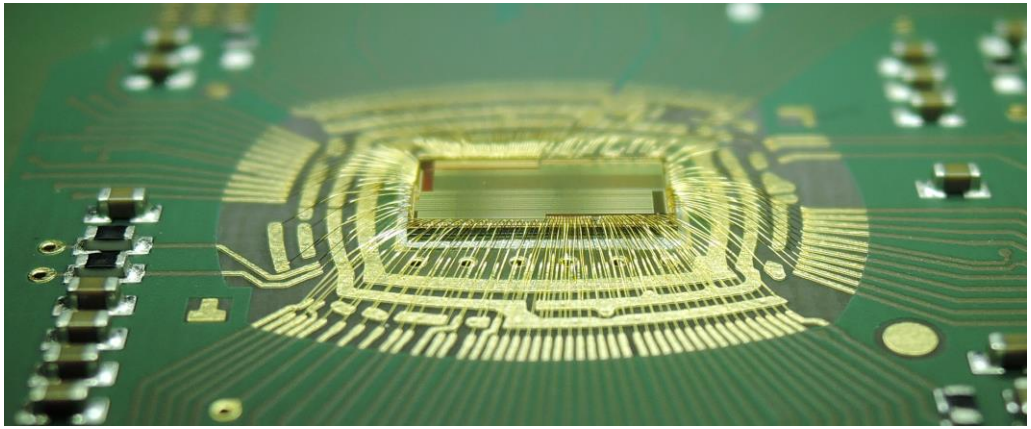
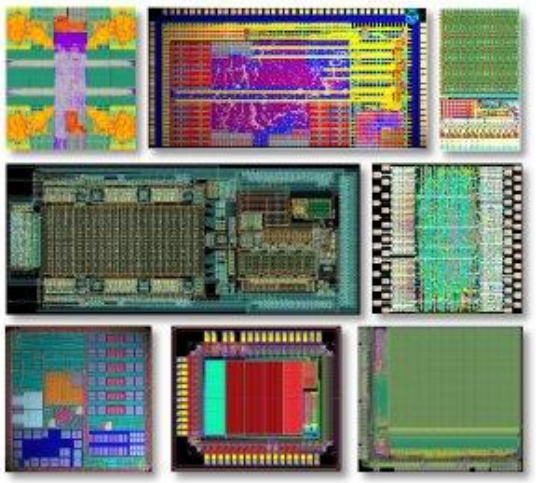
Circuit and System Design



FAKULTÄT **ELEKTROTECHNIK** UND **INFORMATIONSTECHNIK**

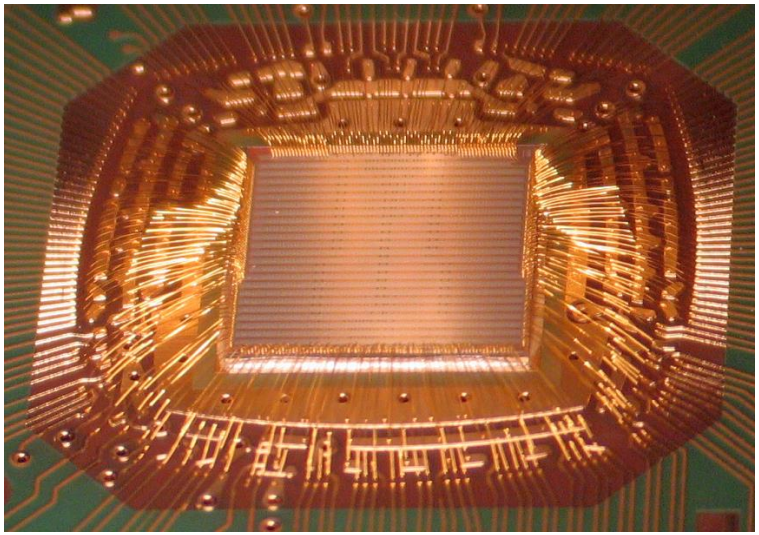
DRESDEN concept
Exzellenz aus Wissenschaft und Kultur



- Dr.-Ing. Sebastian Höppner
- Room TOE216
- sebastian.hoeppner@tu-dresden.de
- https://tu-dresden.de/ing/elektrotechnik/iee/hpsn

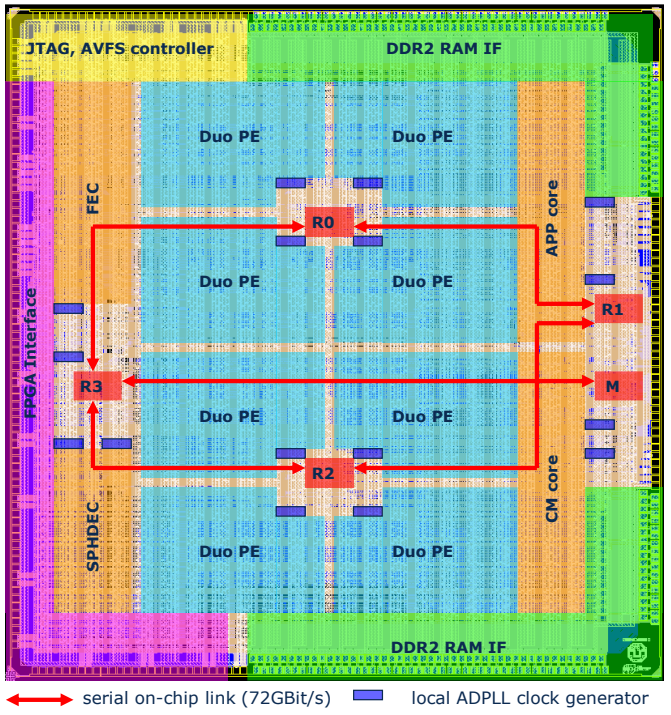
- Lecture slides webpage:
  - https://tu-dresden.de/ing/elektrotechnik/iee/hpsn/studium/materialien

- Design of integrated circuits („Chips“) as a core component of modern electronic products.
- Implementation of complex circuits on a chip makes it possible to have:
  - High Functionality
  - High Integration Density
  - Low Power Consumption
  - High Reliablity
- The downsizing of fabrication technologies (Miniaturisation) enables the integration of more complex systems.
- Challenge in design and verification.



Complex Mutliprocessor System-on-Chip

- **„Tomahawk2"** Software-Defined-Radio Baseband Processor
  - chip area of 36 mm² in 65nm CMOS Technology
  - 465 Pads
  - 10,2 million Gates
  - 750kByte SRAM
- 20 processor cores
- Adaptive and Dynamic Power Management
- 17 Clock Generators
- Network-on-Chip with serial on-chip links with data rate up to 72GBit/s
- DDR2 memory interface
- FPGA Interface with 10GBit/s

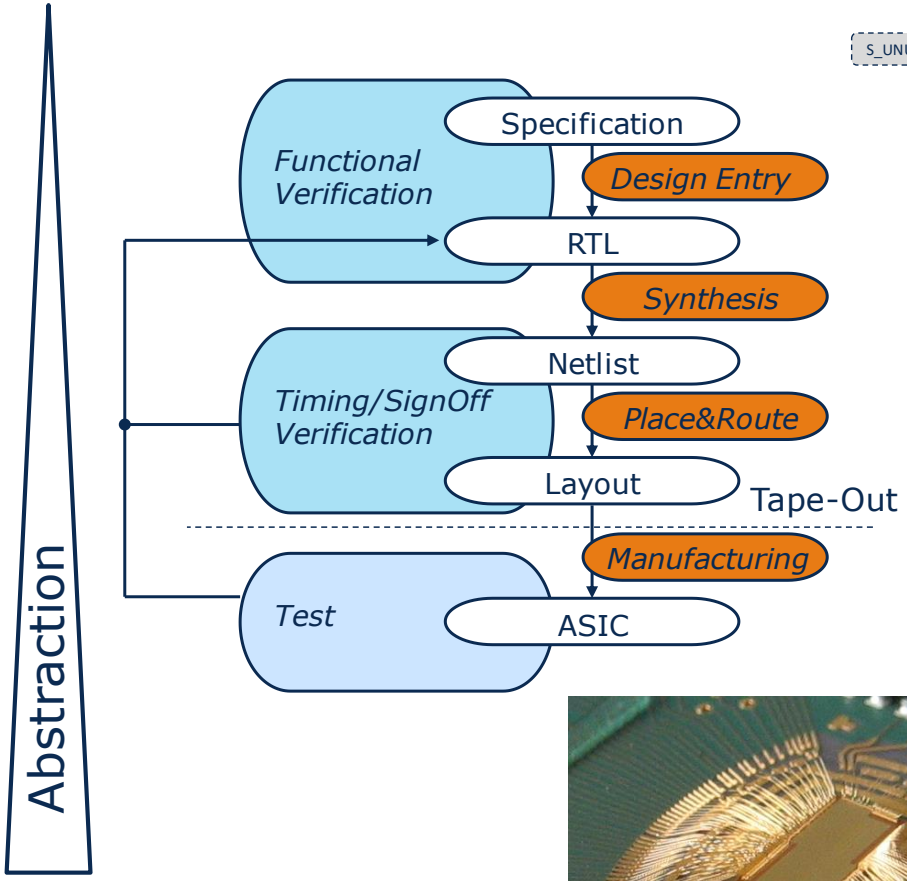


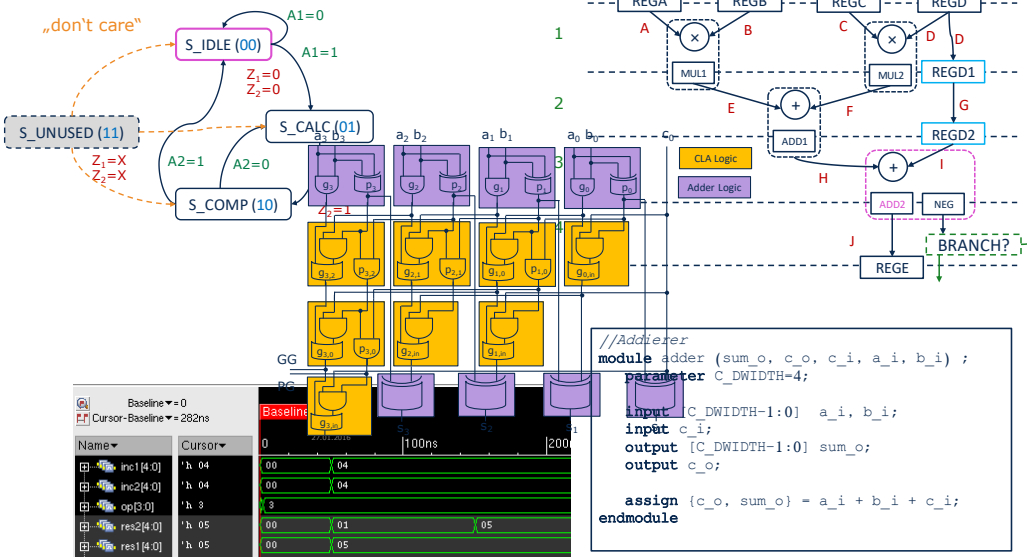
Developed by HPSN and Vodafone Chair (Prof. Fettweis) in the CoolBaseStations project

