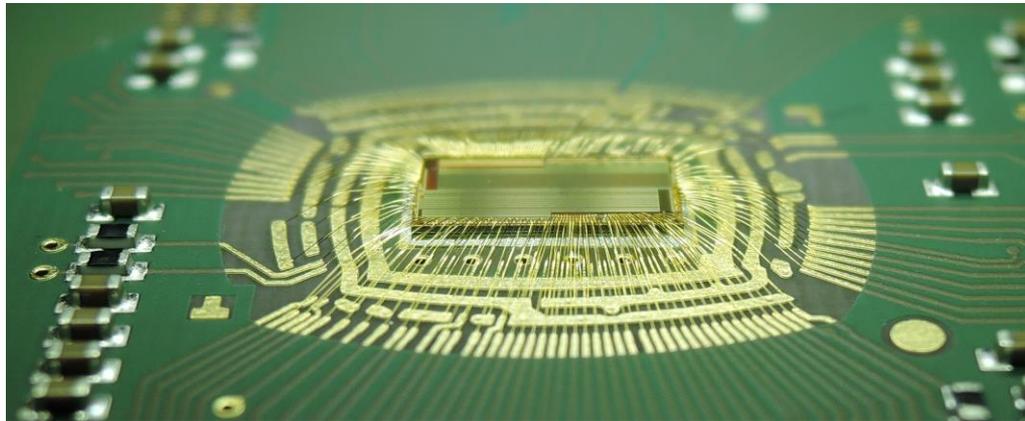


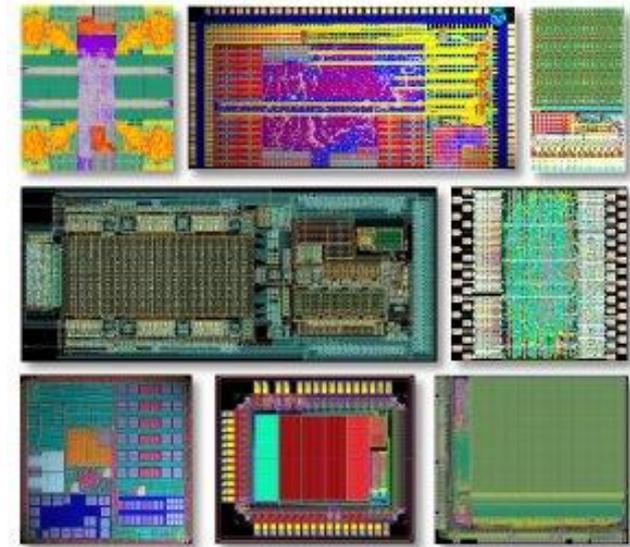


Schaltkreis- und Systementwurf

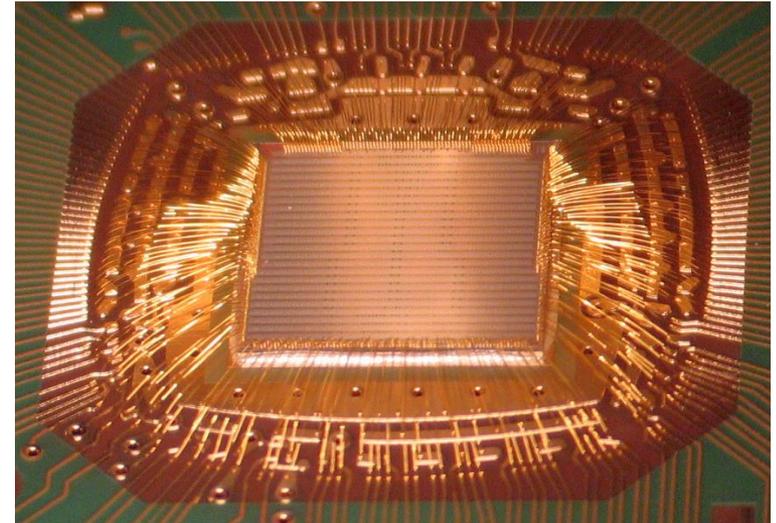


- Dr.-Ing. Sebastian Höppner
- Raum TOE216
- sebastian.hoepfner@tu-dresden.de
- <https://tu-dresden.de/ing/elektrotechnik/iee/hpsn>

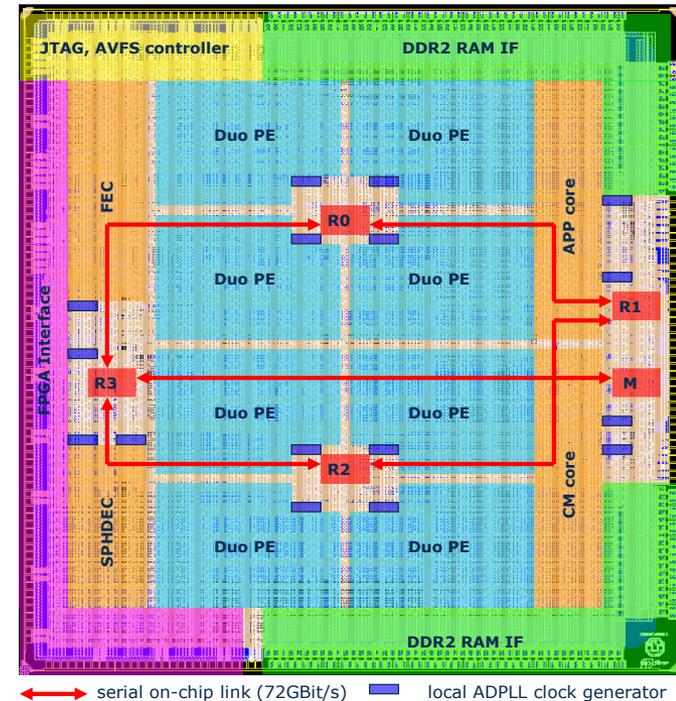
- Vorlesungsfolien unter:
 - <https://tu-dresden.de/ing/elektrotechnik/iee/hpsn/studium/materialien>
 - <https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/26417856515>



- Entwurf von integrierten Schaltkreisen („Chips“) als Kernbestandteil moderner Elektronikprodukte
- Realisierung komplexer Schaltungen auf einem Chip ermöglicht:
 - Hohe Funktionalität
 - Hohe Integrationsdichte
 - Geringe Verlustleistung
 - Hohe Zuverlässigkeit
- Die Verkleinerung der Fertigungstechnologien ermöglicht die Integration immer komplexerer Systeme
- Herausforderung bei Entwurf und Verifikation



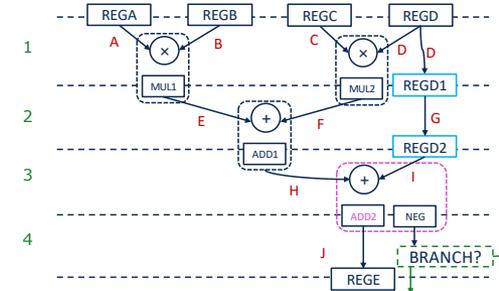
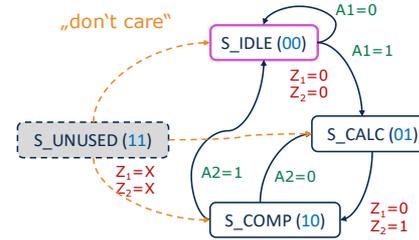
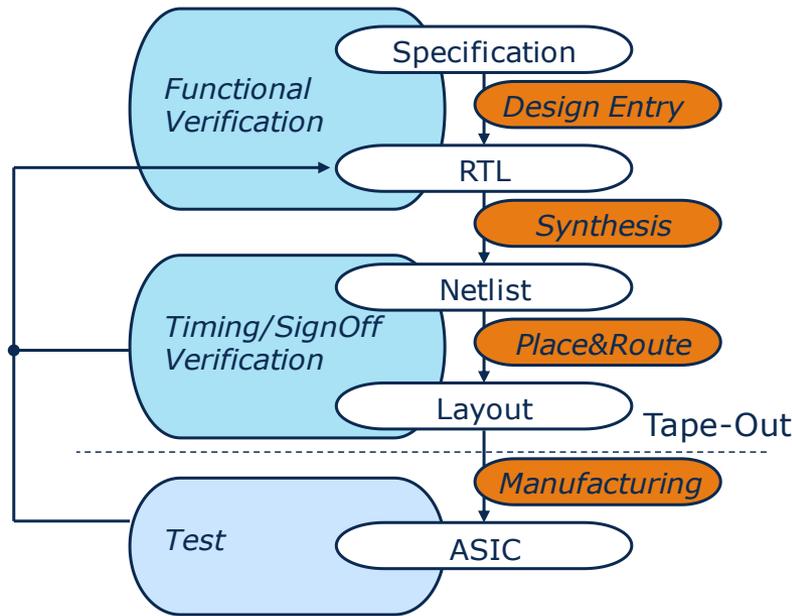
- **„Tomahawk2“** Software-Defined-Radio Basisbandprozessor
 - 36 mm² Chipfläche in 65nm CMOS Technologie
 - 465 Pads
 - 10,2 Millionen Gatter
 - 750kByte SRAM
- 20 Prozessorkerne
- Adaptives und dynamisches Power-Management
- 17 Taktgeneratoren
- Network-on-Chip mit seriellen on-chip links mit bis zu 72Gbit/s Datenrate
- DDR2 Speicherinterface
- FPGA Interface mit 10Gbit/s



Entwickelt durch die Professuren HPSN, und Vodafone Chair (Prof. Fettweis) im CoolBaseStations Projekt

- Entwurfsablauf integrierter digitaler Schaltungen
- Realisierung von Algorithmen in Hardware
- Hardwarebeschreibungssprache Verilog
- Schaltungssimulation und Verifikation
- Themen zu digitalen Schaltungen in CMOS Technologie
 - Timing,
 - Verlustleistung und Power-Management

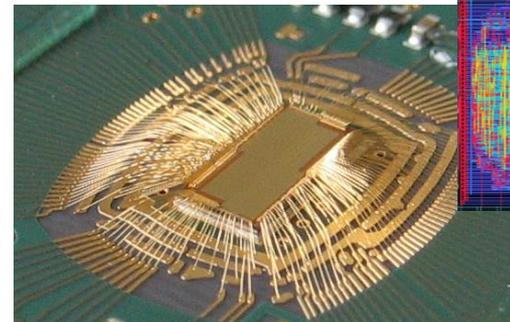
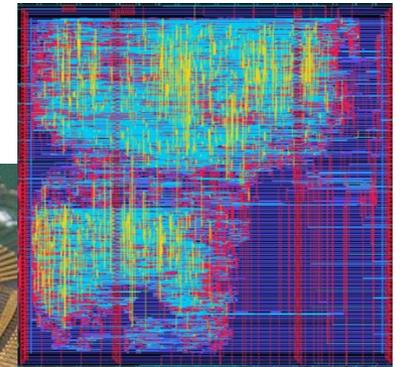
Abstraktion



```
//Addierer
module adder (sum_o, c_o, c_i, a_i, b_i);
parameter C_DWIDTH=4;

input [C_DWIDTH-1:0] a_i, b_i;
input c_i;
output [C_DWIDTH-1:0] sum_o;
output c_o;

assign {c_o, sum_o} = a_i + b_i + c_i;
endmodule
```



