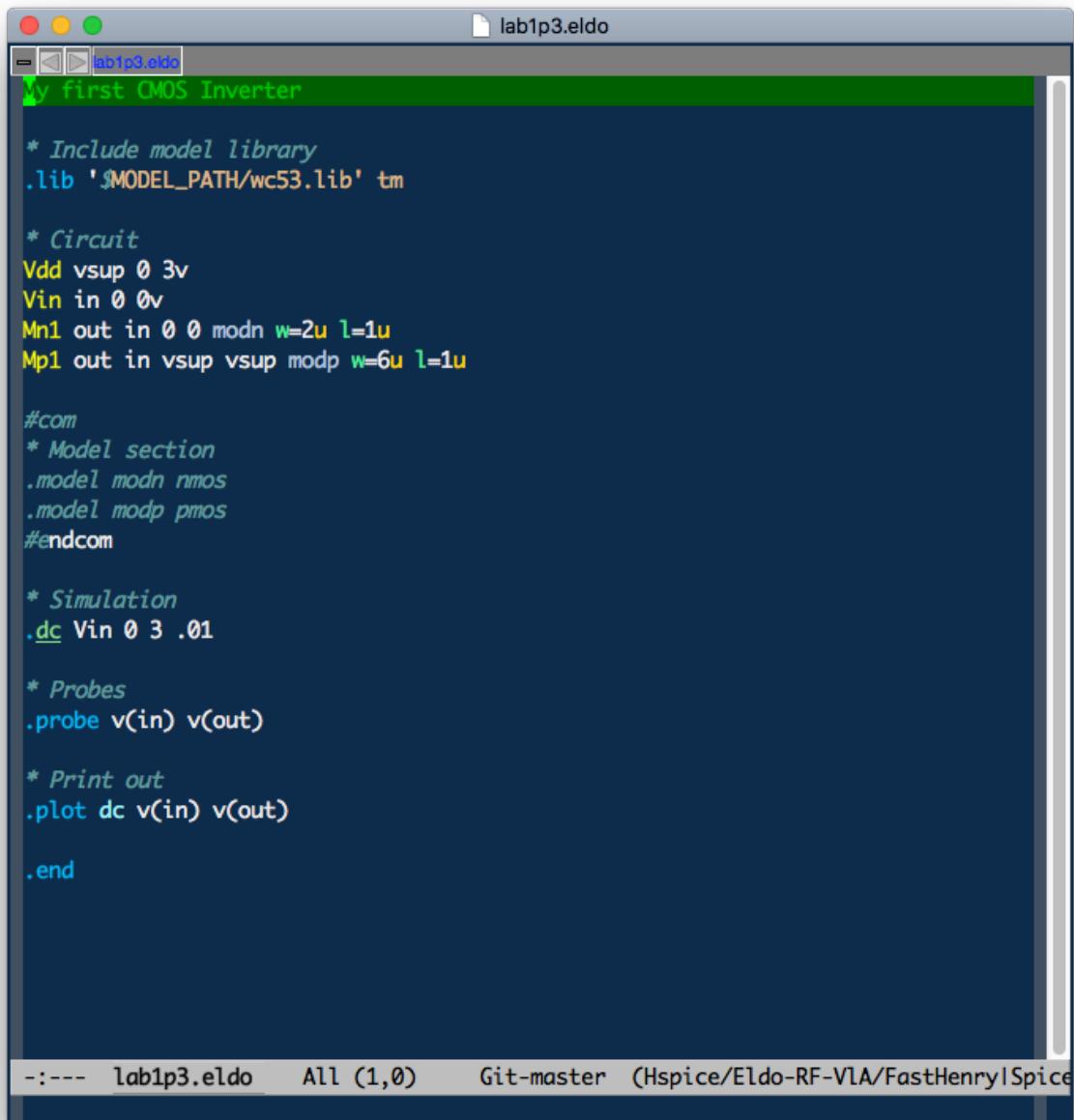
- Full Custom - vollständig manuell: ASIC
  - Überwiegend analoge Schaltungen
  - Einfache digitale Gatter
  - Volle Kontrolle, aber lange Entwicklungszeit (bis zu Jahren)
- Semi-custom: ASIC-Fertigung mit Verwendung von vorgefertigten Teilen
  - Standardzellen, Makrozellen, IP's
  - Wiederverwendung von vordefinierten Blöcken oder Maskensätzen
  - Eingeschränkte Kontrolle/Flexibilität, aber kürzere Entwicklungszeit (bis zu Wochen)
- Vollständig automatisiert: Keine Fertigung, reprogrammierbare ASICs
  - FPGA, PLA
  - Ausschließlich digitale Schaltungen
  - Schnelles Prototyping