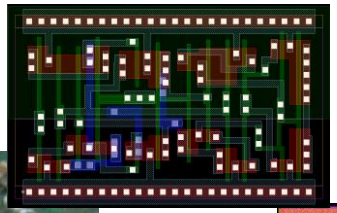
Content of the Course

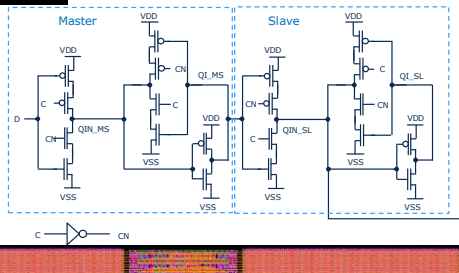
- Design Flow of Digital Integrated Cicuits
- Hardware Implementation of Algorithms
- Verilog Hardware Description Language
- Circuit Simulation and Verification
- Digital Circuits in CMOS Technology

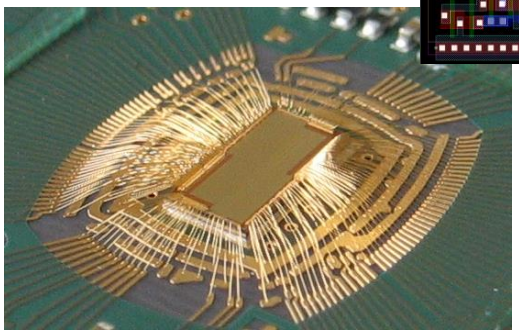
Content of the Course

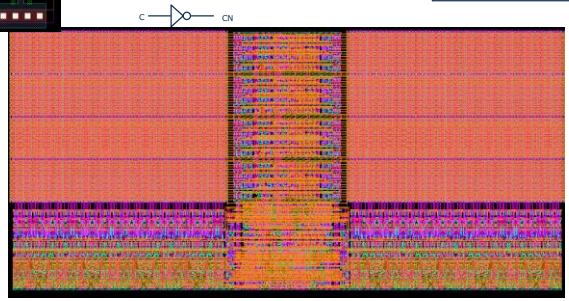












Paperwork

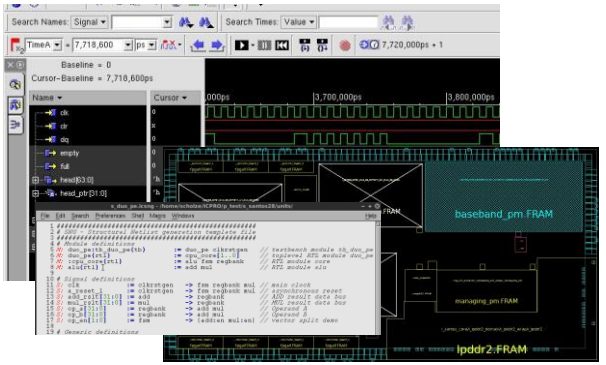
- Hardware implementation of own choice algorithm
- Reference Implementation in Software
  - Architcture Development (Scheduling, Datapath, Flow Control)
  - Circuit implementation using datapath building blocks
  - Verification via circuit simulation
  - Circuit Synthesis and Place&Route
- Written document(Details in Exercise!)
- Electronic submission (.pdf)
- Submission :
  - Exam registration in the semester for the submission to take place!
  - **Deadline until 15.09. (SS) resp. 15.03. (WS)**

Course Objectives

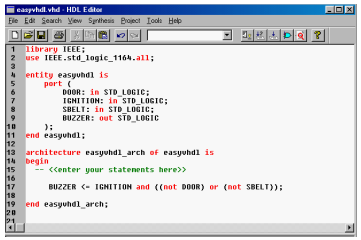
- This course will:
  - present digital flow for digital integrated circuits
  - present methods of Hardware implementation for algorithms
  - give basics and concepts of Verilog HDL
  - present verification strategies
  - help you deeply understand the basics of implementing digital circuits in CMOS technology
  - give practical experience for implementing a digital circuit
  - give motivation for self-study

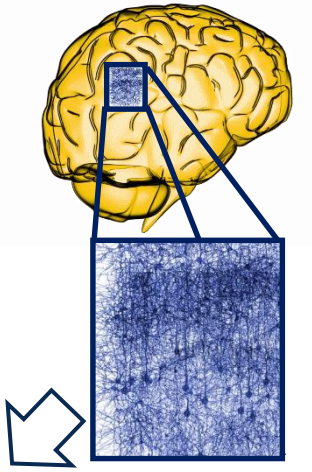
Further Lectures of the HPSN chair

- **VLSI Processor Design**
  - RTL design of a processor
  - Simulation and Verification
  - Top-Down Design RTL2GDS Flow (Synthesis, Timing Analysis, Place&Route, Power Analysis, Sign-off Analysis)
  - Project: Implementation of a processor

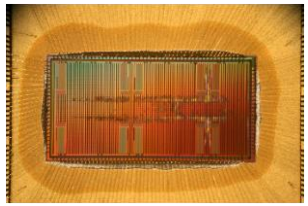


- **VHDL-Design (Information Technology Seminar)**
  - Introduction to Design, Modelling, Verification using VHDL
  - Presentation of tasks and solutions from current research work of the chair
  - Presentations of participants to their projects

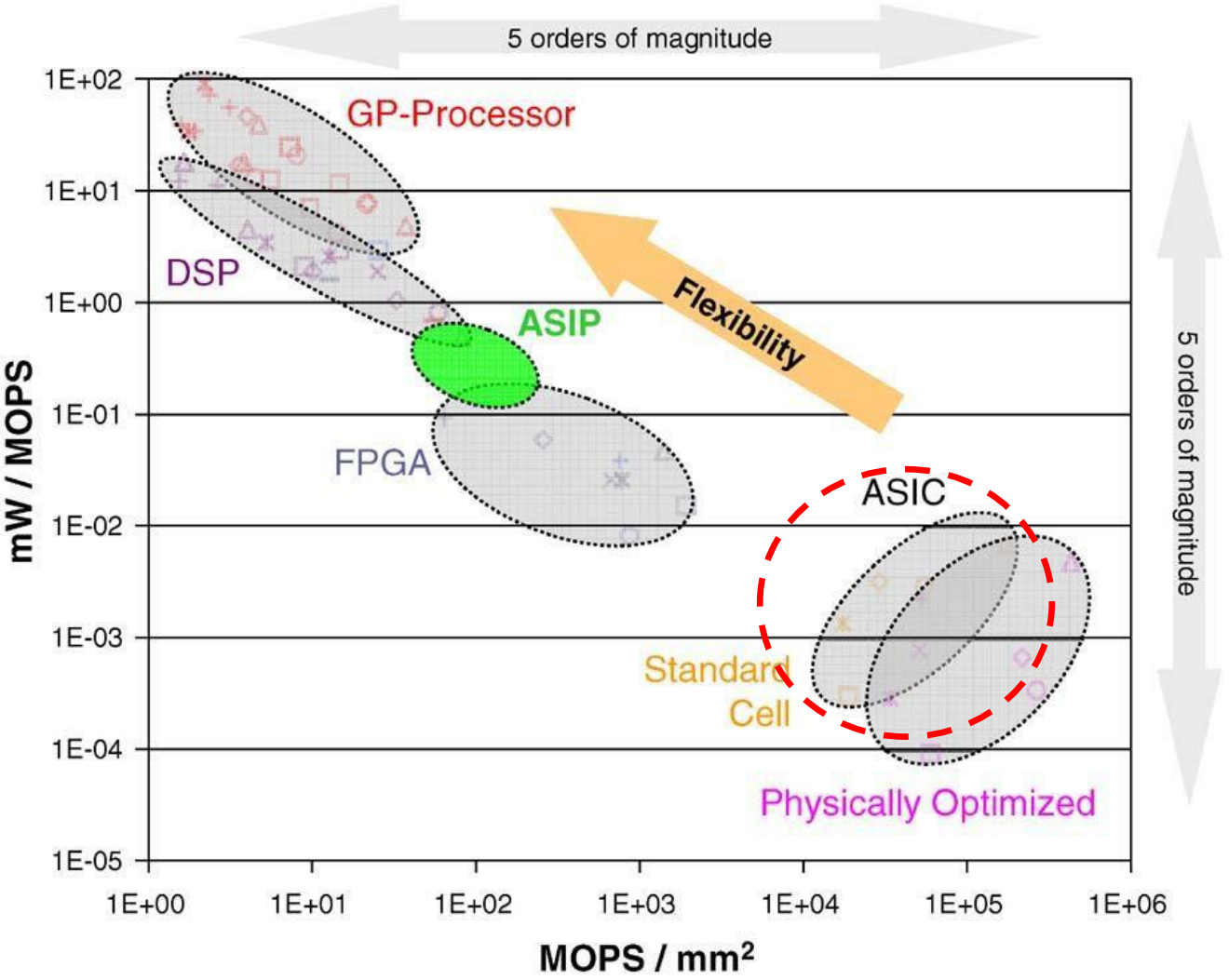




- **Neuromorphic VLSI-Systems**
  - Design methods for analog integrated CMOS circuits and their dimensions
  - Neuromorphic VLSI-Systems und their neurobiologic basics, applications for example, in Brain-Machine-Interfaces
  - Project for creating and analysing analog and neuromorphic CMOS circuits with Cadence DF2 Design Tool