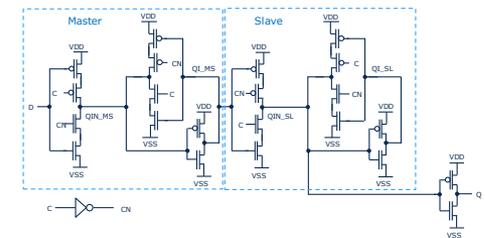
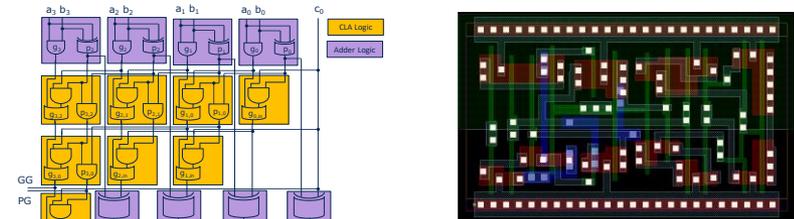
- Wintersemester: 2/1/0, Sommersemester: 0/0/2
- Wintersemester:
 - Vorlesung: **Virtuell in OPAL**
 - Übung: **Mi. 4.DS (13:00), Ort: MATRIX TUD (virtuell)**
 - Konsultation zur Belegarbeit
 - Praktikum: **Fr. 4.DS (13:00) (zum Semesterstart), Ort: MATRIX TUD (Virtuell)**
 - **Virtuelles** Einführungspraktikum und betreutes Praktikum zur individuellen Arbeit am Beleg
- Sommersemester:
 - Praktikum: **... wird rechtzeitig bekannt gegeben**
 - Betreutes Praktikum zur individuellen Arbeit am Beleg

- Realisierung eines selbst gewählten Algorithmus in Hardware
 - Referenzimplementierung in Software
 - Architekturentwicklung (Scheduling, Datenpfad, Ablaufsteuerung)
 - Implementierung der Schaltung mittels Datenpfadbaublöcken
 - Verifikation durch Schaltungssimulation
 - Synthese und Place&Route der Schaltung
- Schriftlicher Beleg ([Details in der Übung!](#))
- Elektronische Abgabe (.pdf)
- Abgabe:
 - Prüfungseinschreibung in dem Semester in dem Abgabe erfolgt!
 - **Abgabe bis 15.09. (SS) bzw. 15.03. (WS)**

- Diese Lehrveranstaltung wird
 - den Design Flow für digitale integrierte Schaltungen vorstellen
 - Methoden zur Implementierung von Algorithmen in Hardware vorstellen
 - Grundlagen und Konzepte der Verilog HDL vorstellen
 - Verifikationsstrategien vermitteln
 - Vertiefende Grundlagen zur Realisierung digitaler Schaltungen in CMOS Technologien vermitteln
- Praktische Erfahrung in der Implementierung einer digitalen Schaltung vermitteln
- Anregung zum Selbststudium geben

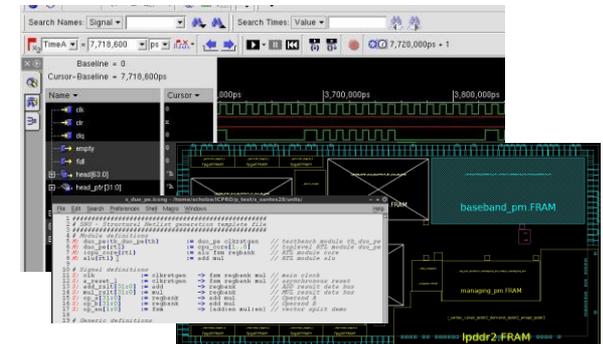
- **Digitale Schaltungstechnik**

- Grundlagen zu digitalen Schaltungen
- Arithmetik, Zahlenformate
- Grundlagen digitale CMOS Schaltungen
- Speicherschaltungen
- Mixed-Signal Schaltungen, Signalwandler

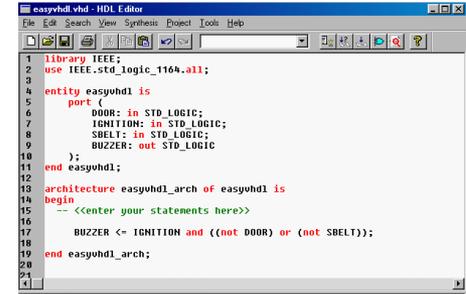


- **VLSI Prozessorentwurf**

- RTL Entwurf eines Prozessors
- Simulation und Verifikation
- Top-Down Design RTL2GDS Flow (Synthese, Timing Analyse, Place&Route, Power Analyse, Sign-off Analyse)
- Praktikum: Implementierung eines Prozessors



- **VHDL-Entwurf (Oberseminar Informationstechnik)**
 - Einführung in Entwurf, Modellierung, Verifikation mittels VHDL
 - Vorstellung von Aufgaben und Lösungsansätzen aus aktuellen Forschungsarbeiten des Lehrstuhls
 - Referate der Teilnehmer zu ihren Beleg-Projekten



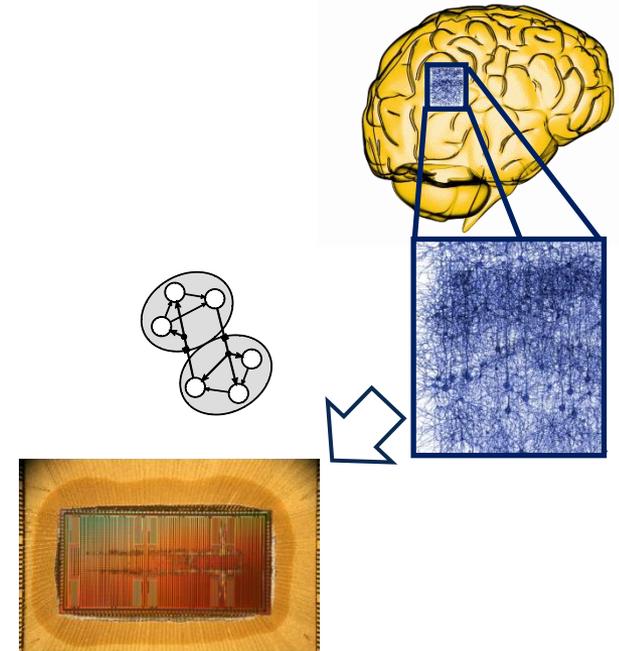
```

1 library IEEE;
2 use IEEE.std_logic_1164.all;
3
4 entity easvhd1 is
5     port (
6         D00R: in STD_LOGIC;
7         IGNITION: in STD_LOGIC;
8         SBELT: in STD_LOGIC;
9         BUZZER: out STD_LOGIC
10    );
11 end easvhd1;
12
13 architecture easvhd1_arch of easvhd1 is
14     begin
15         -- <<enter your statements here>>
16
17         BUZZER <- IGNITION and ((not D00R) or (not SBELT));
18
19     end easvhd1_arch;
20
21
22
23
24
25

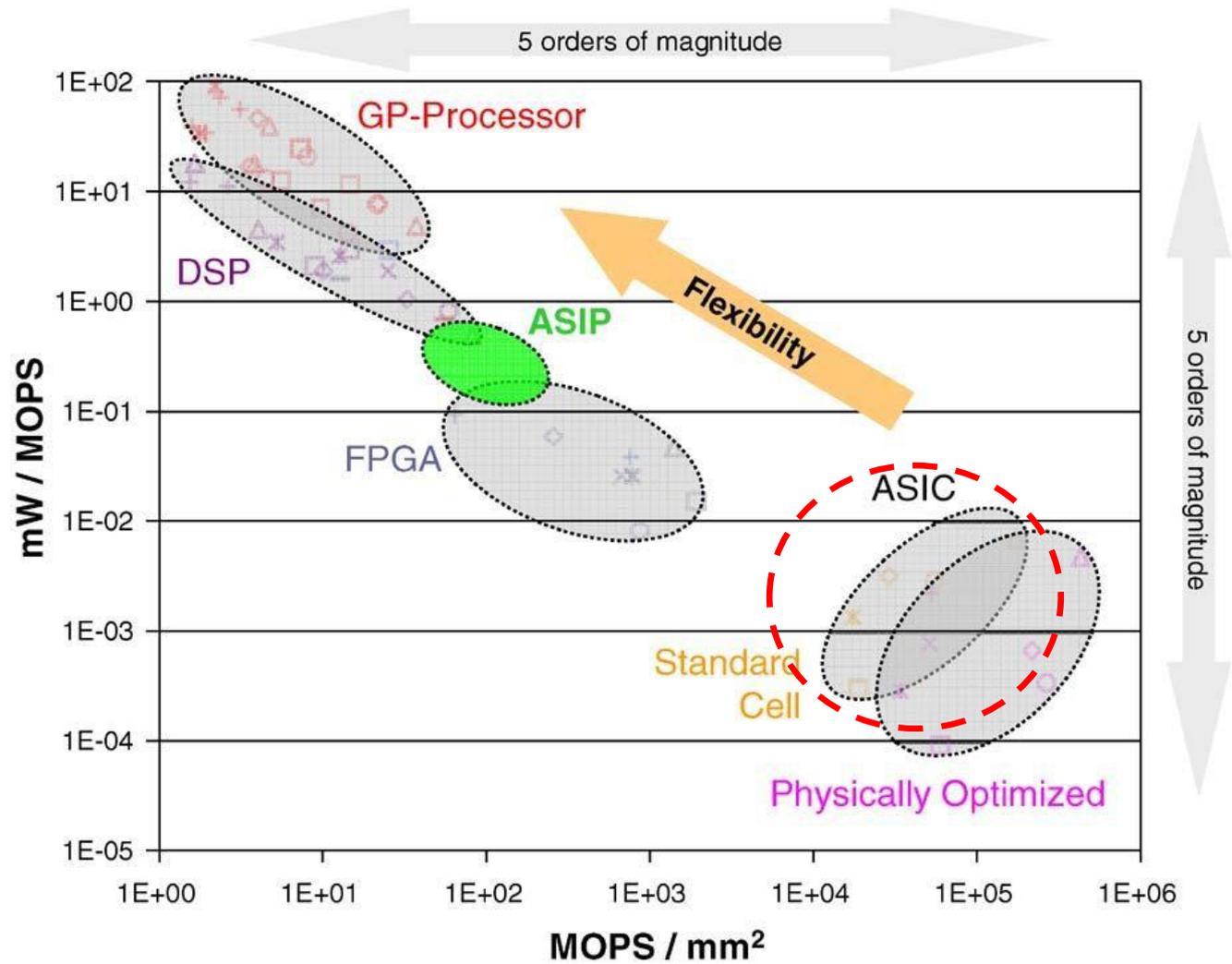
```

