SIEMENS
Ingenuity for life

Digital Chip Design

Herbert Taucher
Head of Research Group Electronic Design

Digital Chip Design

Outline of the lecture

Drivers for industrial ASIC design

ASIC design is the critical track in product design

Abstract view on ASIC design – Collaboration models with ASIC vendors

ASIC design flow in detail

- Requirements and specification
- ASIC Vendor selection
- The functional model
- The netlist
- Digital layout
- Production

History of Integrated Circuits

SIEMENS
Ingenuity for life

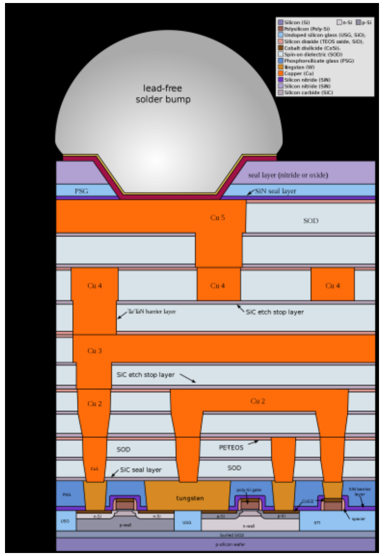
1936: Alan Turing invented the **Turing Machine**, which is a model of a general purpose computer
1947: A team around Grace Hopper working on Mark II coined the term “**Debugging**”
1949: Werner Jacobi (Siemens) invented a 5-transistor 3-stage **integrated amplifier**
1952: Grace Hopper invented the **Compiler** for “A-0”

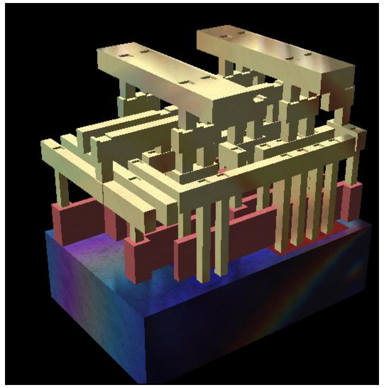
1958/1959: **First Integrated Circuit (IC) prototype** and patent filings

- Jack Kilby – How to **integrate** transistors
- Kurt Lehovec – How to **isolate** components on a semiconductor crystal
- Robert Noyce, Jean Hoerni – How to **interconnect** individual transistors

1961: Fairchild announced their **first commercial IC** series. Price dropped from US$1,000 to US$20–30
1965: Gordon Moore described in a paper a **doubling of number of components** per IC every year
1974: Robert Dennard – Power density for transistors stays constant, so that the power use stays in proportion with area: both voltage and current scale downwards with length. **Performance per watt** of computing **is growing exponentially**.

2000: Jack Kilby was awarded with the **Nobel Prize** in Physics for his part in the invention of the Integrated Circuit



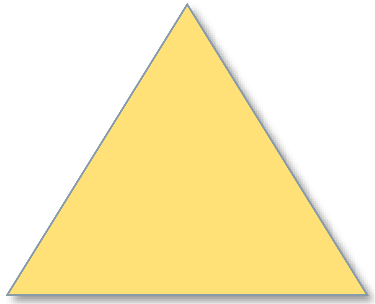


Source: Wikipedia

Drivers for industrial ASIC design

SIEMENS
Ingenuity for life

There are three drivers for industrial ASIC design processes



Performance, Power,
Complexity, etc.

Costs,
Time-to-Market

Quality, First-Time-Right,
Reliability, Lifetime

- How to make best use of technical properties of the Silicon technology node?
- How to minimize costs and Time-to-Market?
- How to ensure a one-pass ASIC design and how to ensure reliability and lifetime?

Why ASIC?

ASIC unveils the full potential of silicon technology nodes

There are many choices for implementation of functionality available

- **Commercial processor chips**: good cost position due to economy of scale but limited performance, power efficiency and real-time capabilities
- **Commercial processor chips + FPGA**: A costly solution especially for higher volume
- **FPGA-only**: Soft processor cores with limited performance are the limiting factor; high silicon overhead compared to ASIC
- **SoC-FPGAs**: A FPGA including a optimized processor subsystem on chip supports custom solutions at a good price point for mid-range volume
- **DSP**: A powerful compromise for specific applications in signal processing
- **ASIC**: The solution for high volume; best performance to power ration