## 1.35. Analog Design Entry



1. Einleitung und Übersicht

## 1.36. Netlist



The screenshot shows a terminal window titled "lab1p3.eldo" containing a SPICE netlist for a CMOS inverter. The netlist includes library inclusion, component definitions, model sections, simulation parameters, probe definitions, plot commands, and an end statement.

```
* Include model library
.lib '$MODEL_PATH/wc53.lib' tm

* Circuit
Vdd vsup 0 3v
Vin in 0 0v
Mn1 out in 0 0 modn w=2u l=1u
Mp1 out in vsup vsup modp w=6u l=1u

#com
* Model section
.model modn nmos
.model modp pmos
#endcom

* Simulation
.dc Vin 0 3 .01

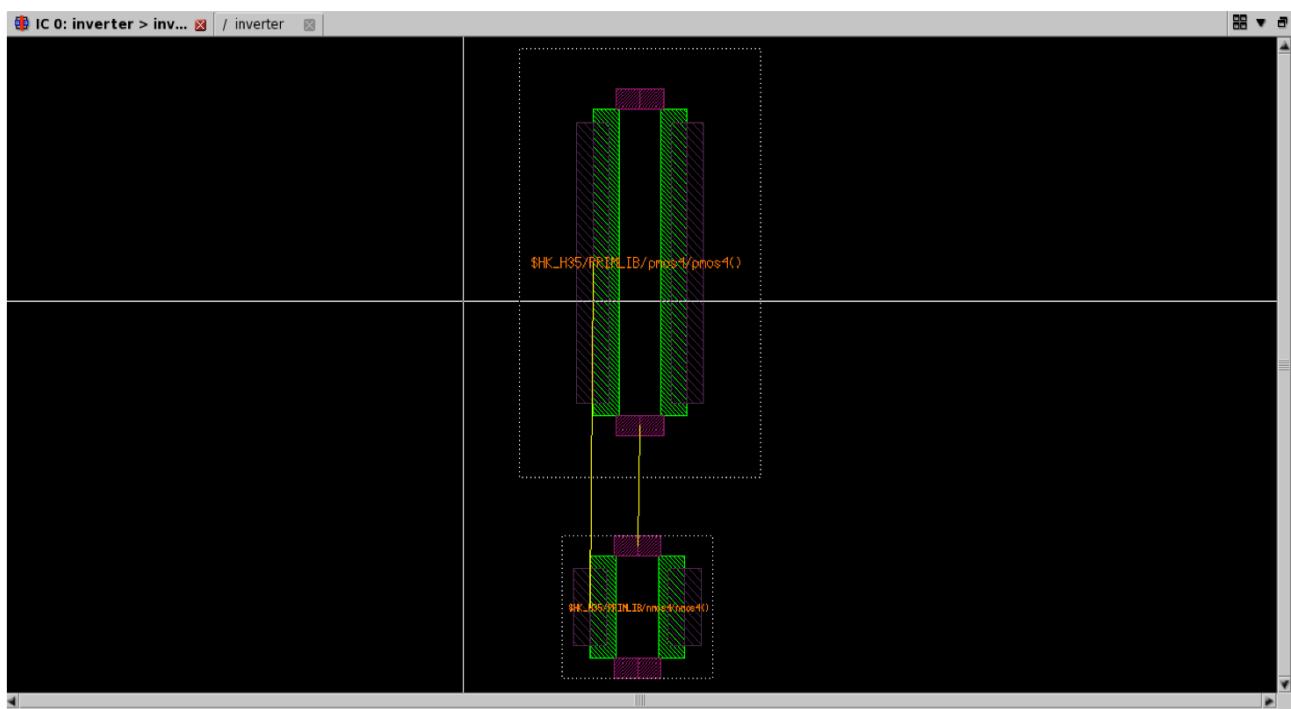
* Probes
.probe v(in) v(out)

* Print out
.plot dc v(in) v(out)

.end

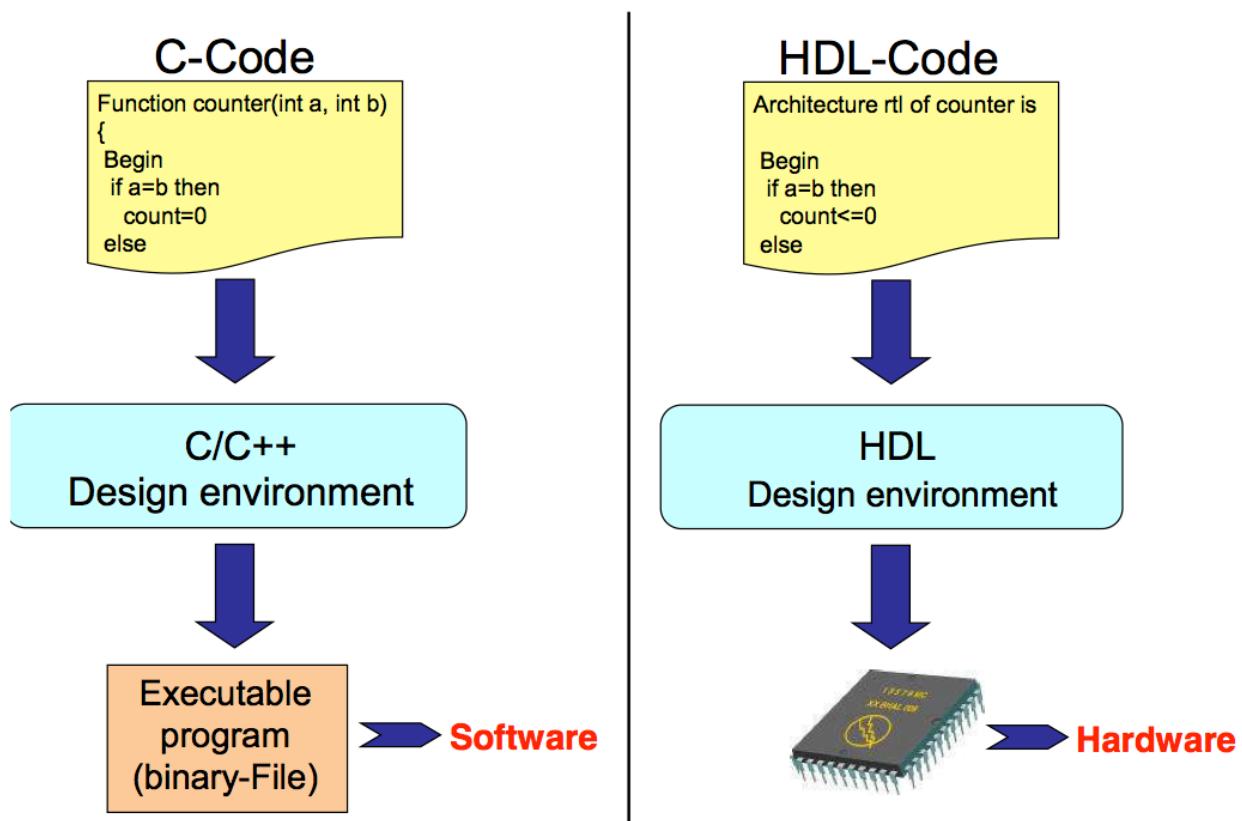
-:--- lab1p3.eldo All (1,0) Git-master (Hspice/Eldo-RF-VLA/FastHenry|Spice
```