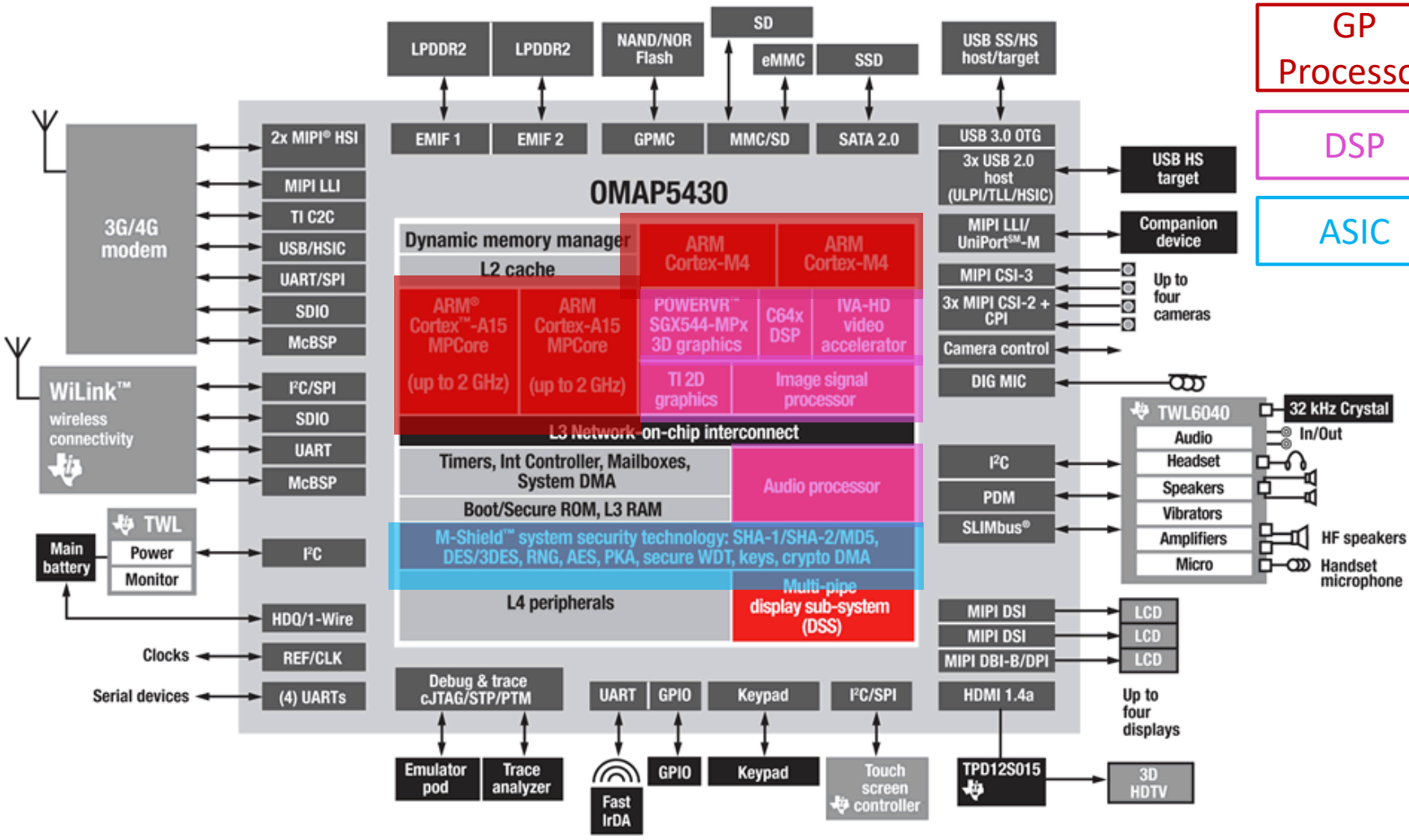


Application Specific Integrated Circuits (ASICs)



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 ReconfigurableSystems, Springer, 2010

Example on System-on-Chip: TI OMAP5430

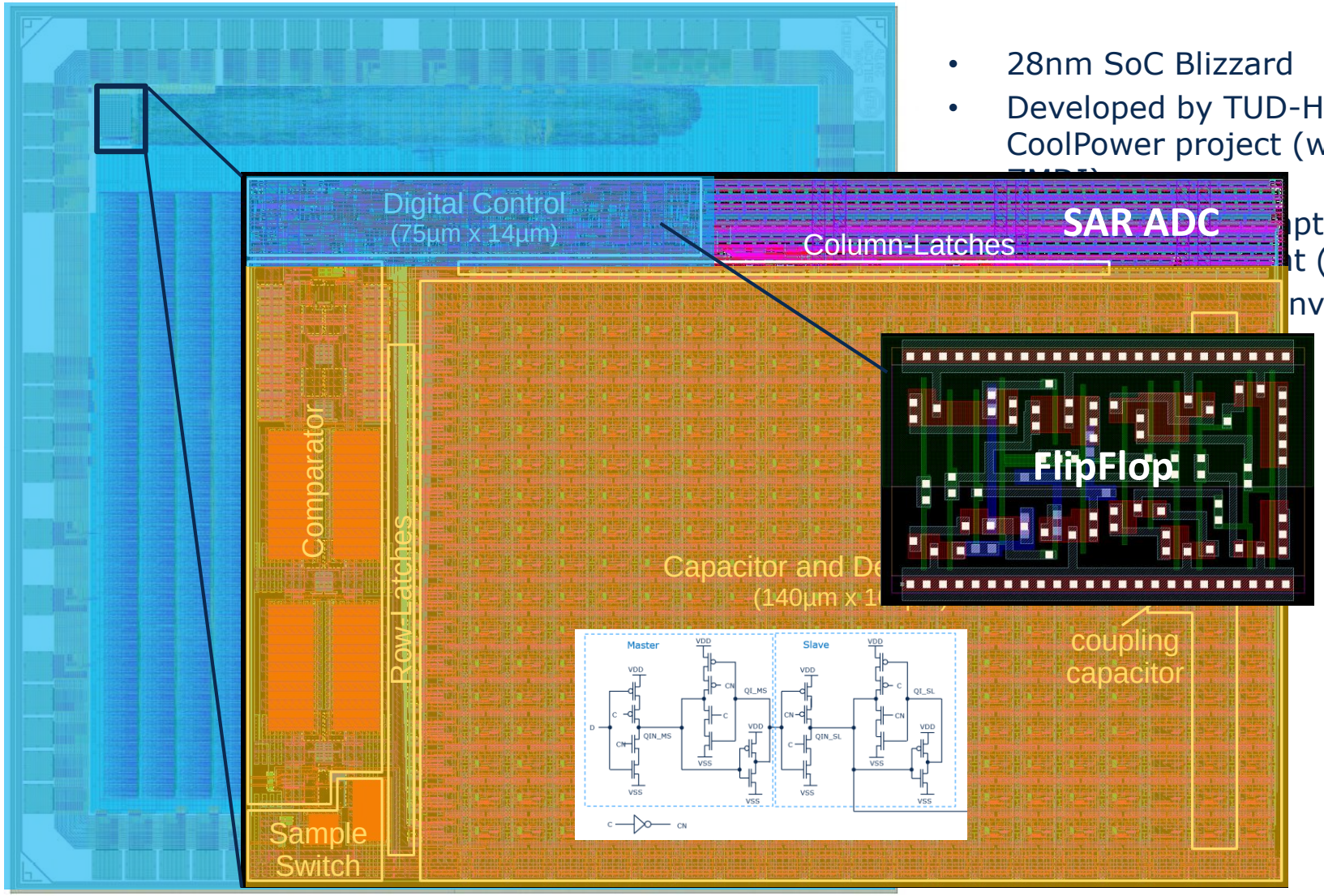


Quelle: anandtech.com

Design Styles

- **Full-Custom Design**
  - Manually drawing circuit diagram (schematic) and layout
  - Advantages:
    - Detailed optimizations are possible
  - Disadvantages:
    - Low productivity
  - Applications:
    - Analog and Mixed-Signal circuits
    - Basic elements of digital circuits (standard cells, I/O cells, memory)
    - Specially optimized digital circuits (e.g, High-speed, Ultra-Low-Power)
- **Semi-Custom Design**
  - Manually writing a Hardware Description Language (HDL)
  - (Partially-) automted genertion of design data (netlist, layout)
  - Advantages:
    - Very high productivity
  - Disadvantages:
    - Limited optimization of circuit properties
  - Applications:
    - Complex digital circuits
    - Systems-on-Chip

Example: Design Style (System-on-Chip Blizzard)



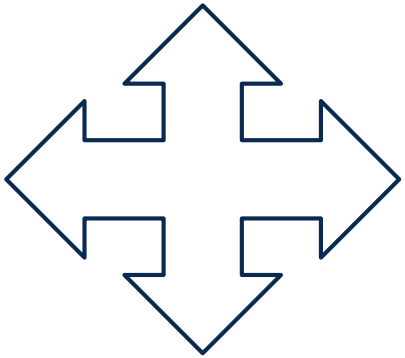
- 28nm SoC Blizzard
- Developed by TUD-HPSN in CoolPower project (with
- ptive
- t (AVFS)
- nverter

Market

- Requirements (Specification)
- Customer
- Chip prise
- Development time (Time-to-Market)

Additional Value

- New functionality (e.g new wireless standard)
- Better functionaltiy (e.g longer battery life)
- Smaller form factor (housing size)
- Reduced number of chips in the system
- Reliability