- **Neuromorphe VLSI-Systeme**

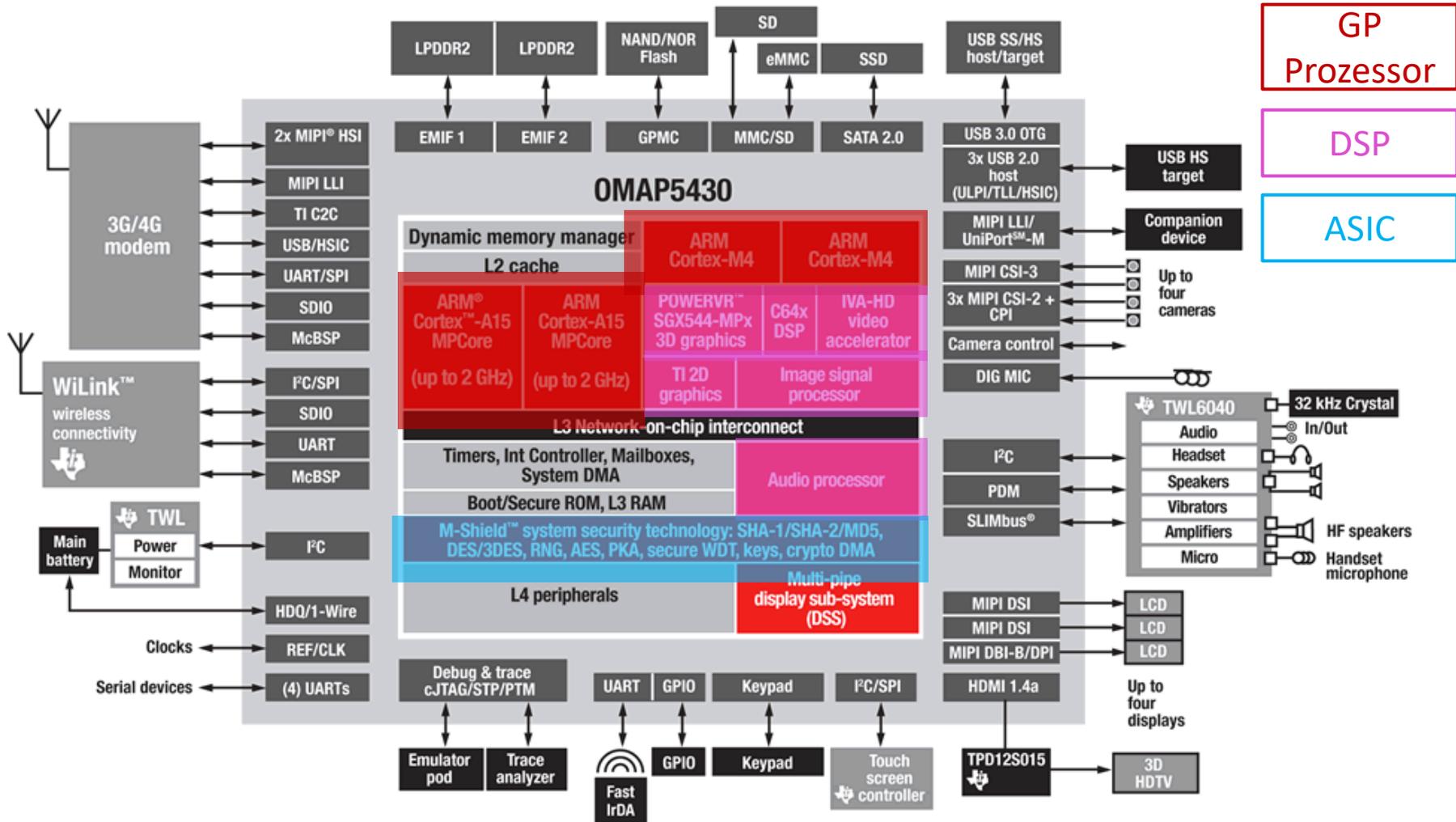
- Entwurfsmethoden für integrierte analoge CMOS-Schaltungen und deren Schaltungsdimensionierung
- Neuromorphe VLSI-Systeme und deren neurobiologische Grundlagen, Anwendungen z.B. in Brain-Machine-Interfaces
- Praktikum zur Erstellung und Analyse von analogen und neuromorphen CMOS-Schaltungen mit der Entwurfssoftware Cadence DF2



Anwendungsspezifische Integrierte Schaltungen (ASICs)



Quelle: Tobias G. Noll, Thorsten von Sydow, Bernd Neumann, Jochen Schleifer, Thomas Coenen, and Gotz Kappen "Reconfigurable Components for Application-Specific Processor Architecture" in Dynamically Reconfigurable Systems, Springer, 2010



Quelle: anandtech.com

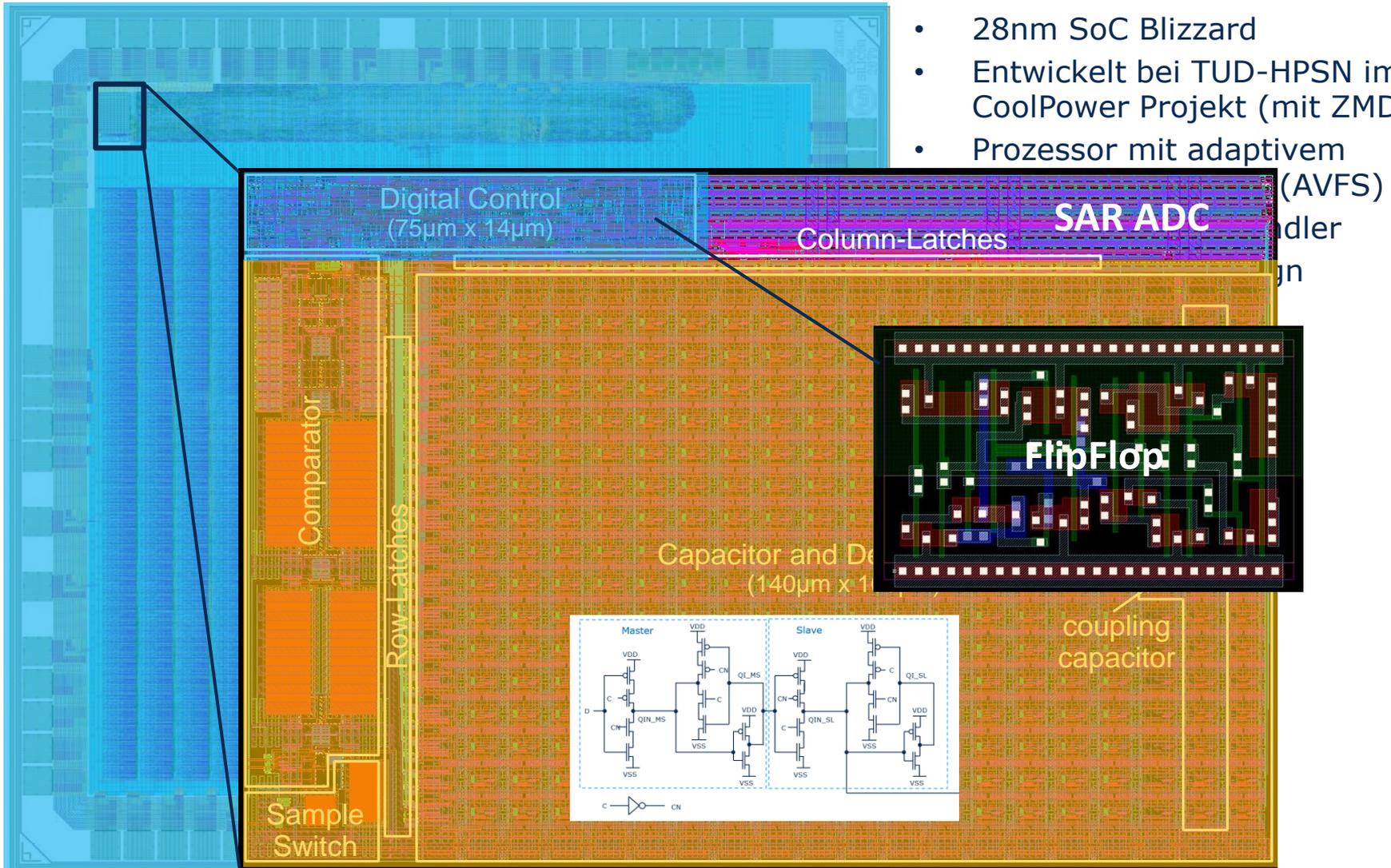
- **Full-Custom Design**

- Manuelle Eingabe von Schaltplan (schematic) und Layout
- **Vorteile:**
 - Detaillierte Optimierungen möglich
- **Nachteile:**
 - Geringe Produktivität
- Anwendungen:
 - Analoge und Mixed-Signal Schaltungen
 - Grundelemente digitaler Schaltungen (Standardzellen, I/O Zellen, Speicher)
 - Speziell optimierte digitale Schaltungen (z.B. High-speed, Ultra-Low-Power)

- **Semi-Custom Design**

- Manuelle Eingabe einer Hardwarebeschreibung (HDL)
- (Teil-) automatisierte Erzeugung der Entwurfsdaten (Netzliste, Layout)
- **Vorteile:**
 - Sehr hohe Produktivität
- **Nachteile:**
 - Eingeschränkte Optimierung von Schaltungseigenschaften
- Anwendungen:
 - Komplexe Digitale Schaltungen
 - Systems-on-Chip

- 28nm SoC Blizzard
- Entwickelt bei TUD-HPSN im CoolPower Projekt (mit ZMDI)
- Prozessor mit adaptivem

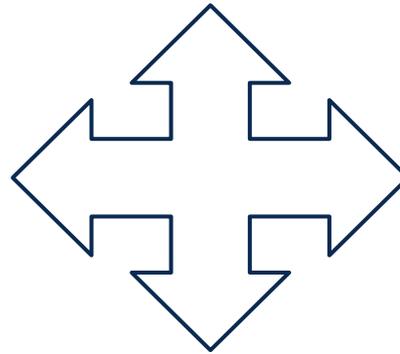


Markt

- Anforderungen (Spezifikation)
- Kunden
- Preis des Chips
- Entwicklungszeit (Time-to-Market)

Mehrwert

- Neue Funktionalität (z.B. neuer Funkstandard)
- Bessere Funktionalität (z.B. längere Akkulaufzeit)
- Kleinerer Form-Faktor (Gehäuse Größe)
- Reduzierte Anzahl der Chips im System
- Zuverlässigkeit