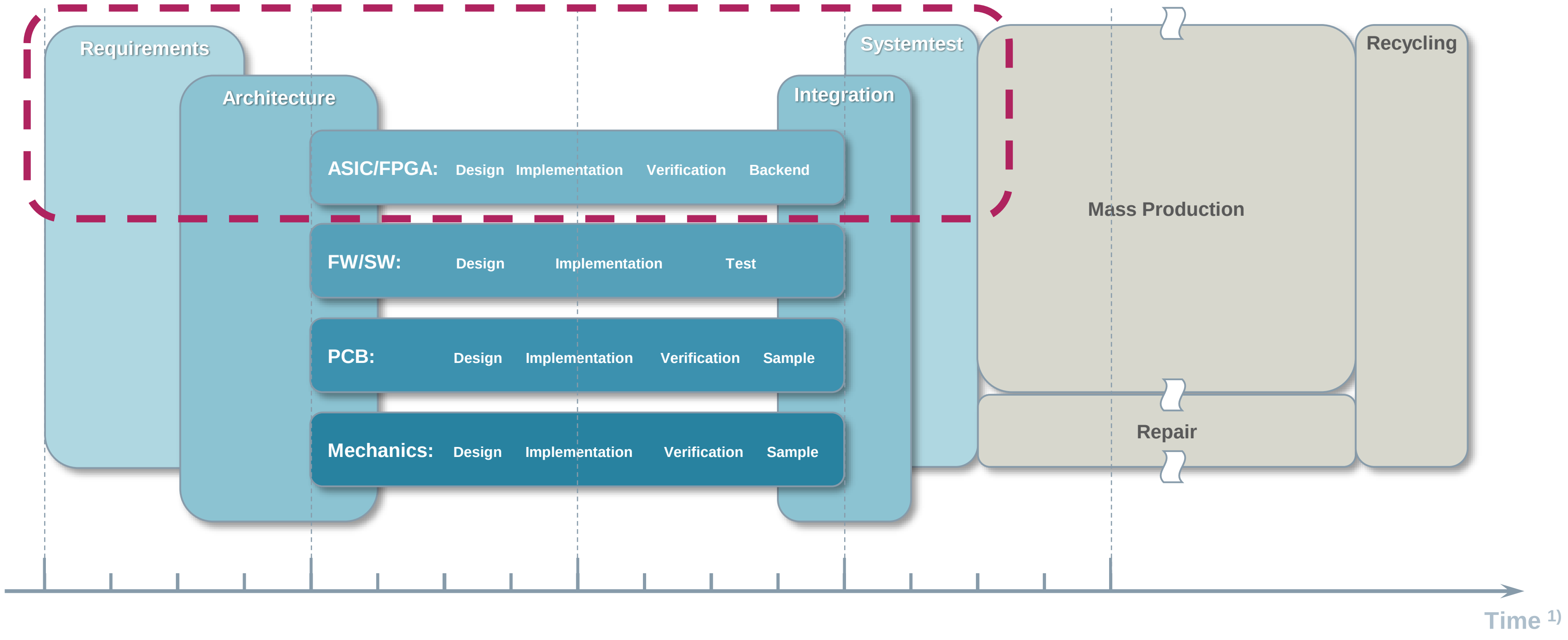
Trends

- DSPs are loosing ground due to DSP-capabilities of ARM-based processor chips
- SoC-FPGAs show increasing market shares
- ASIC market consolidated on vendor and customer side

Product design integrates various disciplines – one of them is ASIC design

SIEMENS
Ingenuity for life

- ASIC design flow typically defines the **critical path** of the product design
- ASIC design introduces **risks** to products due to long bug fixing iteration cycles
- But products benefit from **highest performance at lowest power**



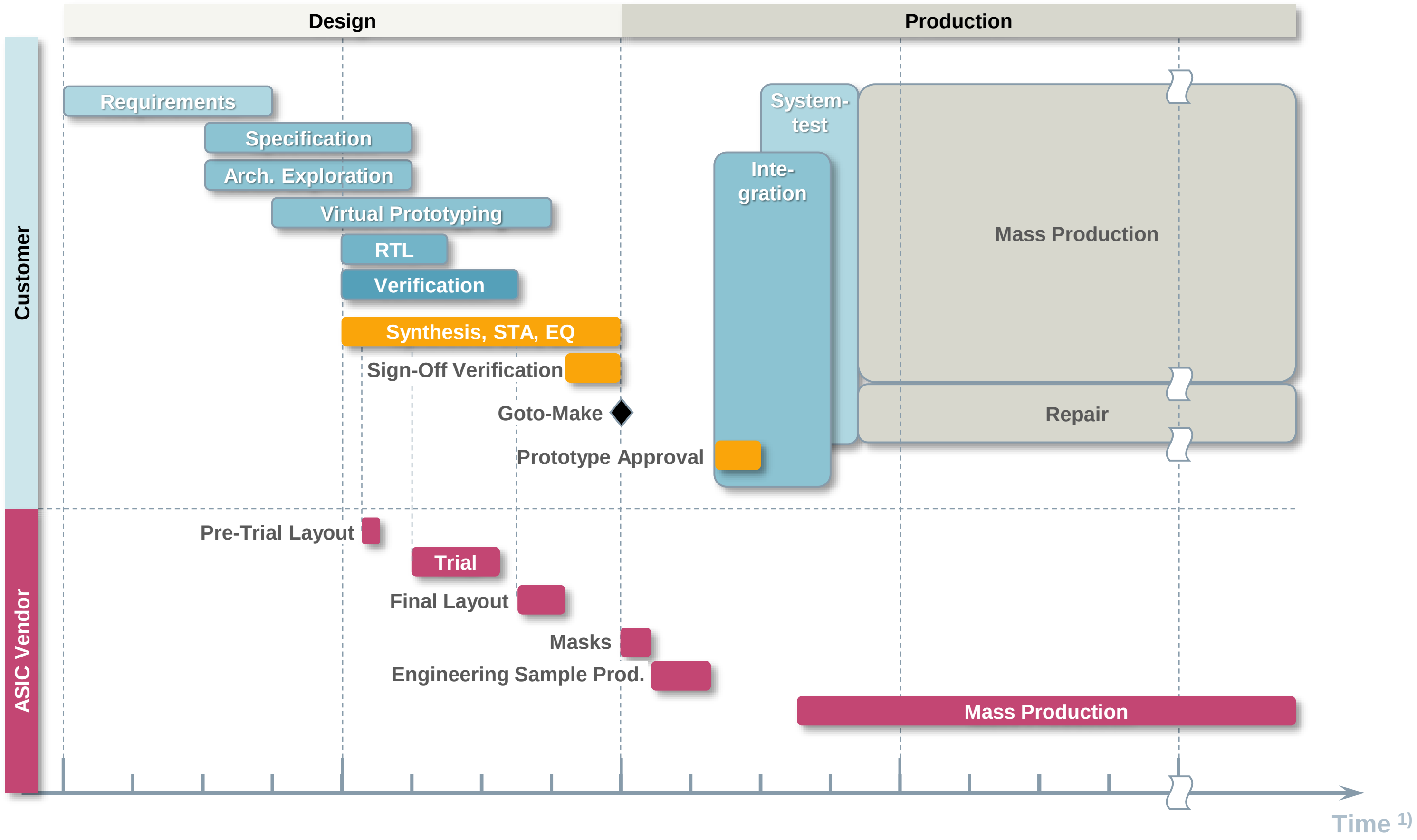
Requirements
Architecture
ASIC/FPGA: Design Implementation Verification Backend
FW/SW: Design Implementation Test
PCB: Design Implementation Verification Sample
Mechanics: Design Implementation Verification Sample
Integration
Systemtest
Mass Production
Repair
Recycling
Time 1)