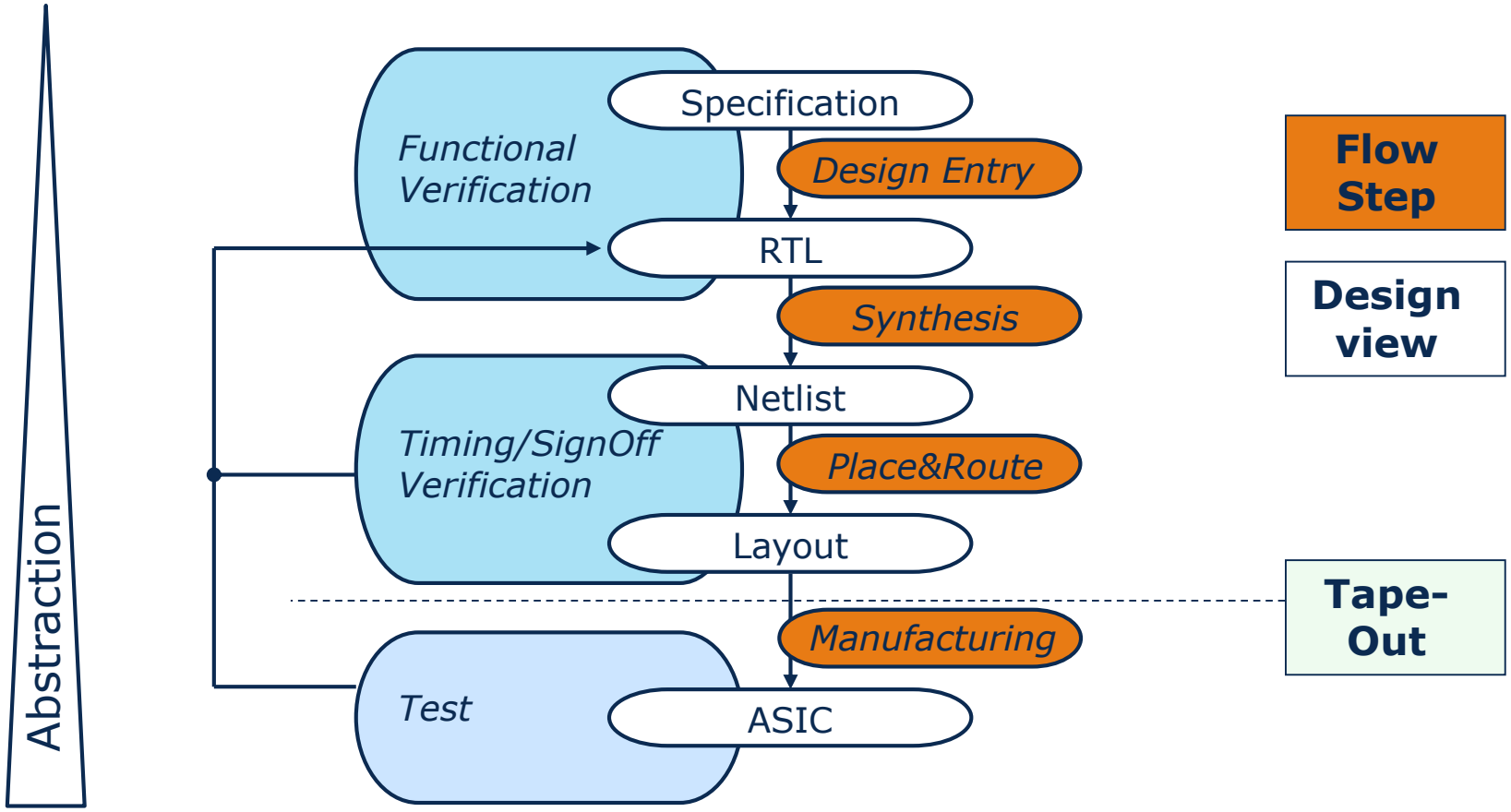
Costs

- Development (Design, Implementation, Verification)
- IP Purchase
- Production (Masks, Wafer production, Packaging)
- Testing the produced chips

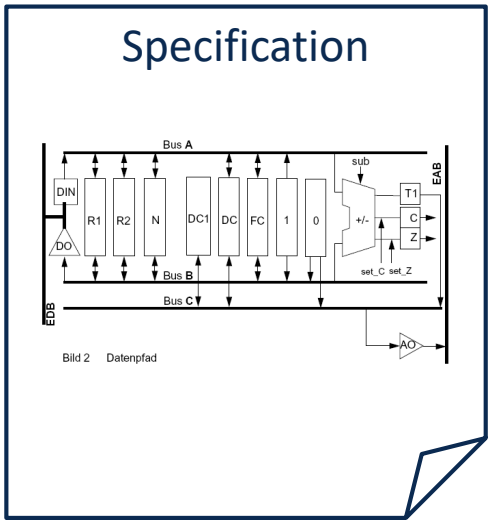
- Achitecture decision
- Technology decision
- Package
- Design methodology

Design Process of Integrated Digital Circuits

Design Flow: Semi-Custom Digital Design



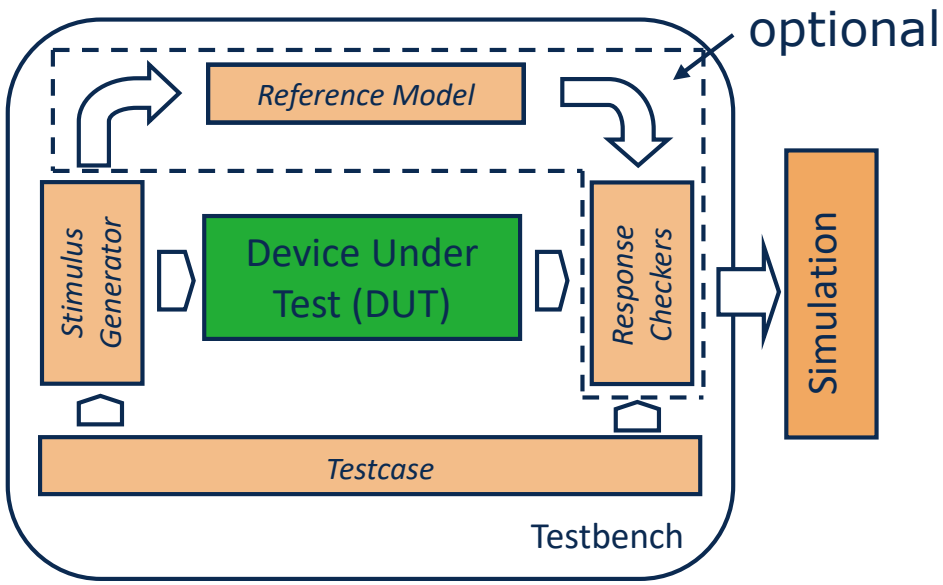
- Design of circuit models based on system specification and described as Register-Transfer-Level (RTL)
- Use of Hardware Description Language (HDL) such as Verilog, VHDL
- Synthesizable description
- Check of RTL Coding Guidelines




```
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
```

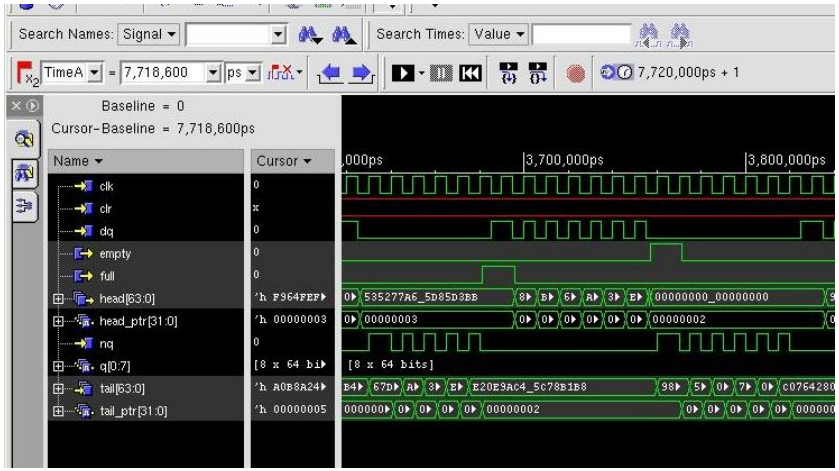
Testbench implementation

- Testbench: Simulation environment for a system module
- Input signals generator
- Response checker for output signals according to specification
- Optional: Comparing with a reference model is possible
- Testbench usually implemented with HDL as a behavioral description

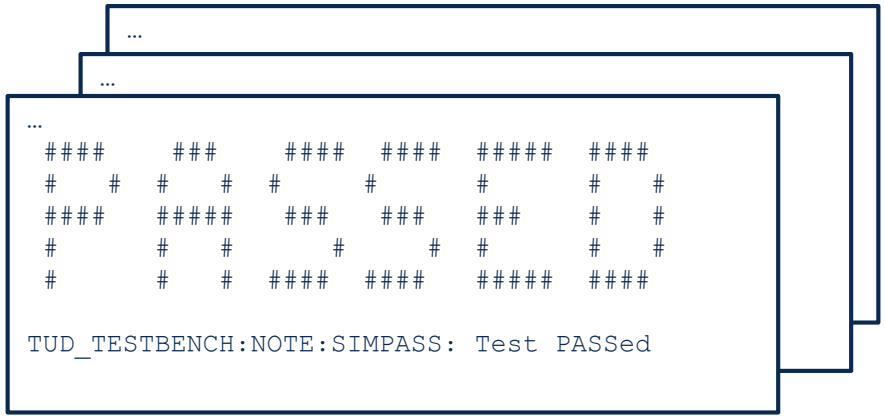


- Testbench simulation with event-based digital simulator
- Tools, i.a.
  - Cadence NCSIM
  - Mentor Questa

Interactive Analysis/Debugging



Self-checking Testbench
Test PASS/FAIL

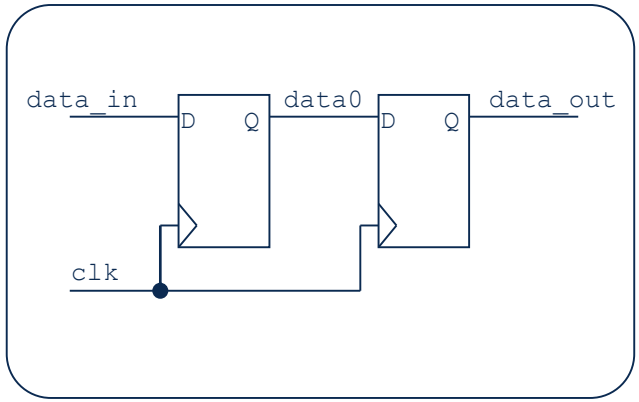


- Representing a synthesizable RTL description into a gate netlist based on standard cell libraries
- Library File (.lib) includes information about functionality, timing, power
- Specification of constraints, e.g. clock frequency, input and output delay
- Tools, i.a.
  - Synopsys DesignCompiler
  - Cadence RTLCompiler