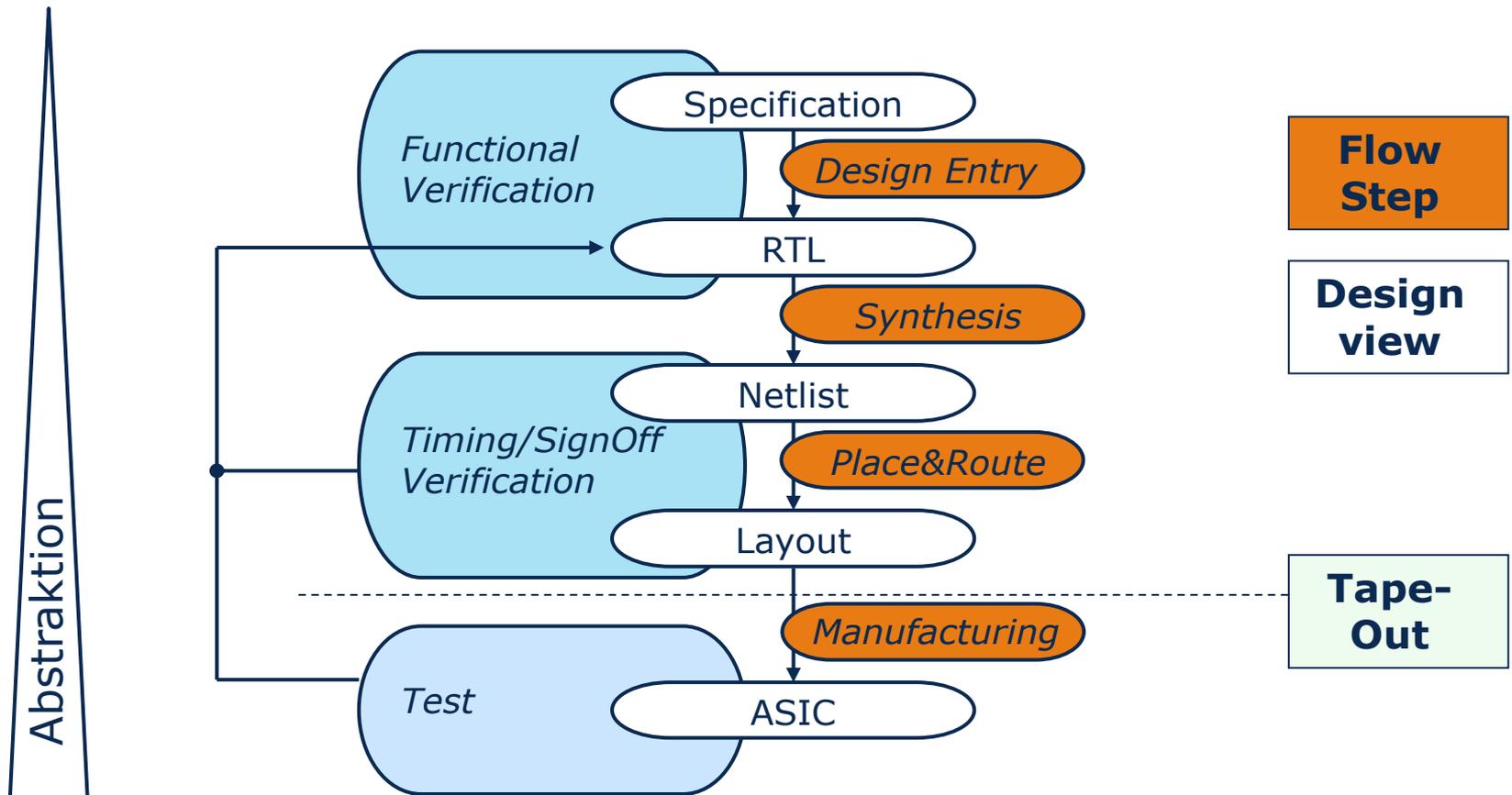


Kosten

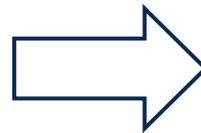
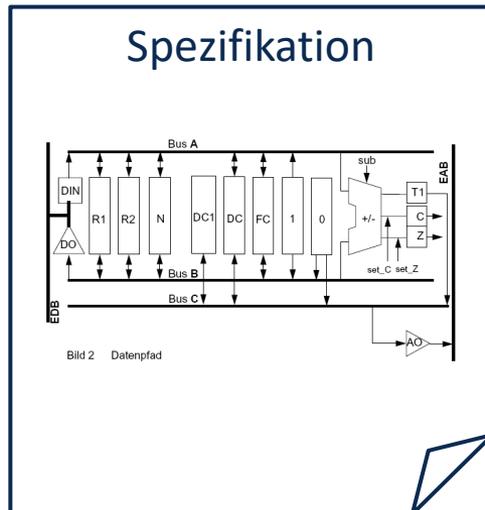
- Entwicklung (Design, Implementierung, Verifikation)
- Einkauf von IP
- Produktion (Masken, Waferherstellung, Packaging)
- Test der produzierten Chips

- **Architekturentscheidung**
- **Technologieentscheidung**
- **Package**
- **Entwurfsmethodik**

Entwurfsablauf integrierter Digitalschaltungen



- Entwurf von Schaltungsmodulen basierend auf Systemspezifikation als Register-Transfer-Level (RTL) Beschreibung
- Verwendung von Hardwarebeschreibungssprachen (HDL) wie z.B. Verilog, VHDL
- Synthesegerechte Beschreibung
- Check von RTL Coding Guidelines

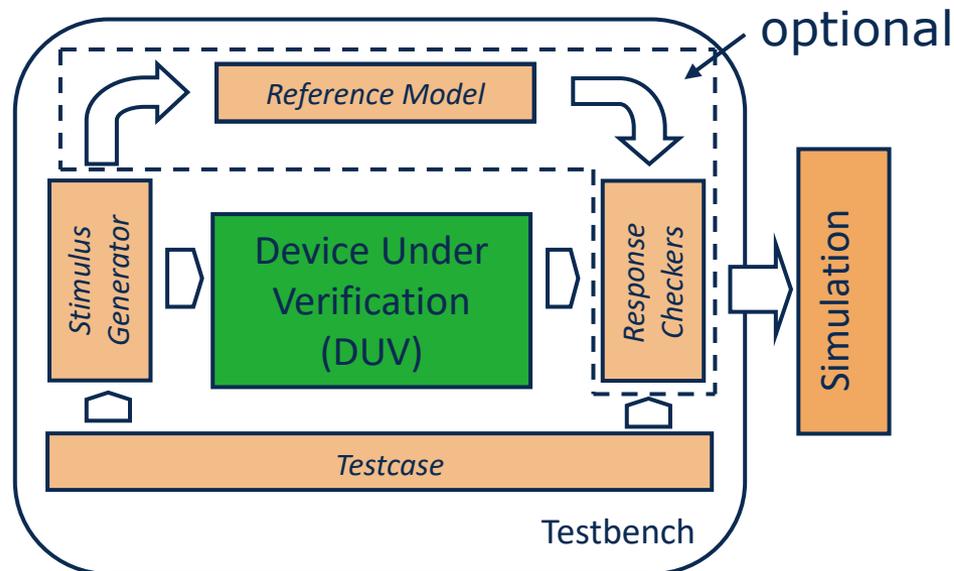


```

module fsm (
  module decoder (
    module adder_reg (clk_i, a_i, b_i, c_o);
      input clk_i;
      input [7:0] a_i, b_i;
      output [8:0] c_o;
      reg [8:0] c_r;
      always @(posedge clk_i) begin
        c_r<=a_i+b_i;
      end
      assign c_o=c_r;
    endmodule
  endmodule
endmodule

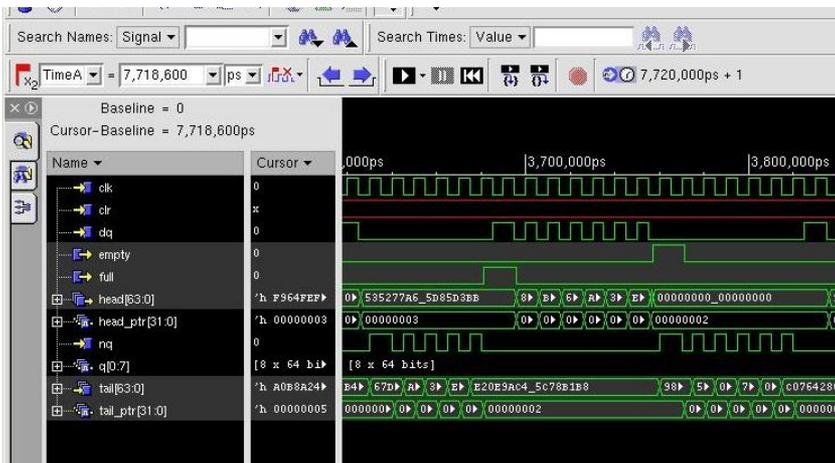
```

- Testbench: Simulationsumgebung für ein Systemmodul
- Stimulus Generator erzeugt Eingangssignale
- Response Checker prüft Ausgangssignale gemäß Spezifikation
- Optional: Vergleich mit Referenzmodell möglich
- Testbench üblicherweise mit HDL als Verhaltensbeschreibung implementiert



- Simulation der Testbench mit eventbasiertem Digitalsimulator
- Tools, u.a.
 - Cadence NCSIM
 - Mentor Questa

Interaktive Analyse/Debugging



Self-checking Testbench
Test PASS/FAIL

```

...
...
#####  ###  #####  #####  #####  #####
#  #  #  #  #  #  #  #  #  #  #
#####  #####  ###  ###  ###  #  #
#  #  #  #  #  #  #  #  #  #
#  #  #  #####  #####  #####  #####

TUD_TESTBENCH:NOTE:SIMPASS: Test PASsed
    
```