1) Time scale varies significantly from product to product

ASIC design flow for a netlist handover business model

SIEMENS
Ingenuity for life

Design phases and major milestones



1) Time scale varies significantly from product to product

Digital Chip Design

SIEMENS
Ingenuity for life

Outline of the lecture

Drivers for industrial ASIC design

ASIC design is the critical track in product design

Abstract view on ASIC design – Collaboration models with ASIC vendors

ASIC design flow in detail

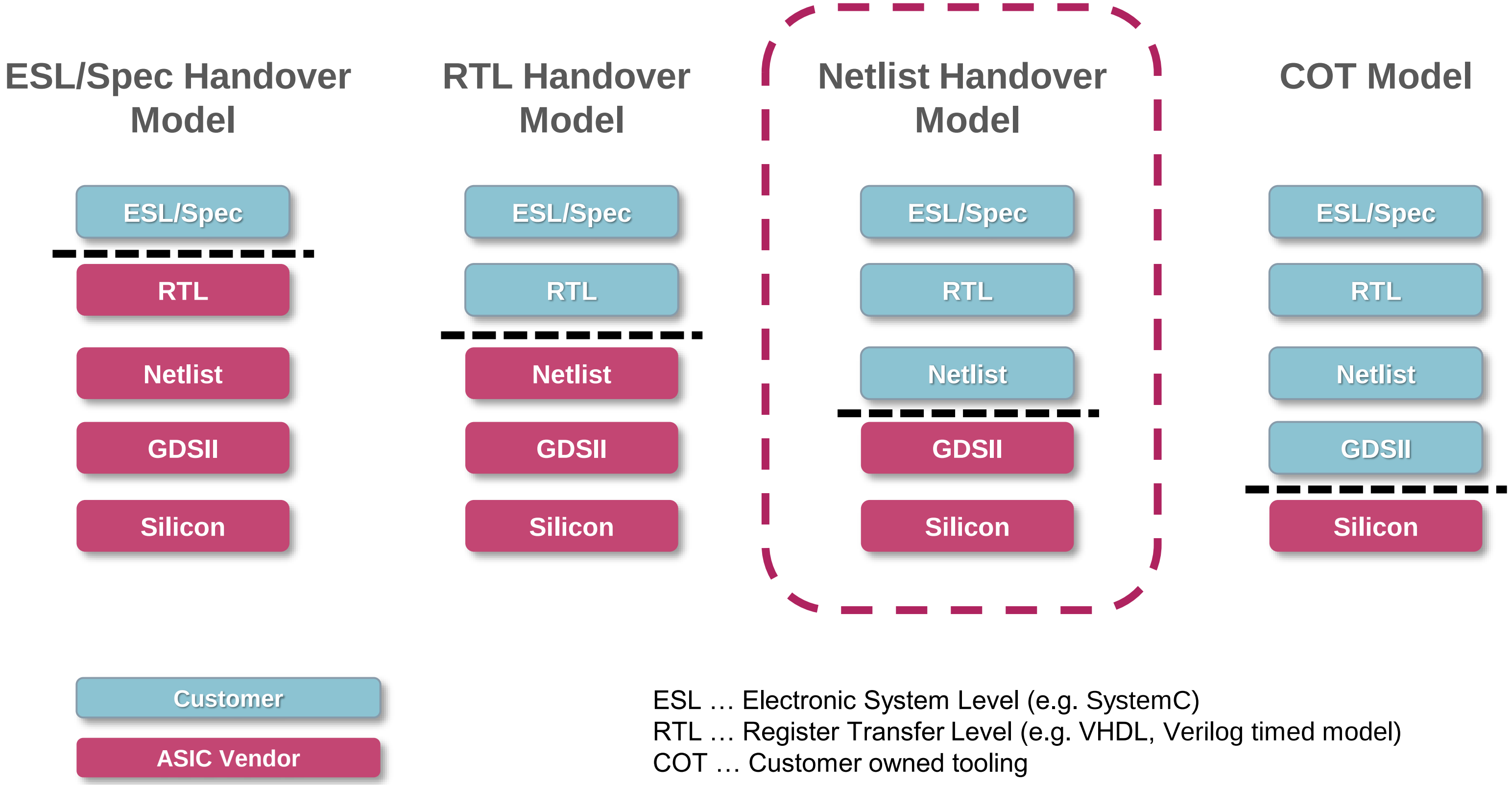
- Requirements and specification
- ASIC Vendor selection
- The functional model
- The netlist
- Digital layout
- Production