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

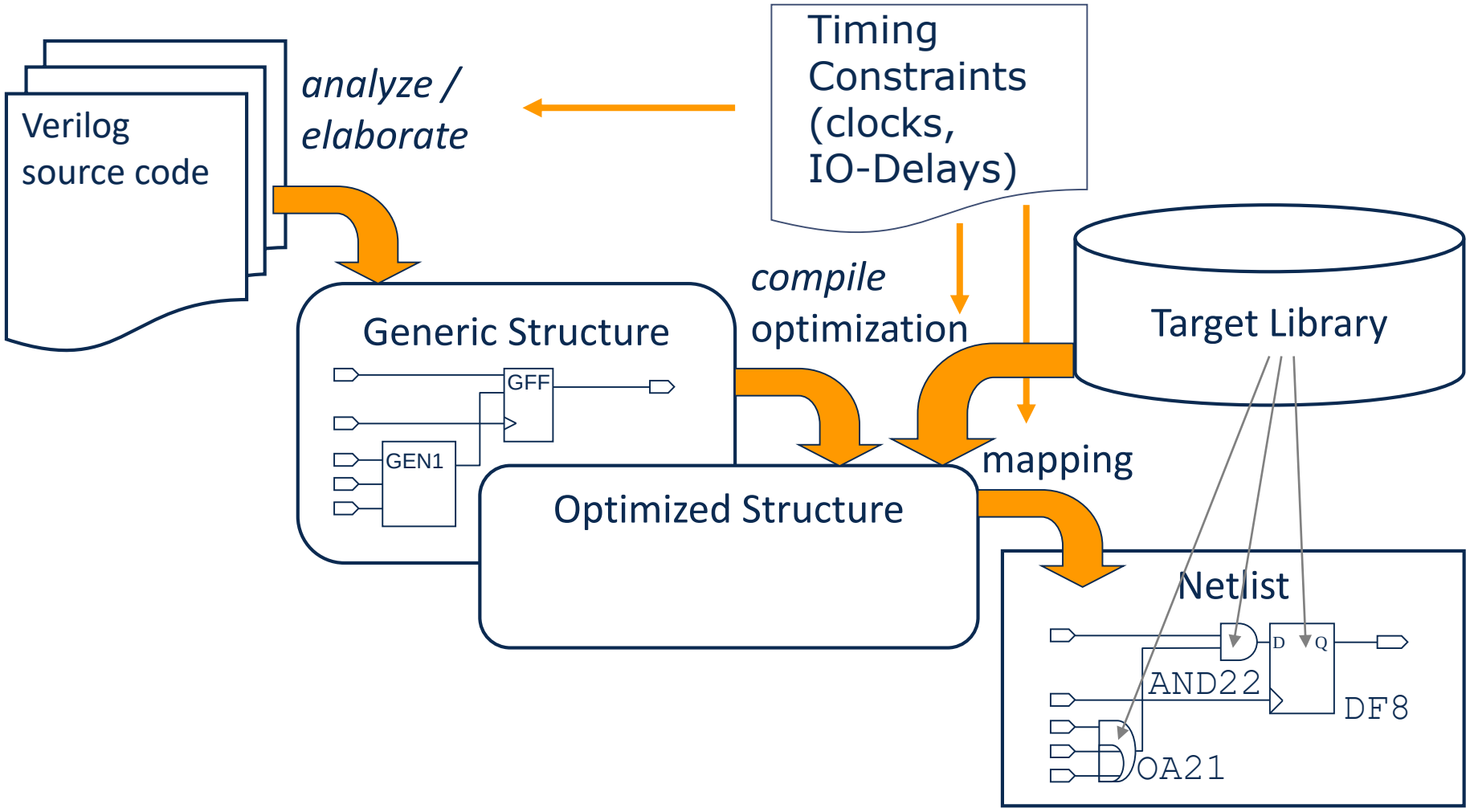
Verilog RTL

RTL Synthesis

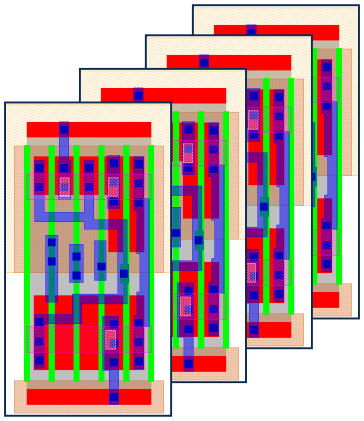


Equivalent gate netlist

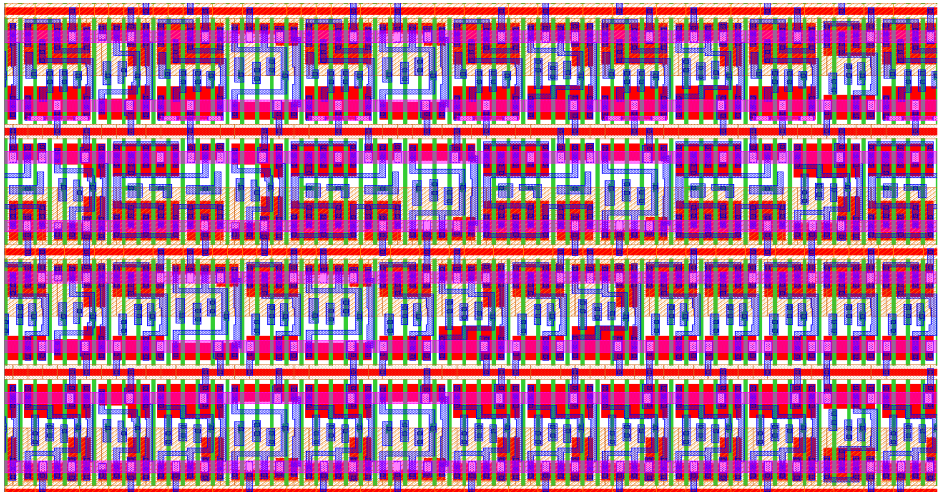
Logic Synthesis Procedure



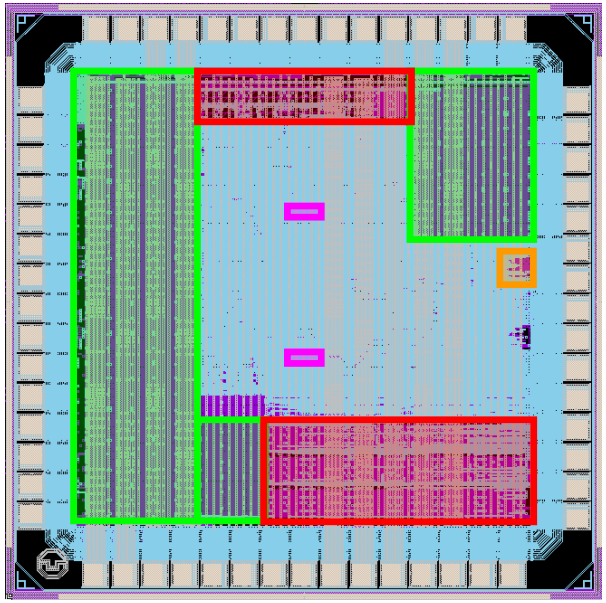
- Automated placement and wiring of standard cells
- Synthesis of Clock Trees
- Extraction of parasitic layout elements



Standard cells

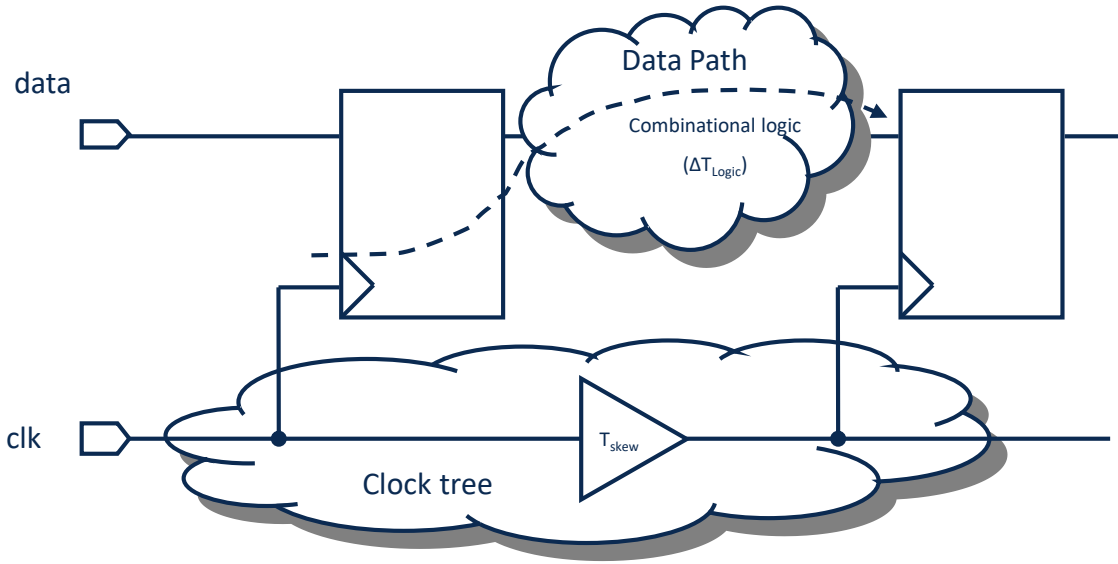


Circuit block



Chip

- Timing constraints check

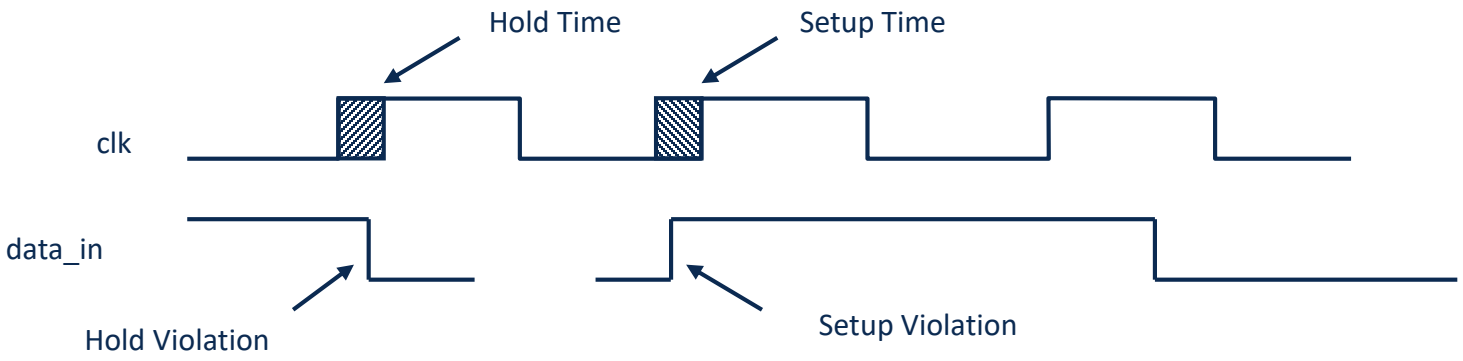


Setup Check:

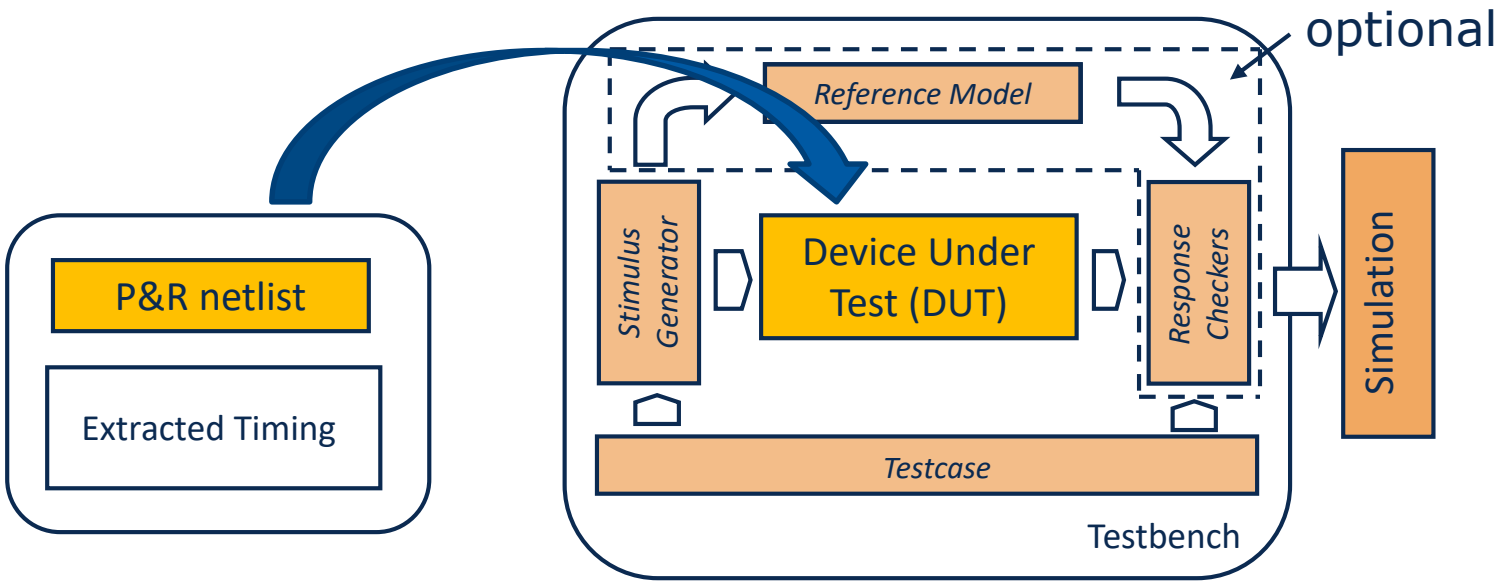
$T_{clk} + T_{skew} \geq \Delta T_{Logic} + T_{setup}$

Hold Check:

$T_{skew} + T_{hold} \leq \Delta T_{Logic}$



- Annotation of delay times to the gate netlist
- Individual delay and constraint checks (Setup, Hold) for every gate
- Time consuming

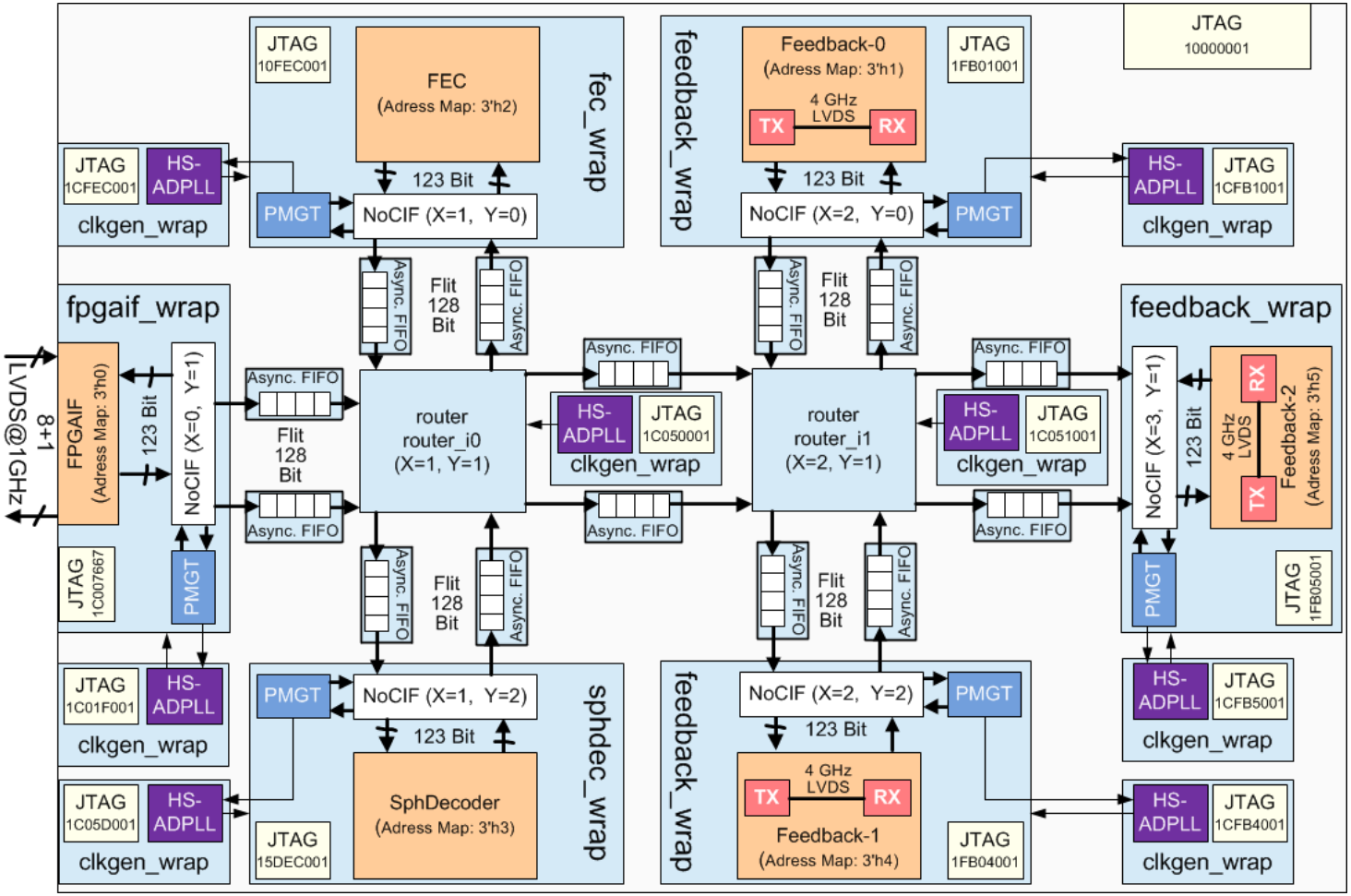


Design Views

- A circuit module can be represented differently:

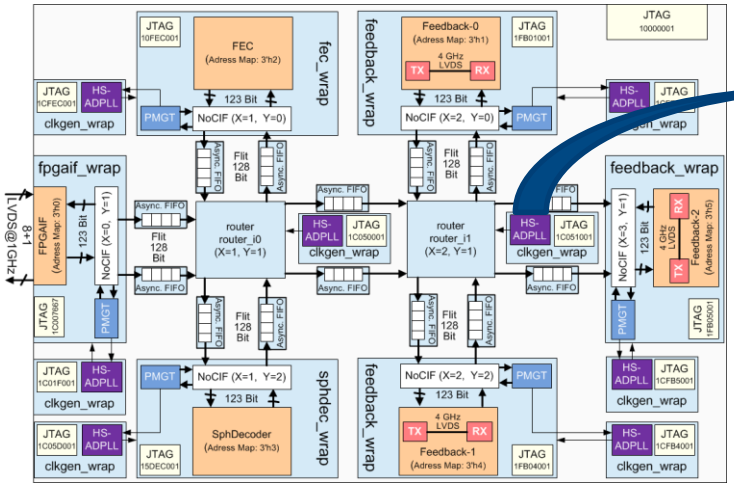
  - Behavioral Model
    - Behavioral model for digital simulator
    - „Executable System Specification“
  - RTL
    - Synthesized RTL description
  - Gate Netlist
    - Gate netlist as a result of synthesis/P&R
  - Transistor-Netlist (schematic netlist)
    - Netlist on Transistor level
  - Transistor-Netlist (extracted netlist)
    - Netlist with Transistors and parasitic layout elements
  - Layout
    - Physical representation of the circuit geometries to be fabricated

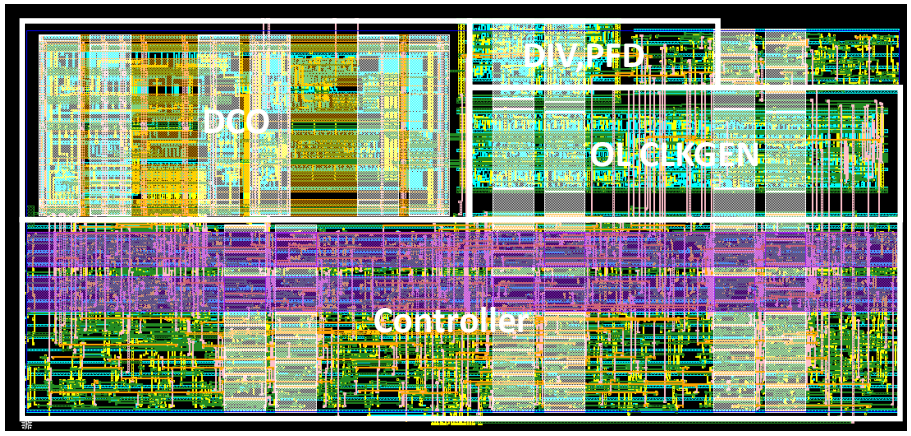
- Hierarchical structure of complex systems