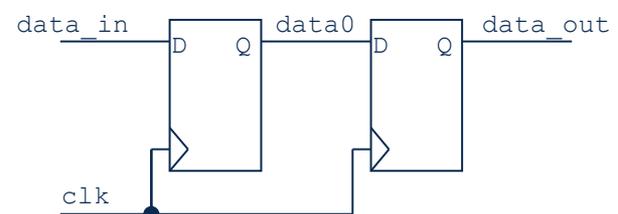
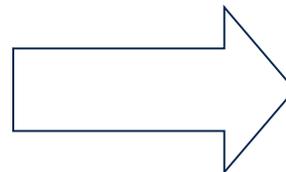
- Abbildung einer synthesesegerechten RTL Beschreibung in eine Gatternetzliste basierend auf Standardzell Bibliotheken
- Library File (.lib) beinhaltet Informationen zu Funktionalität, Timing, Power
- Vorgabe von Constraints, z.B. Taktfrequenz, Eingangs- und Ausgangsdelay
- Tools, u.a.
 - Synopsys DesignCompiler
 - Cadence RTLCompiler

```

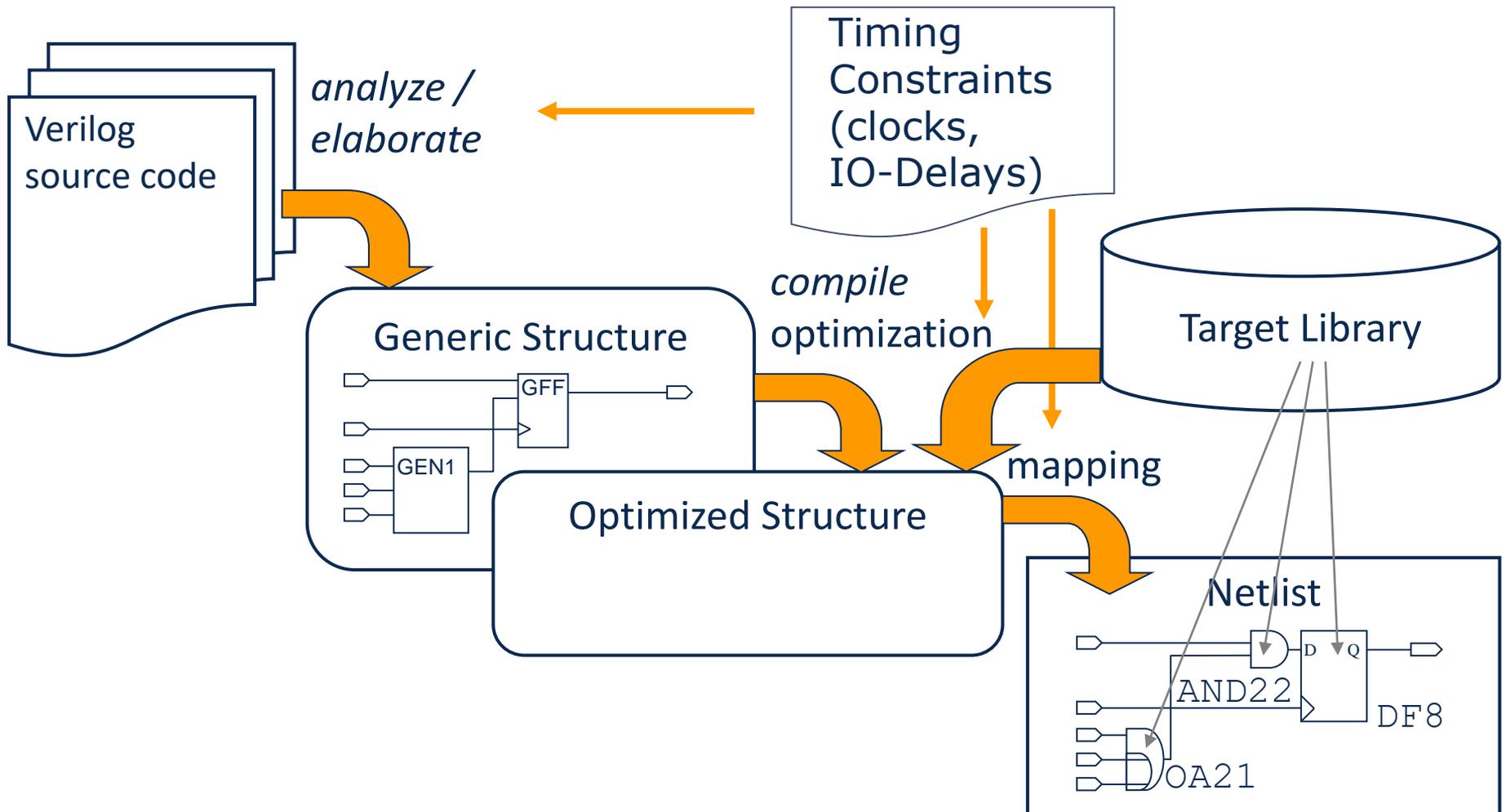
always @(posedge clk)
begin
    data0 <= data_in;
end
always @(posedge clk)
begin
    data_out <= data0;
end
  
```

Verilog RTL

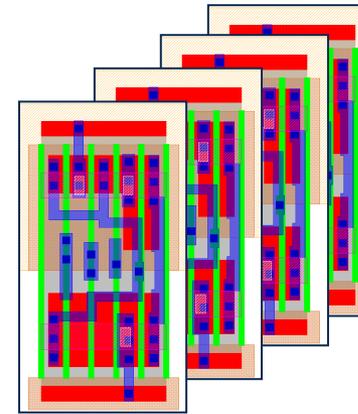
RTL Synthese



Äquivalente Gatter-Netzliste



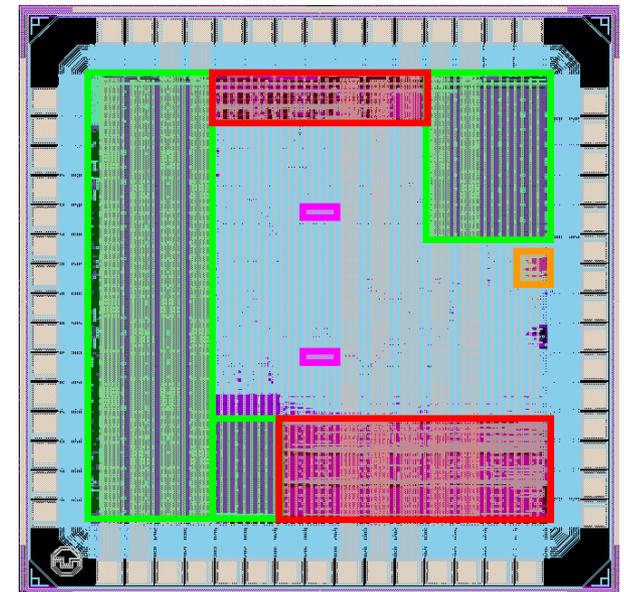
- Automatisiertes Platzieren und Verdrahten der Standardzellen
- Synthese von Clock Trees
- Extraktion von parasitären Layout Elementen



Standardzellen

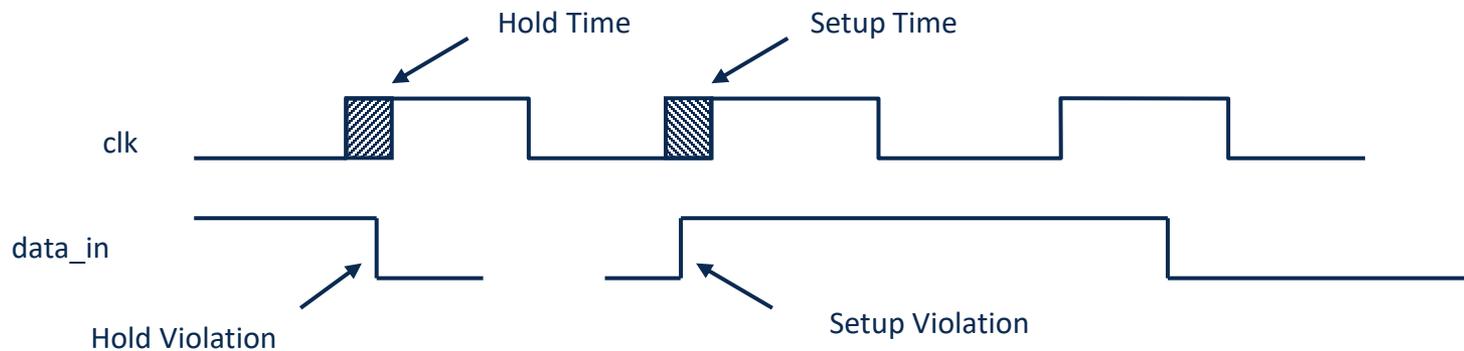
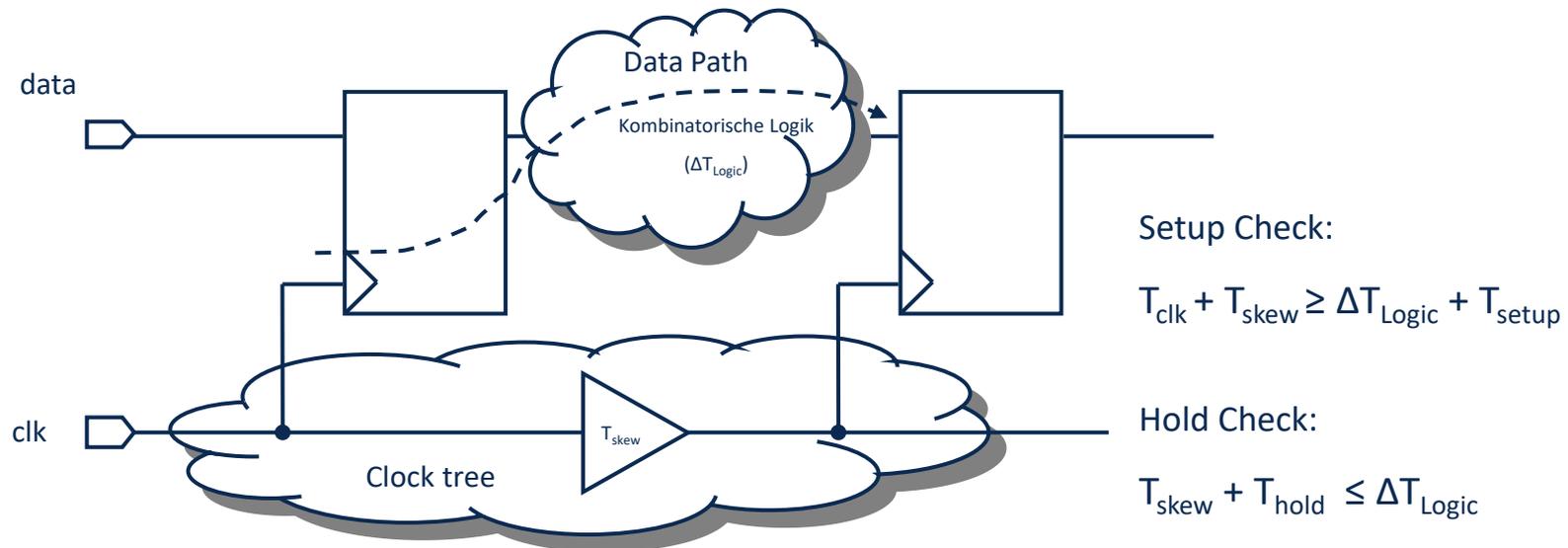