Collaboration models define the worksplit between ASIC vendor and customer

SIEMENS
Ingenuity for life

Four collaboration models are widely used in the industry

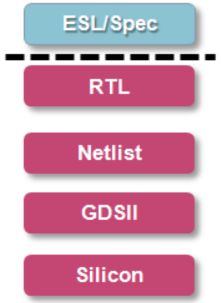
- **Expertise** on customer side in digital design and **IP protection** are driving decision
- Netlist handover is the most widespread collaboration model



ESL … Electronic System Level (e.g. SystemC)
RTL … Register Transfer Level (e.g. VHDL, Verilog timed model)
COT … Customer owned tooling

ESL/Specification Handover Model
How to benefit from ASIC w/o digital design expertise

Customer specifies only functionality in written or executable format



Pros

Customer

- **No digital design expertise** required
- No **costly** Electronic Design Automation **infrastructure** required
- No continuous **Silicon technology expertise** required

ASIC vendor

- **Short** in-house **loops** from digital layout back to RTL design
- **Deep insight** into customers application domain
- Cross domain and cross customer **synergies**
- **Full control** of RTL and layout

Cons

Customer

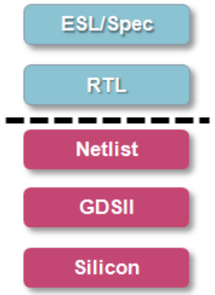
- Leveraging full **technology capabilities** is limited
- Risk for **unintended functionality** due to weaknesses in specification
- **Iterations** with vendor to refine specification

ASIC vendor

- **Expertise in application domain** required on top of technology expertise
- Limited expertise at customer side complicates **communication**
- Complex **contracting** to cover responsibilities

RTL Handover Model
Customer has full control of functionality

Customer specifies functionality cycle accurate



ESL/Spec

RTL

Netlist

GDSII

Silicon

Pros

Customer

- No risk for unintended functionality due to **cycle accurate model**
- **Short in-house iterations** for alignment of functionality
- Quite **simple** contracting due to complete specification

ASIC vendor

- **Short in-house loops** from digital layout back to synthesis
- **Deep insight** into customers application domain
- **Full control** of synthesis and layout

Cons

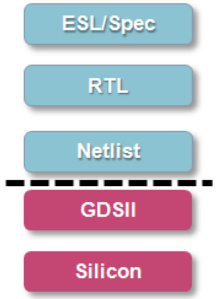
Customer

- Leveraging full **technology capabilities** is limited
- Specification of **design constraints** suffers from limited technology expertise
- Full **IP disclosure**

ASIC vendor

- Time consuming **iterations with customer** for RTL optimizations to support layout

Netlist Handover Model
Customer gets early insight into vendor's technology



ESL/Spec

RTL

Netlist

GDSII

Silicon

Insight into technology enables customer to drive optimizations close to the limits

Pros

Customer

- **Synthesis forecasts** results from layout with high confidence level
- **Optimizations** for performance and power close to the technology limits

ASIC vendor

- High level of expertise at customer side eases the **communication**
- Almost **no iterations** with customer due to high quality of correlation of synthesis with layout

Cons

Customer

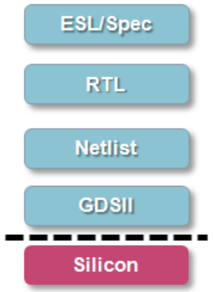
- Synthesis and technology **expertise** required
- **Maintaining expertise** requires continuous work with relevant technology nodes

ASIC vendor

- **Effort** for technical support at customer side
- Very limited **insight** in functionality which is required or al least beneficial for optimizations