## 1.37. Layout

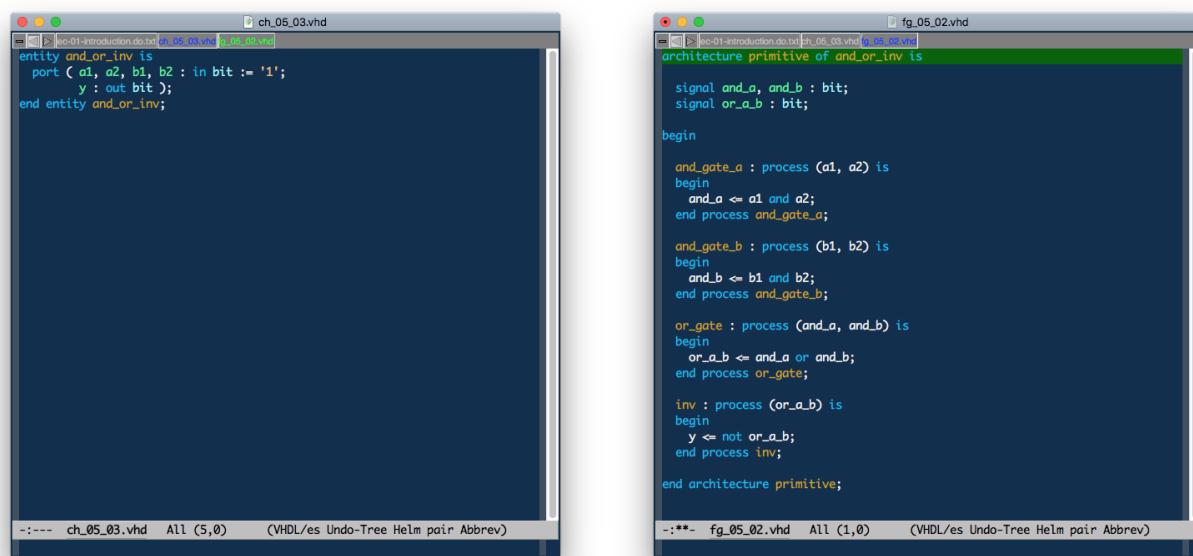


1. Einleitung und Übersicht

### 1.38. Digital Design Entry



## 1.39. Hardwarebeschreibungssprachen



The image shows two side-by-side screenshots of a VHDL code editor interface. Both windows have a dark blue background and white text.

**Left Window (ch\_05\_03.vhd):**

```

entity and_or_inv is
  port ( a1, a2, b1, b2 : in bit := '1';
        y : out bit );
end entity and_or_inv;