Schaltungsblock

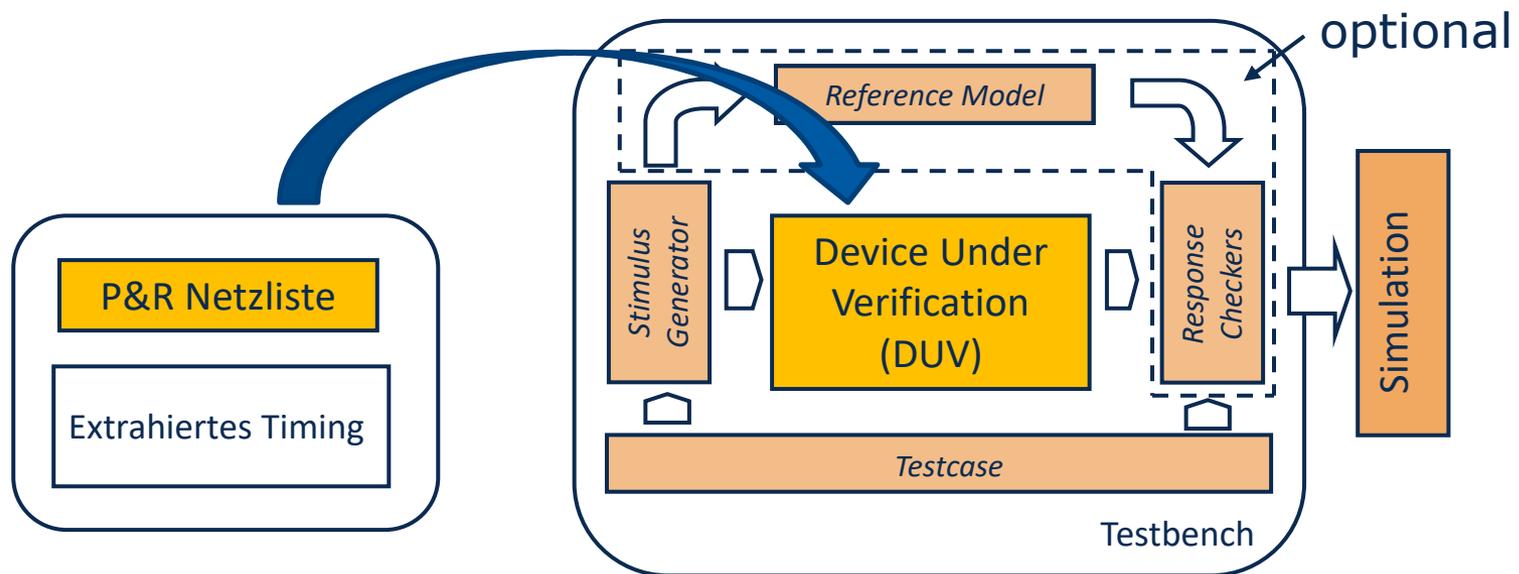


Chip

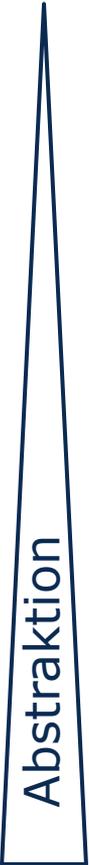
- Check von Timing Constraints



- Annotation von Verzögerungszeiten an die Gatternetzliste
- Individuelle Delays und Constraint Checks (Setup, Hold) für jedes Gatter
- zeitaufwändig

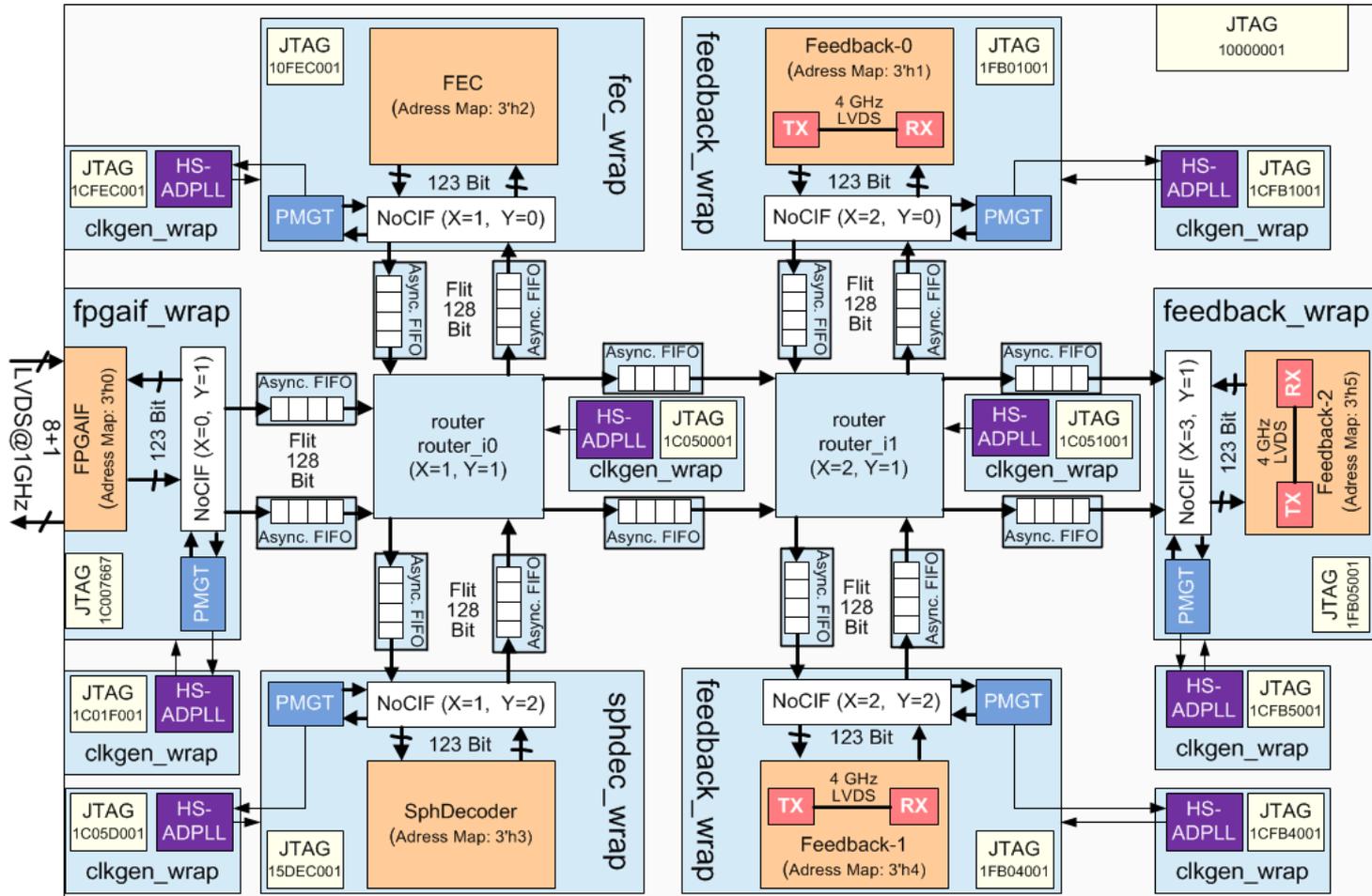


- Ein Schaltungsmodul kann verschieden repräsentiert sein:
 - Verhaltensmodel (behavioral)
 - Verhaltensmodel für Digitalsimulator
 - „Executable System Specification“
 - RTL
 - Synthesefähige RTL Beschreibung
 - Gatter Netzliste
 - Gatternetzliste als Ergebnis von Synthese/P&R
 - Transistor-Netzliste (schematic netlist)
 - Netzliste auf Transistorlevel
 - Transistor-Netzliste (extracted netlist)
 - Netzliste mit Transistoren und parasitären Layout Elementen
 - Layout
 - Physische Darstellung der zu fertigenden Schaltungsgeometrien



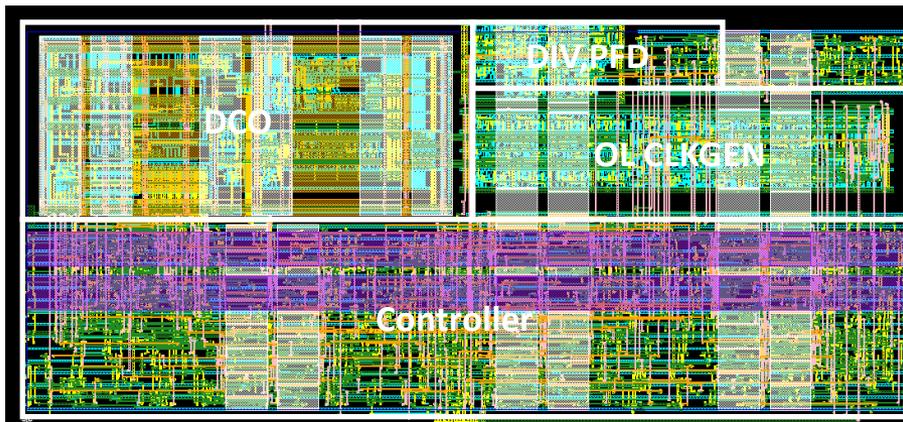
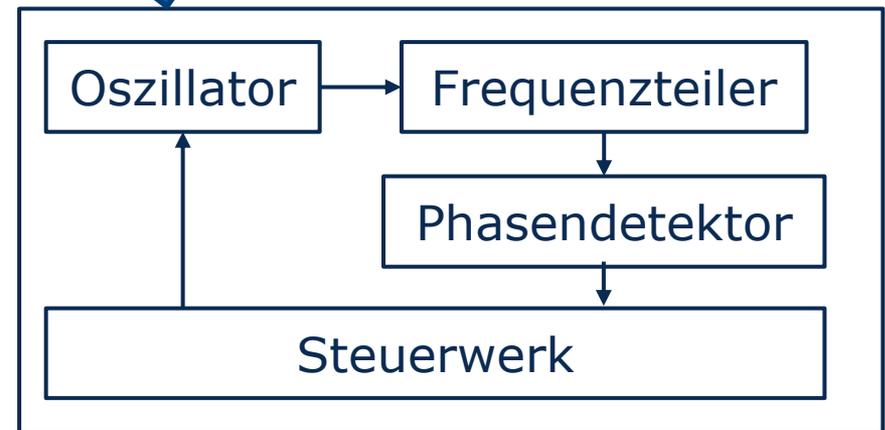
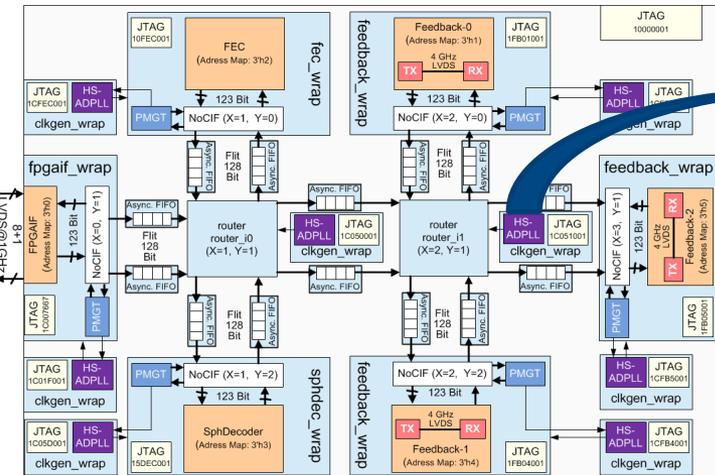
Abstraktion