COT Model
ASIC vendor just provides technology

SIEMENS
Ingenuity for life


ESL/Spec
RTL
Netlist
GDSII
Silicon

The customer takes full responsibility for the ASIC

Pros

Customer

- **Full control** of everything
- No external communication and **full IP protection**

ASIC vendor

- Very limited **effort** for customer support
- **Simple contracting** due to manufacturing-only responsibility

Cons

Customer

- Complete digital design **expertise** incl. DfT and layout required
- **Expertise** for digital technology node required

ASIC vendor

- **?**

Digital Chip Design

Outline of the lecture

Drivers for industrial ASIC design

ASIC design is the critical track in product design

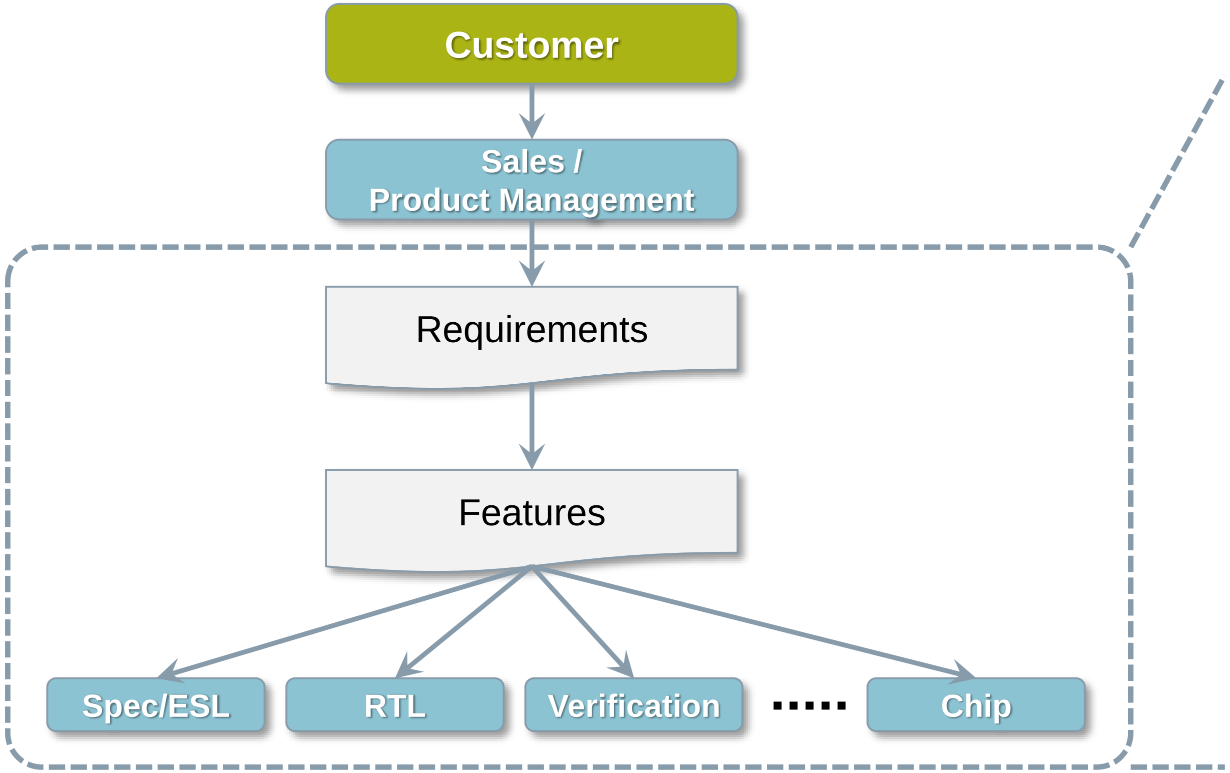
Abstract view on ASIC design – Collaboration models with ASIC vendors

ASIC design flow in detail

- Requirements and specification
- ASIC Vendor selection
- The functional model
- The netlist
- Digital layout
- Production

Completeness and correctness of requirements are key for a successful ASIC

Requirements Engineering is the starting point for any ASIC design



Traceability

- Forward and backwards
- Through all artifacts

Coverage

- Are all requirements/features considered?
- Are the requirements/features considered in the relevant artifacts?

Traceability and coverage improves quality

Certification (e.g. safety) requires traceability and coverage

Architecture Exploration enables identification of most appropriate/optimized architecture

Ever increasing design complexity requires novel modeling technologies for exploration

„Excel + Block Diagram“

- **Block diagrams** are the first and most abstract representation of an ASIC
- The **architecture standards** (e.g. AMBA) and the related **design IP** market define the granularity of block diagrams
- Bus interconnects and **networks-on-chip** integrate the various blocks
- **Excel-based** assessments and evaluations model aspects of **performance** (e.g. throughput, latency, power, area)
- **Architecture exploration** (e.g. HW-SW worksplit) is **a manual task** and therefore time-consuming
- There is no executable specification of the architecture