```

**Right Window (fg\_05\_02.vhd):**

```

architecture primitive of and_or_inv is
begin
  and_gate_a : process (a1, a2) is
  begin
    and_a <= a1 and a2;
  end process and_gate_a;

  and_gate_b : process (b1, b2) is
  begin
    and_b <= b1 and b2;
  end process and_gate_b;

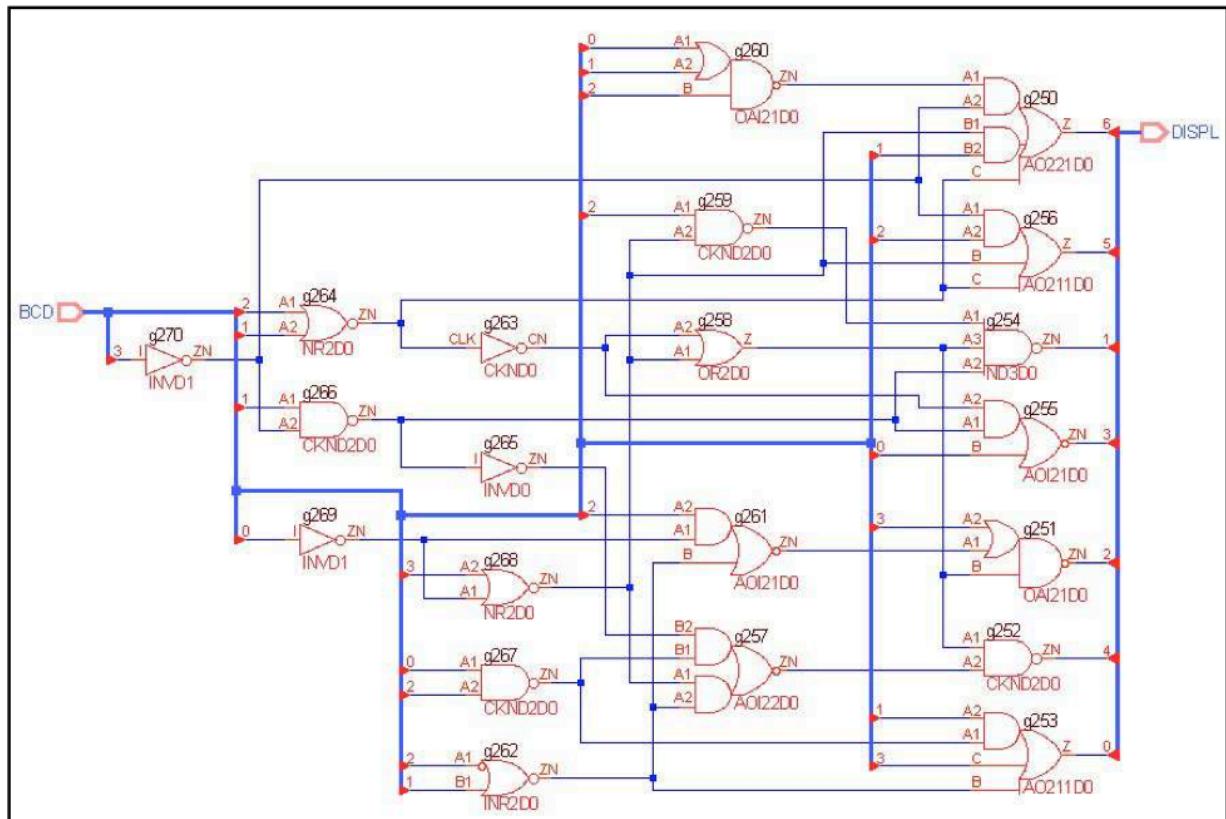
  or_gate : process (and_a, and_b) is
  begin
    or_ab <= and_a or and_b;
  end process or_gate;

  inv : process (or_ab) is
  begin
    y <= not or_ab;
  end process inv;