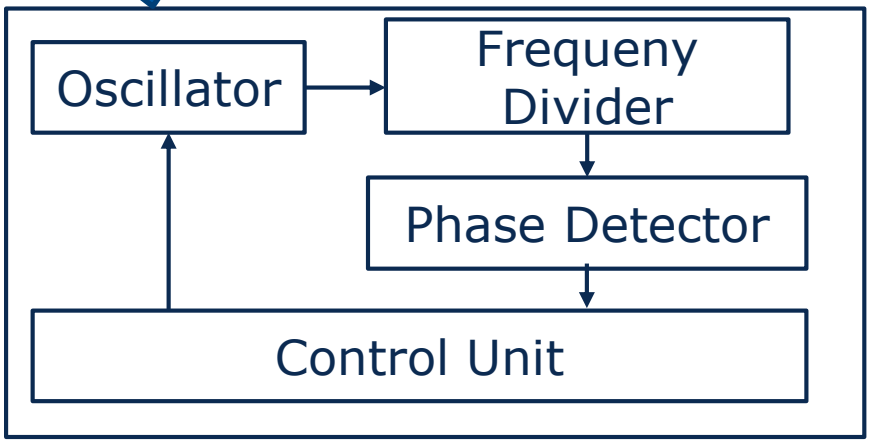


Example: Multiprocessor System „Tommy“

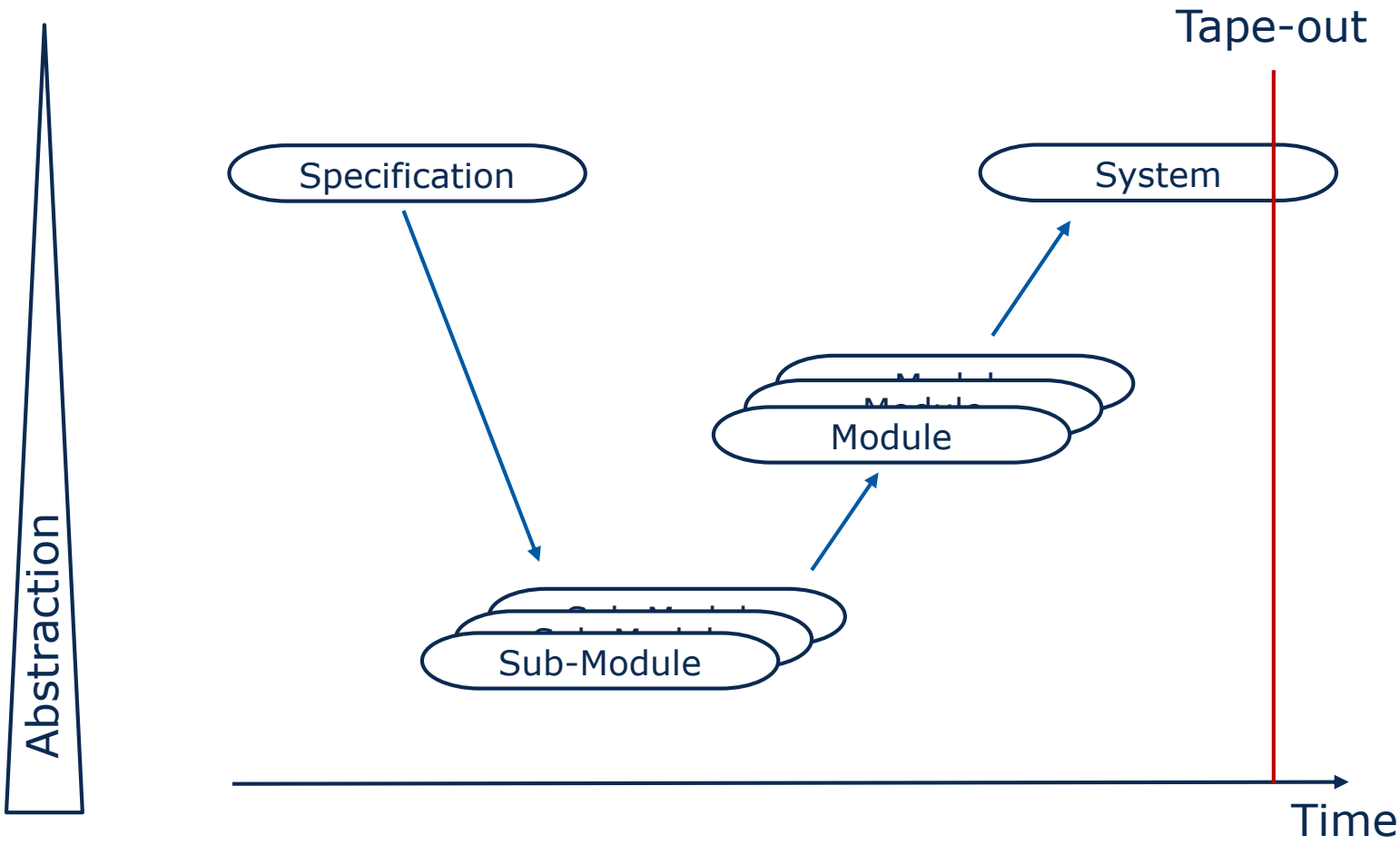
- Example: All Digital Phase Locked Loop (ADPLL) Clock Generator







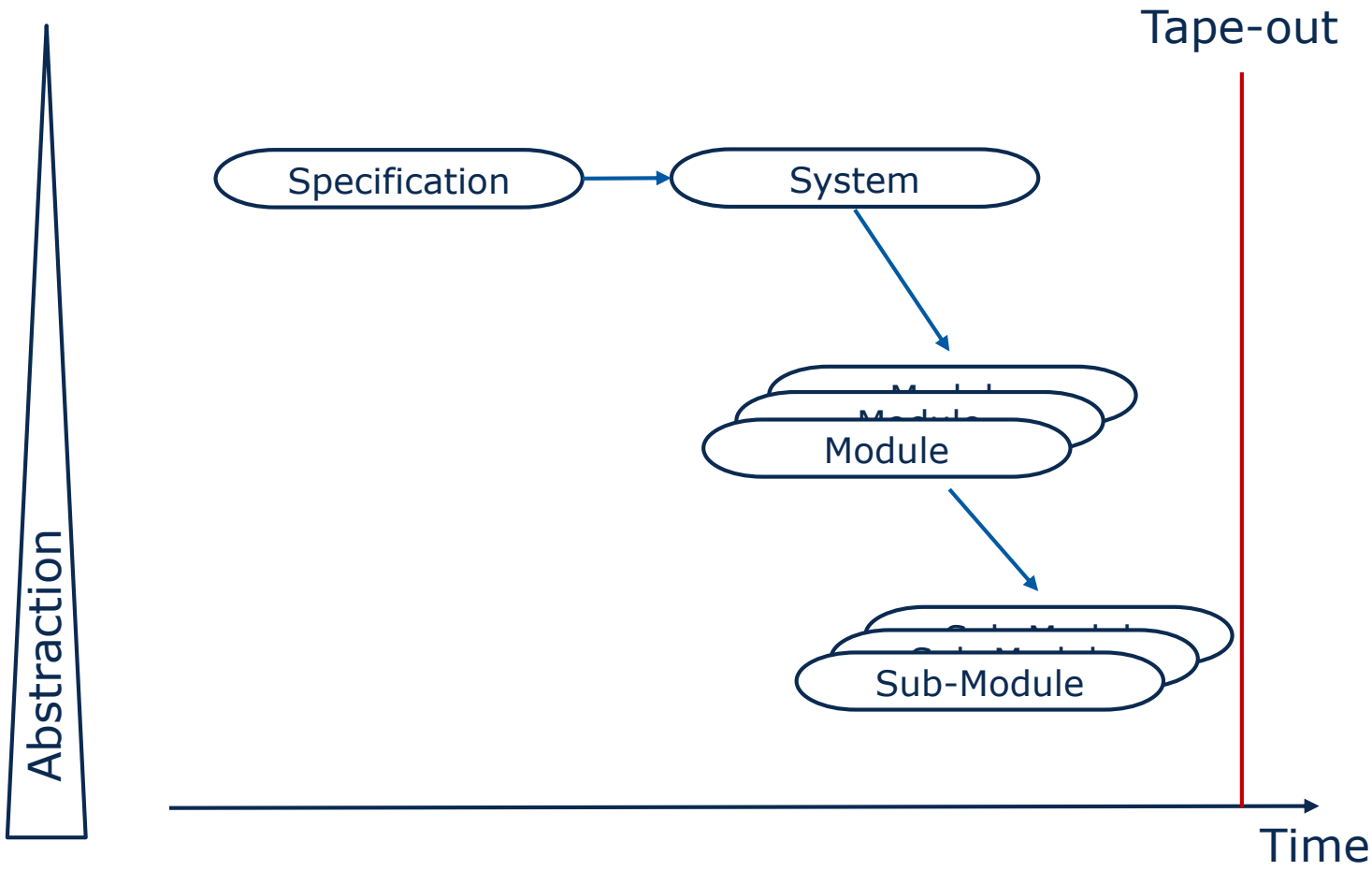
Bottom-Up Design Flow



Bottom-Up Design

- Design of sub-modules based on system specification
- Assembly of the modules to an overall system
- Advantages ☺
  - Structured approach for smaller building blocks in small design teams
  - Suitable for sub-blocks which need higher design effort (e.g. analog, mixed-signal)
  - Little additional modeling effort
  - Does not require complete specification to start designing
- Disadvantages ☹
  - Possibly. Costly re-design cycles are necessary
  - Implementation work in the team requires very precise specification
  - Only sequential design processes are possible

Top-Down Design



Top-Down Design

- System design based on system specification
- Modeling of modules and sub-modules
- Implementation of modules and sub-modules
- Advantages ☺
  - Structured approach for complex systems in large design teams
  - Parallel design work is possible
    - RTL Implementation
    - Synthesis
    - Place&Route
    - Verification
  - Fast availability of a simulated overall system (virtual prototype) for
    - performance estimates
    - Application design (PCB, Firmware)
- Disadvantages ☹
  - Detailed specification needed to start designing
  - Modeling effort for modules and sub-modules