Abstract modeling

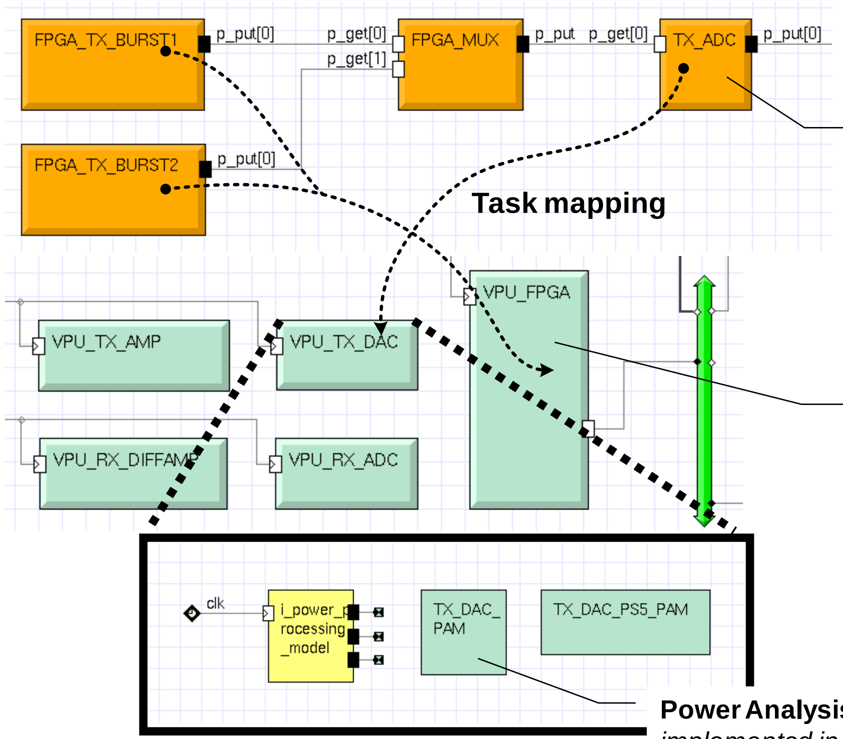
- **Modeling frameworks** enable to capture block diagrams
- Blocks are **scalable** (e.g. performance, capabilities)
- Automated **sweeping of parameters** enable to identify optimized setup
- **What-if** analysis
- Understanding **side-effects and limitations** of architecture
- **Models** for standard interfaces and functionalities available

Architecture Exploration
Application task graph mapping approach

SIEMENS
Ingenuity for life

A commercial framework for abstract modeling to work around SW availability

- Generic **scalable** building blocks to model HW
- A **task graph** abstracts the SW/algorithm
- Extension to also model **power** dissipation is available



Task Graph Design

Task
start time offset
processing delay
wait time
memory access

HW Architecture Design

Virtual Processing Unit
is an abstract functional model

Task Graph Mapping

Power Modeling

Power Analysis Models
implemented in Tcl

Make or buy
Architecture standards support harmonization

SIEMENS
Ingenuity for life

AMBA – Advanced Microcontroller Bus Architecture is the dominating standard

- A significant share of ASICs are SoCs – **System on Chips**
- SoCs combine harmonized **standard processing** with **proprietary functionalities**

- Harmonized standard functionalities are subject of **design IPs**
  - The **embedded processor** landscape consolidated a lot with ARM having a strong position
  - RISC-V is emerging
  - Bus system infrastructure and network-on-chip IPs are of critical importance due to ever increasing complexity of SoCs
  - **Standard interfaces** increase productivity and reduce risk

- A **strategic design IP selection** helps to
  - minimize efforts for ASIC design
  - minimize efforts for SW design
  - minimize risks for re-design
  - optimize the design migration

The completeness and correctness of the specification of features is critical

Specification of features

Prose

- Written format has lots of **limitations**
  - Level of detail
  - Completeness
  - Correctness
  - Verification/review

- Widely used since the early days of ASIC
- **Copy-paste** w/o limits
- **1000s of pages** for complex SoCs
- Only manual **consistency** between document and subsequent model
- Issues are identified in late verification phase and cause **costly iterations**

Executable specification

- Supports **verification** of use-cases
- **Model-driven engineering** in combination with ESL (e.g. SystemC) are a very powerful and systematic approach
- There is no degree of freedom in terms of **interpretation** of written specification
- **Completeness** still not solved
- Verification of use-cases address **correctness**
- Uniform format for **design IPs** is still open
- **Level of details** (e.g. untimed vs. timed models) is still open