end architecture primitive;

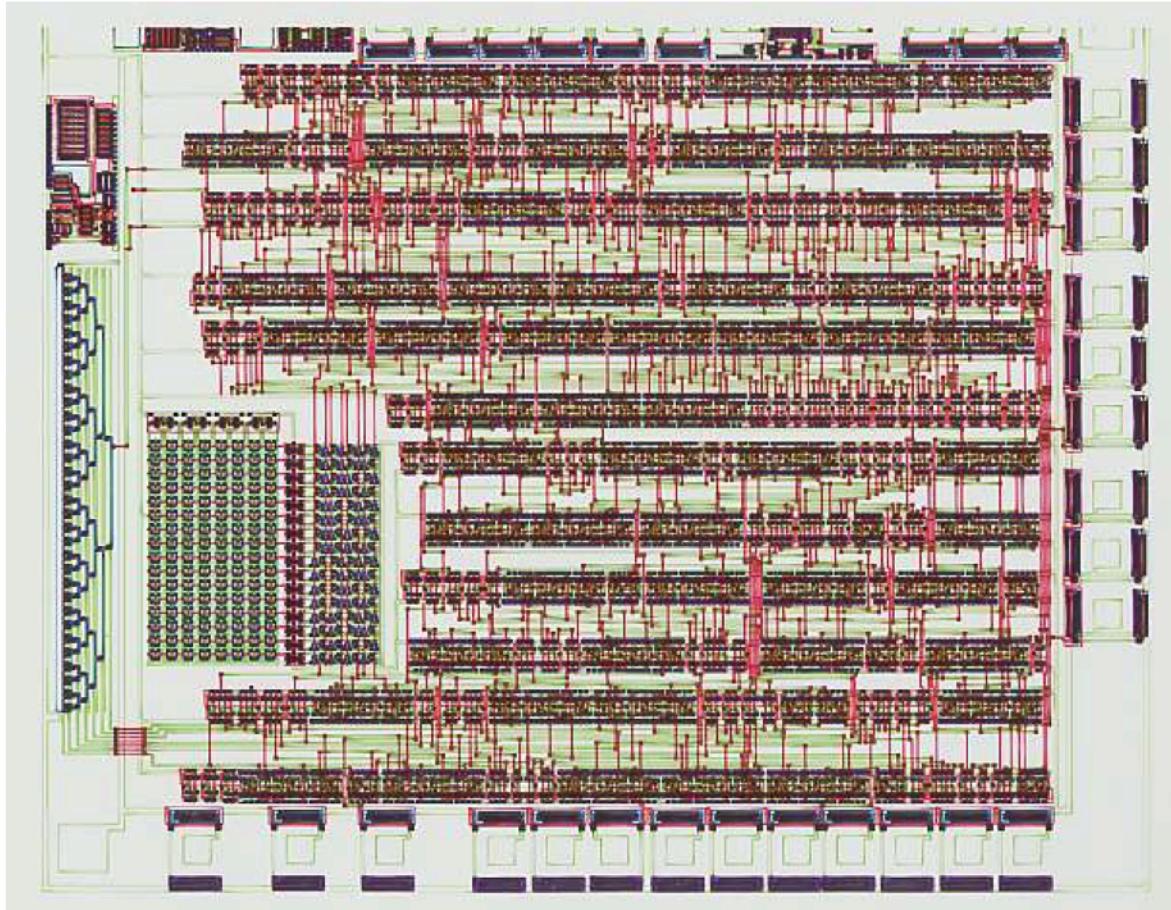
```

1. Einleitung und Übersicht

## 1.40. Technology-Gates und Netlisting



## 1.41. Standard Cell Layout



## 1.42. Nachhaltige Elektronik ...

<https://www.youtube.com/embed/7S5IuaKiZIY>

Geekchester.

## 1.43. Warum es sicht lohnt ...

<https://www.youtube.com/embed/SwPGxwBZw6I>

Circuit Song.

1. Einleitung und Übersicht

## 1.44. Und ab an den Strand ...

<https://www.youtube.com/embed/ekkJlQf-K4I>

Viva la Electronica.