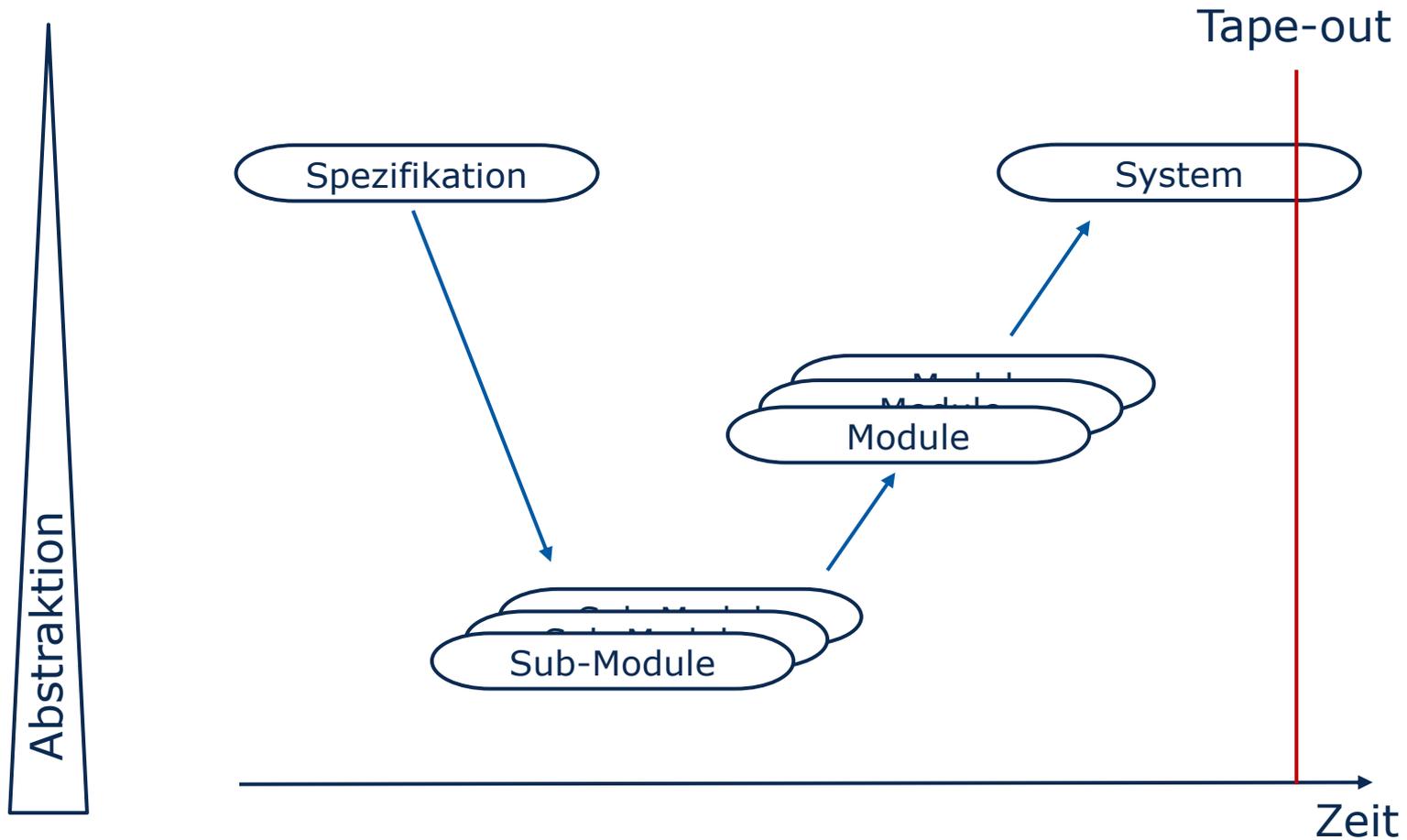
- Hierarchische Gliederung von komplexen Systemen



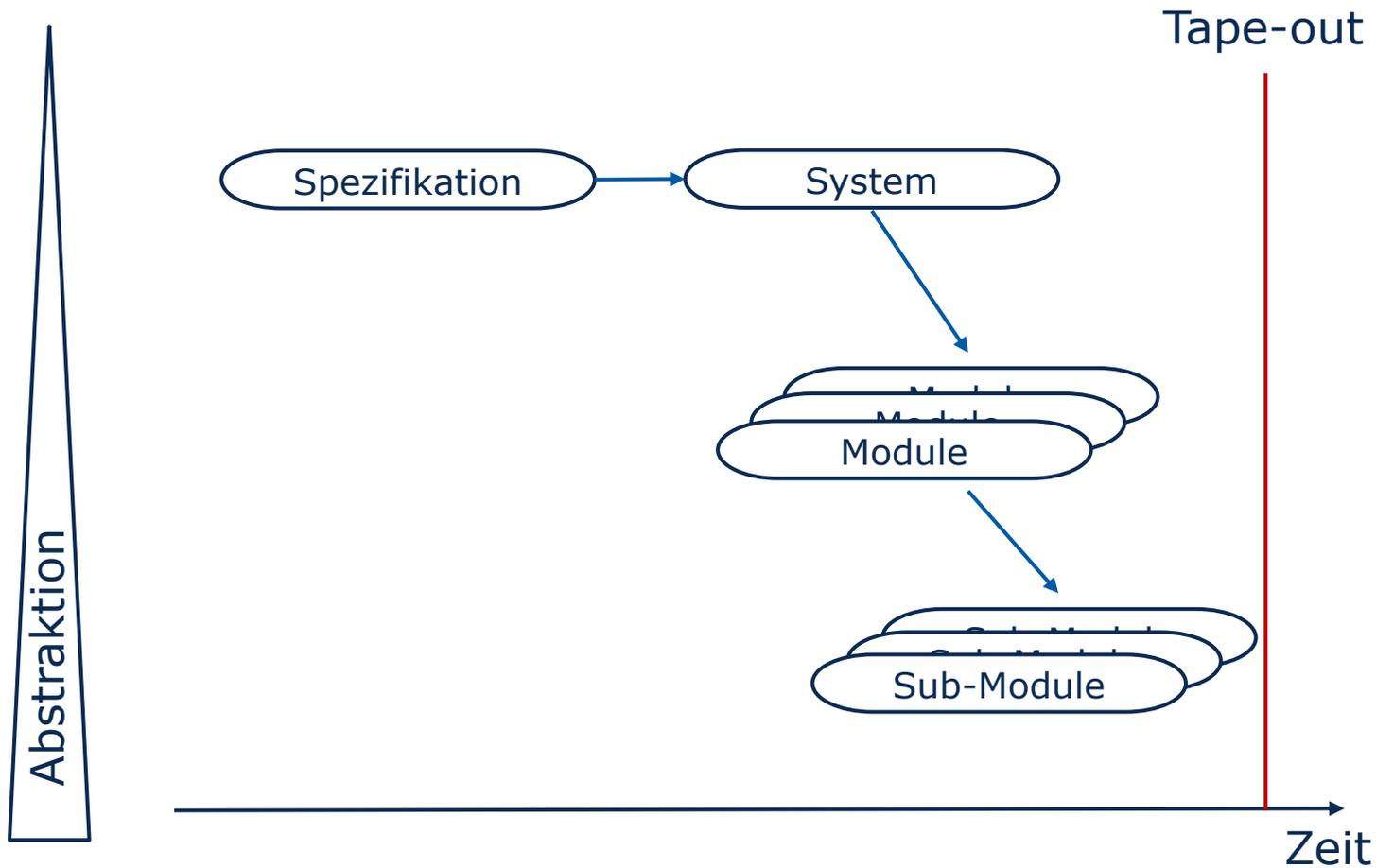
Beispiel: Multiprozessor System „Tommy“

- Beispiel: Volldigitaler Phasenregelkreis (ADPLL) Taktgenerator





- Entwurf von Sub-Modulen basierend auf Systemspezifikation
- Zusammenfügen der Module zum Gesamtsystem
- Vorteile 😊
 - Strukturierte Herangehensweise für kleinere Baublöcke in kleinen Entwurfsteams
 - Geeignet für Sub-Blöcke mit erhöhtem Entwurfsaufwand (z.B. Analog, Mixed-Signal)
 - Geringer zusätzlicher Modellierungsaufwand
 - Benötigt keine vollständige Spezifikation bei Entwurfsbeginn
- Nachteile 😞
 - Ggf. aufwändige Re-Design Zyklen nötig
 - Implementierungsarbeit im Team erfordert sehr genaue Spezifikation
 - Nur sequentielle Entwurfsabläufe möglich



- Entwurf des Systems basierend auf Systemspezifikation
- Modellierung von Modulen- und Sub-Modulen
- Implementierung von Modulen- und Sub-Modulen
- Vorteile 😊
 - Strukturierte Herangehensweise für komplexe Systeme in größeren Entwurfsteams
 - Parallele Entwurfsarbeiten möglich
 - RTL Implementierung
 - Synthese
 - Place&Route
 - Verifikation
 - Schnelle Verfügbarkeit eines simulierbaren Gesamtsystems (virtueller Prototyp) für
 - Performance Abschätzungen
 - Applikationsentwurf (PCB, Firmware)
- Nachteile 😞
 - Detaillierte Spezifikation bei Entwurfsbeginn benötigt
 - Modellierungsaufwand für Module und Sub-Module

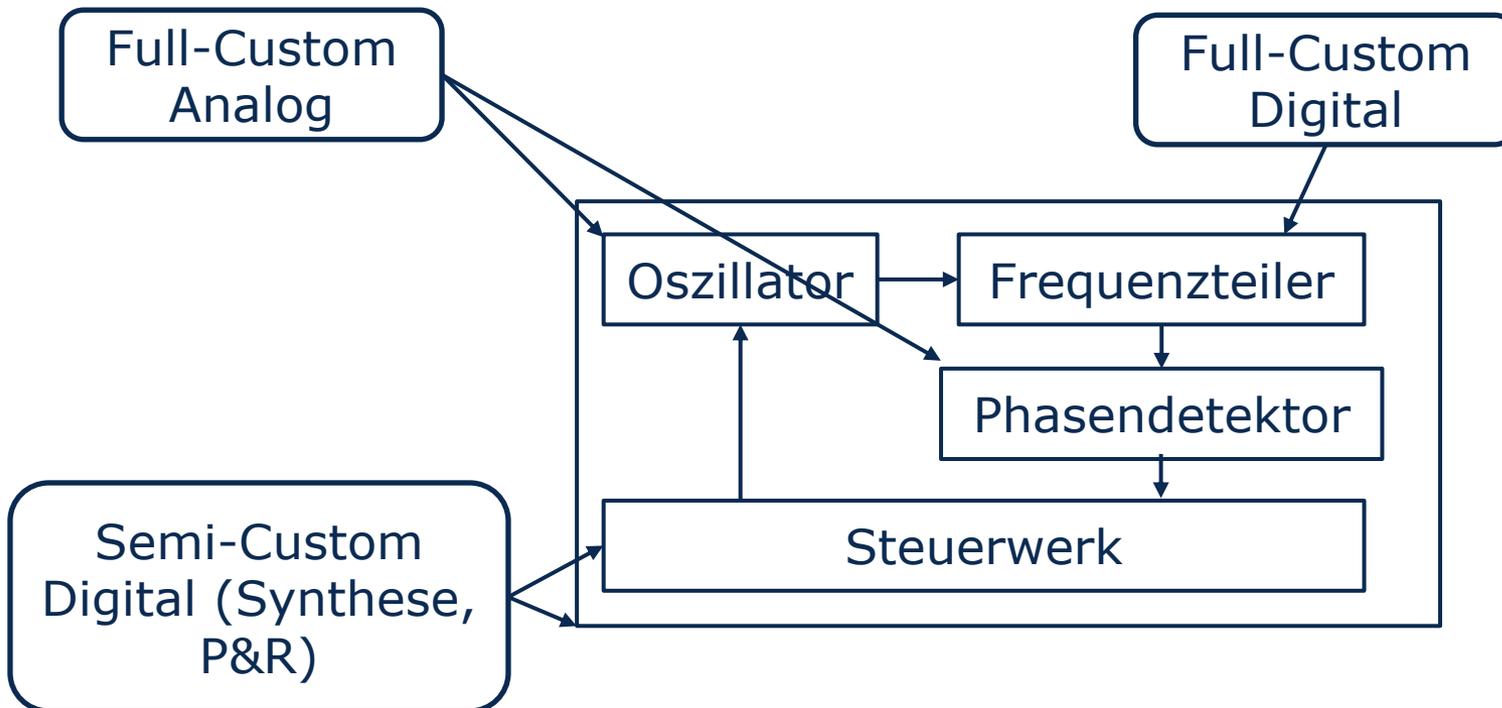
- Möglichkeit der Simulation von Systemkomponenten in unterschiedlichen Abstraktionsebenen (Views)
- Grundlage des Top-Down Designs
- Beispiel: Implementierung eines ADPLL Taktgenerators:

behavioral

RTL

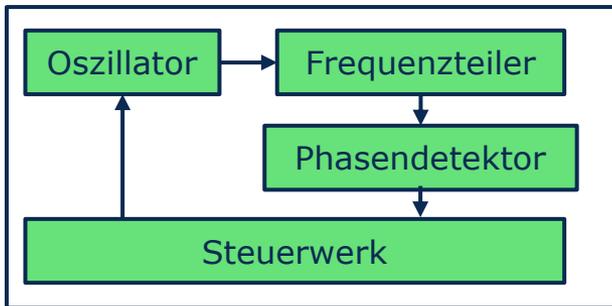
gate

transistor

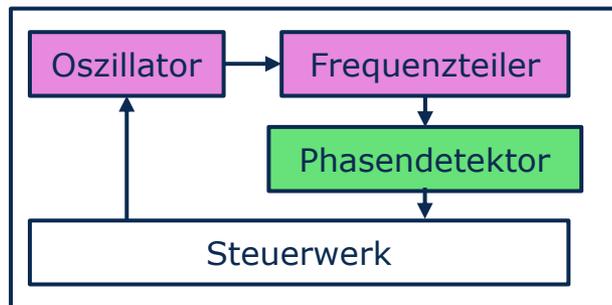
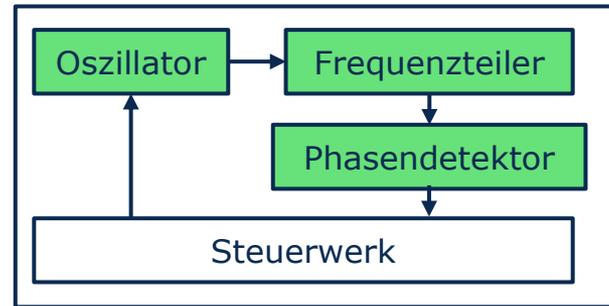


- Mixed-Level Repräsentationen in unterschiedlichen Entwurfsphasen

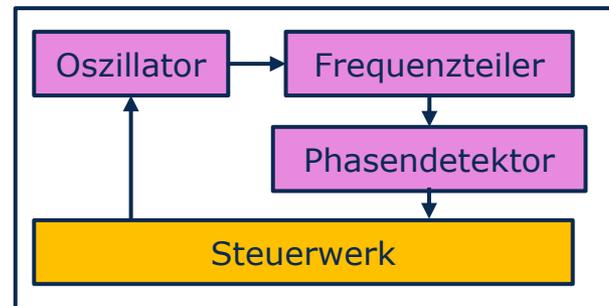
Systementwurf



RTL Verifikation



Oszillator Verifikation



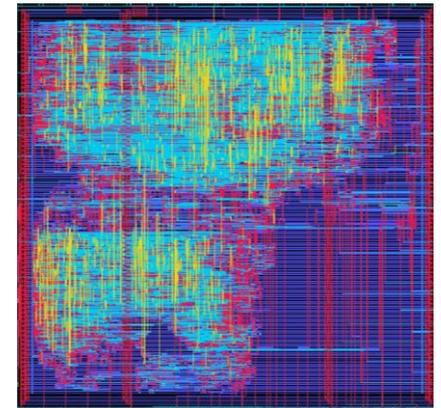
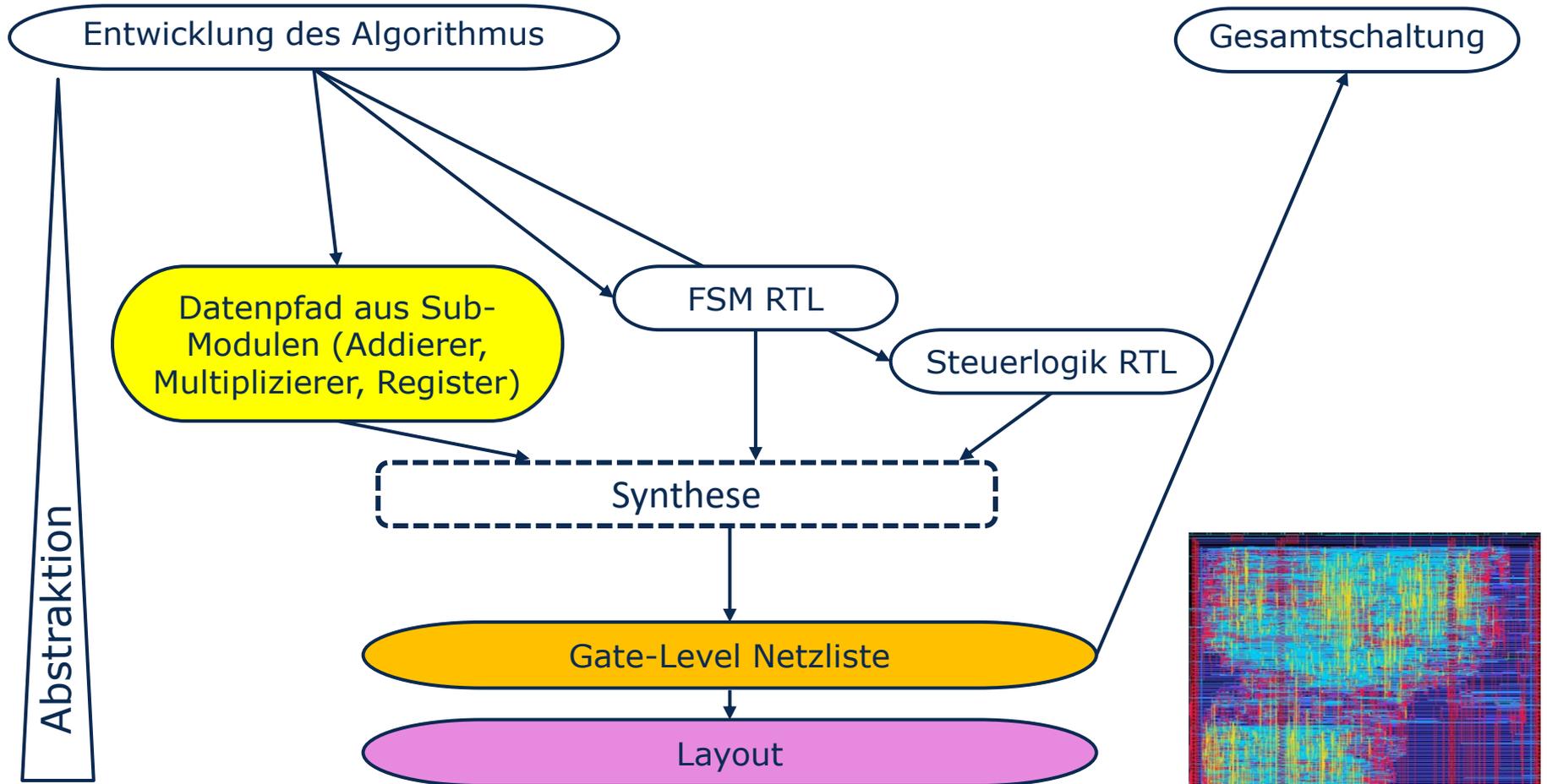
Post-Layout Simulation

behavioral

RTL

gate

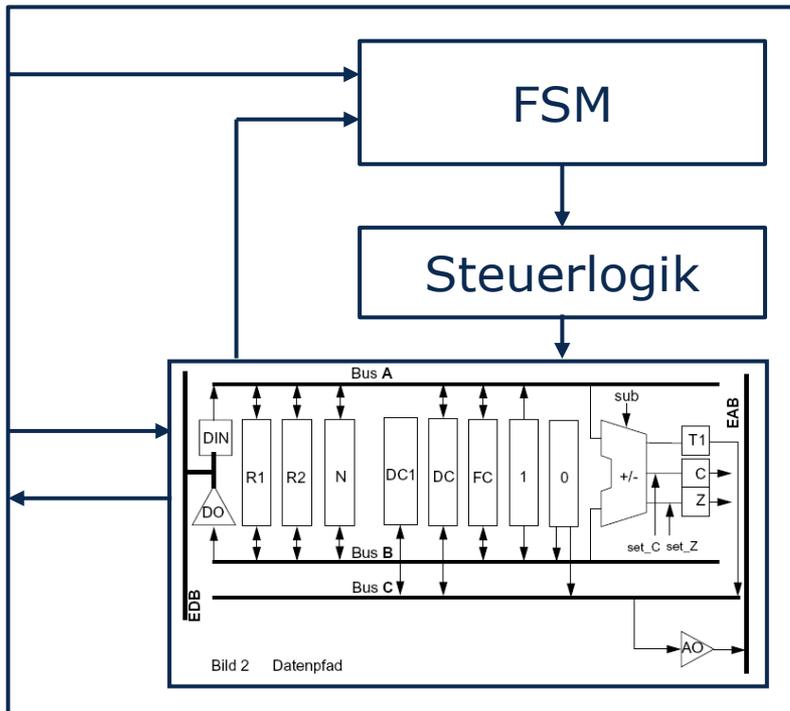
transistor



- Design Struktur

RTL

Gate Netlist



Place&Route Netzliste

- Mehr Infos: → siehe Praktikumsanleitung