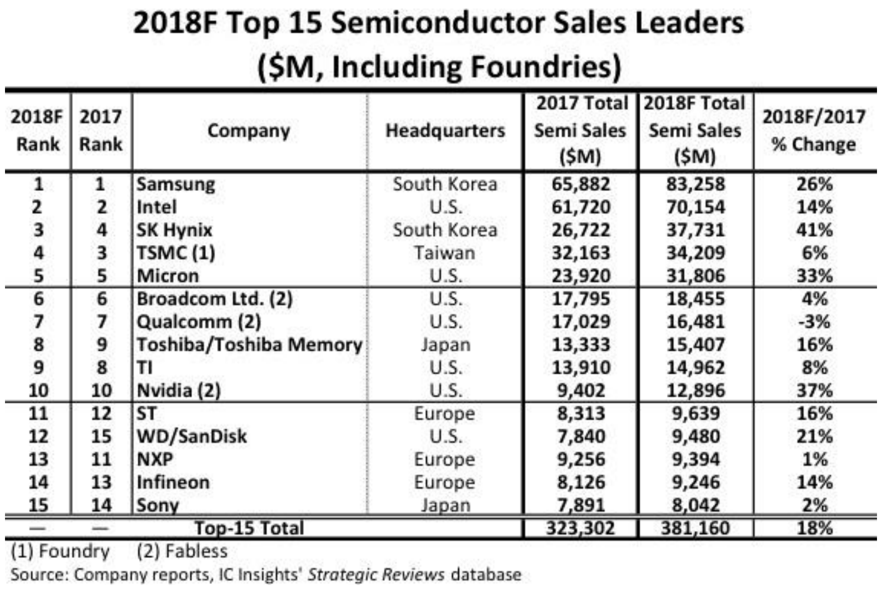
ASIC vendor selection
Balancing technical and commercial arguments

SIEMENS
Ingenuity for life

The ASIC vendor landscape consolidated over the past years

- Only a **few vendors** can afford the development of new technology nodes
- **Major players** in technology are Intel, Samsung, TSMC, Globalfoundries
- **International Technology Roadmap for Semiconductor** (ITRS) documents market and identifies trends
- Smaller technology nodes result in significantly **higher costs** for masks and higher NRE costs (non-recurring engineering costs)

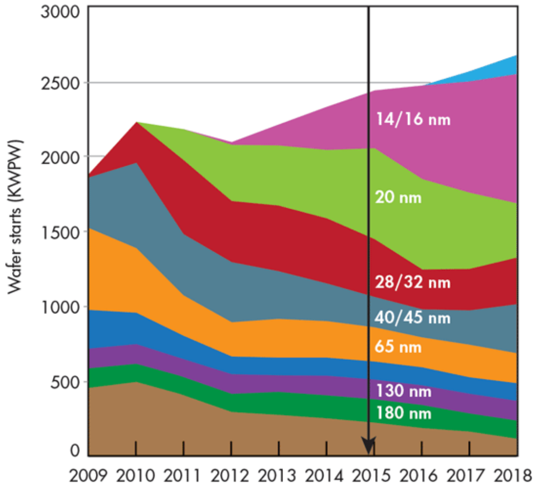
- **Main criteria** are
  - Performance, power dissipation
  - NRE costs, costs of chips
  - Lifetime, reliability
  - Etc.

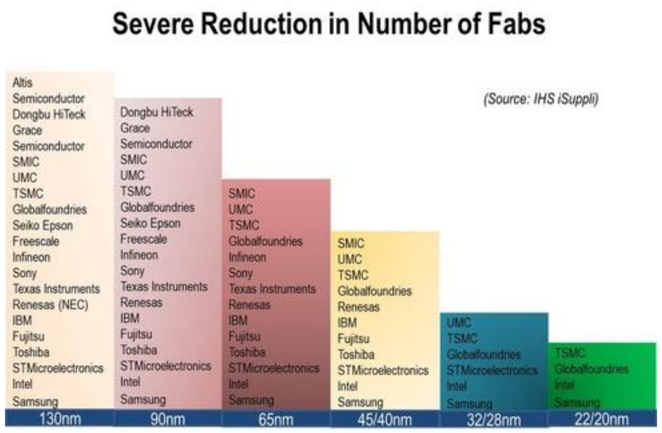


2018F Top 15 Semiconductor Sales Leaders ($M, Including Foundries)

2018F Rank	2017 Rank	Company	Headquarters	2017 Total Semi Sales ($M)	2018F Total Semi Sales ($M)	2018F/2017 % Change
1	1	Samsung	South Korea	65,882	83,258	26%
2	2	Intel	U.S.	61,720	70,154	14%
3	4	SK Hynix	South Korea	26,722	37,731	41%
4	3	TSMC (1)	Taiwan	32,163	34,209	6%
5	5	Micron	U.S.	23,920	31,806	33%
6	6	Broadcom Ltd. (2)	U.S.	17,795	18,455	4%
7	7	Qualcomm (2)	U.S.	17,029	16,481	-3%
8	9	Toshiba/Toshiba Memory	Japan	13,333	15,407	16%
9	8	TI	U.S.	13,910	14,962	8%
10	10	Nvidia (2)	U.S.	9,402	12,896	37%
11	12	ST	Europe	8,313	9,639	16%
12	15	WD/SanDisk	U.S.	7,840	9,480	21%
13	11	NXP	Europe	9,256	9,394	1%
14	13	Infineon	Europe	8,126	9,246	14%
15	14	Sony	Japan	7,891	8,042	2%
—	—	Top-15 Total		323,302	381,160	18%