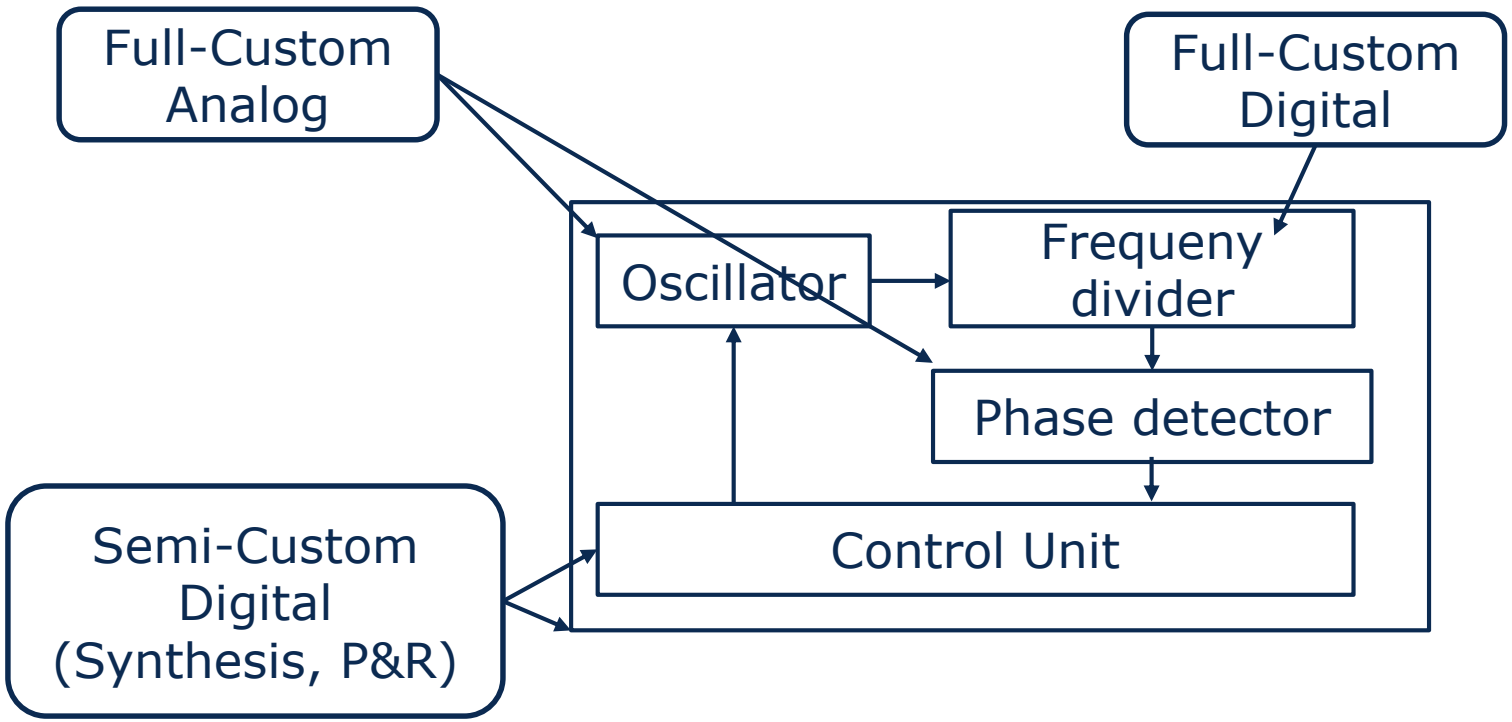
- Simulation of system components is possible on different abstraction levels (Views)
- Fundamentals of the Top-Down design
- Example: Implementation of an ADPLL clock generator:

behavioral

RTL

gate

transistor



- Mixed-Level Representations in different design phases

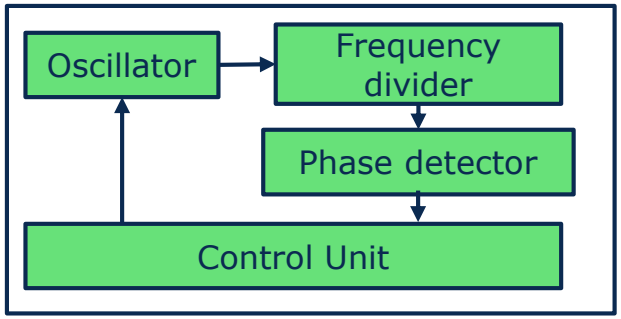
behavioral

RTL

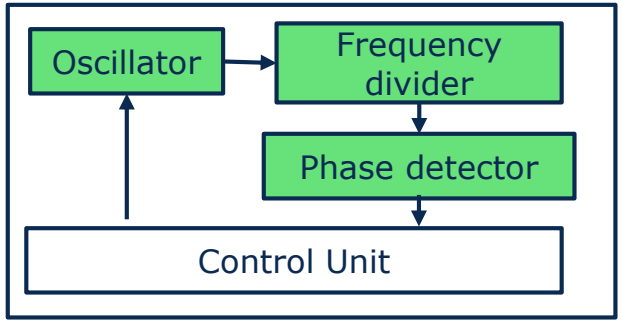
gate

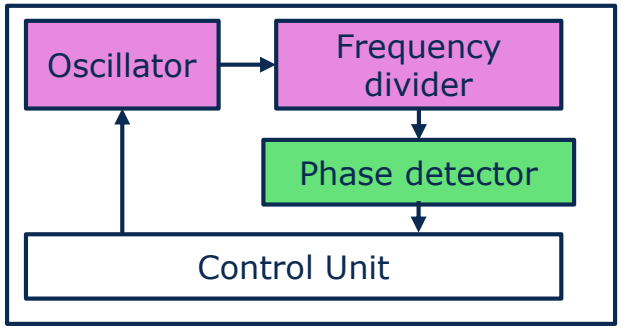
transistor

System Design

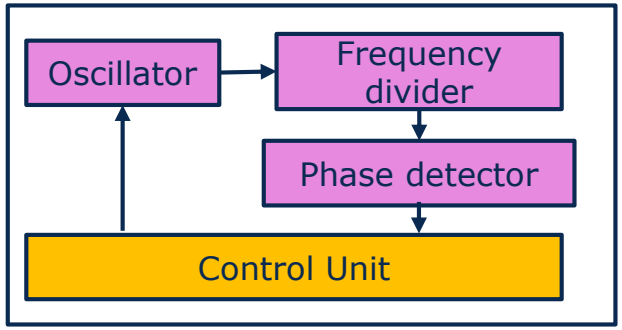


RTL Verification



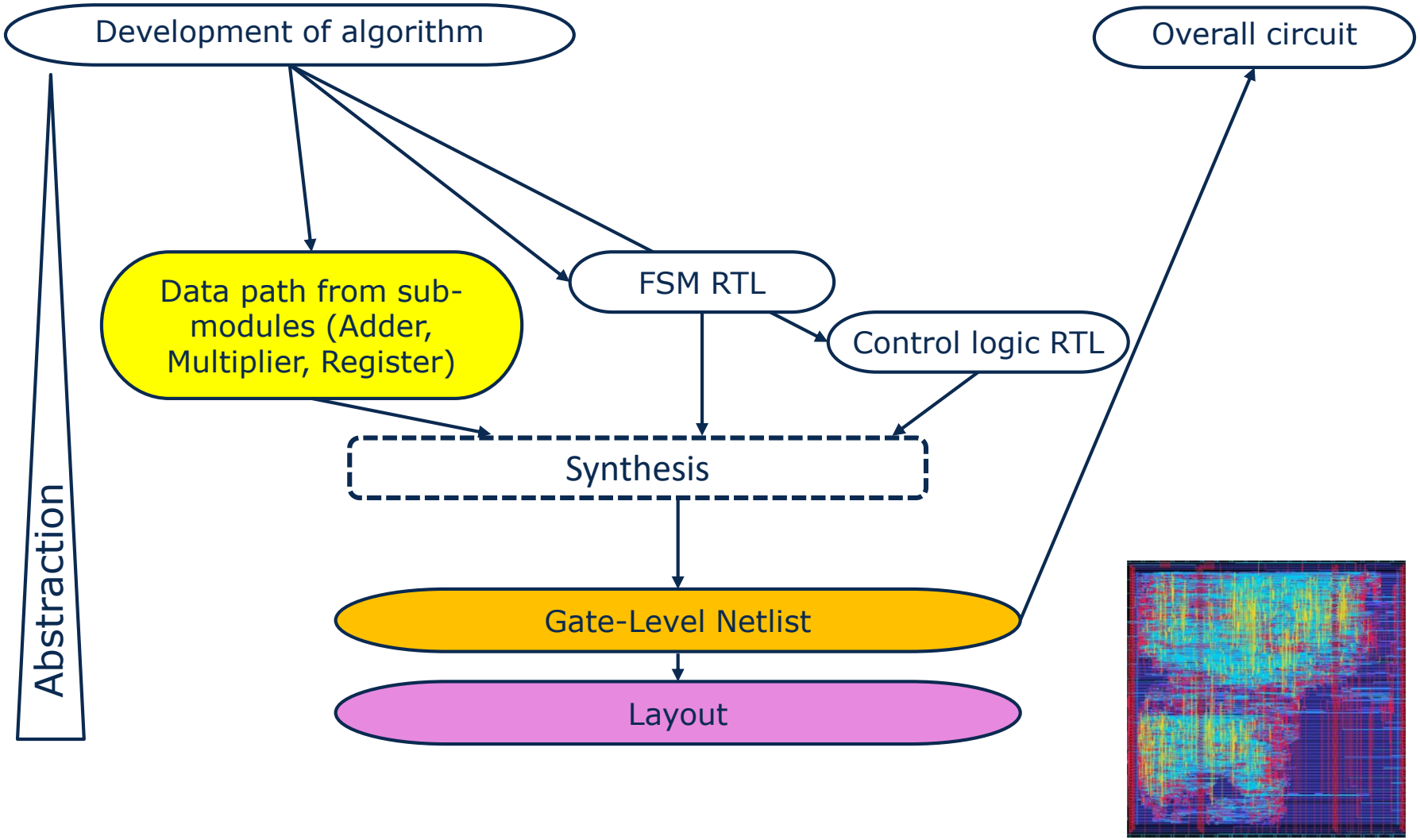


Oscillator Verification



Post-Layout Simulation

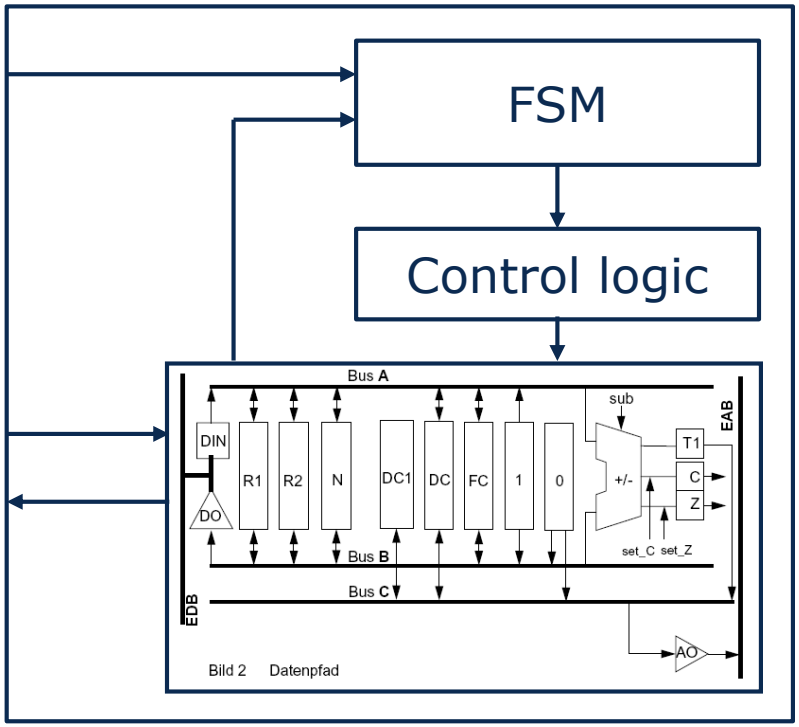
Design Flow in Circuit and System Design Course



- Design structure

RTL

Gate Netlist



Place&Route Netlist