(1) Foundry (2) Fabless
Source: Company reports, IC Insights' *Strategic Reviews* database





Source: VLSI

ASIC vendor collaboration model
Netlist handover is most wide-spread

Key topics of the collaboration model

- Milestones, **Schedule**
  - Customer: Pre-trial, trial and final netlist handover; Goto-make, Release for mass production
  - Vendor: Library available, Floorplan, Layout data, Engineering samples, Mass production start
- **Responsibilities**
  - Vendor: FAE, Libraries, specific tools, IPs, DfT, digital layout, equivalence checking, STA, masks, engineering samples, mass production
  - Customer: Pre-trial, trial and final netlist, design constraints, post layout verification, approvals
- Tool **flow and libraries**
  - Tools and specific versions need to be aligned
  - Library updates need to be aligned
- **Reviews**
  - Starting with a kick-off, each phase closes with a review
  - All handovers have acceptance checks
- **IPs**
  - Is there any technology-related IP required (e.g. IO buffers, SERDES, ADC)?
  - Are there test chips required for these IPs?
  - Which models are available (e.g. simulation, synthesis, STA, equivalence check)

ASIC vendor collaboration model
Netlist handover is most wide-spread

SIEMENS
Ingenuity for life

Key topics of the collaboration model

- **DfT**
  - Are there prerequisites for the netlists? Is there any DRC available to check upfront?
  - Is there any DfT technique the customer wants to have access to (e.g. Boundary Scan)?
  - Who does the STA?
  - Will customer get all tester pattern?
  - What is the (required) coverage for each fault model?
- **Test**
  - Which tests are done at wafer level, which at package level?
  - Which tests are already available for ES (Engineering Samples)?
- **Packaging**
  - Is packaging outsourced?
  - Is there a second source?
- **Bug-fixing**
  - What are the costs and what is the schedule for a metal fix and for a full re-spin?
  - Is there a fibbing service available?
  - Who takes the costs for which issues?

Executable specification supports verification of use-case scenarios in early design phases

Late architectural changes are costly and time consuming

Traditional **prose specification**-based flows

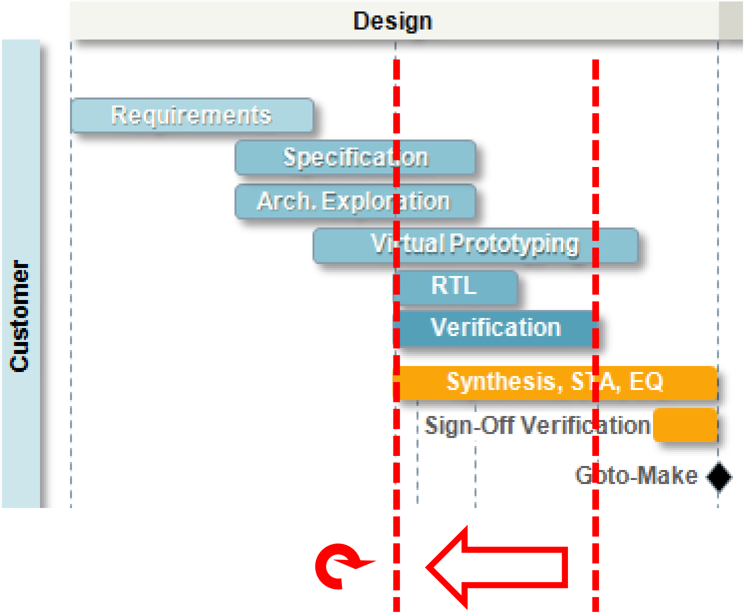
- The verification of the architecture is very late in the RTL verification phase.
- Identified issues often require architectural changes and re-work of significant parts of RTL and verification.

→ Costly and time-consuming loops

Executable specification

- Use-cases are verified early with the architectural model
- For complex designs prose specification have thousands of pages – no way to do comprehensive reviews

→ very short loops at architecture model level before subsequent RTL implementation

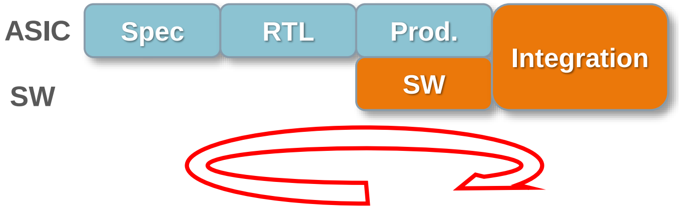


Virtual prototyping for SW design reduces time-to-market significantly

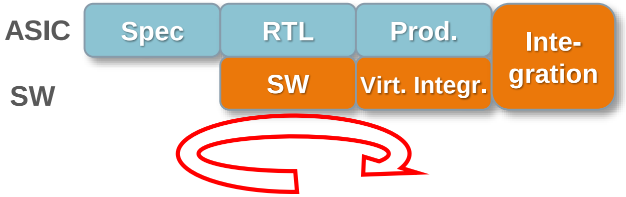
SIEMENS
Ingenuity for life

SW integration requires either HW or a model of the HW for testing

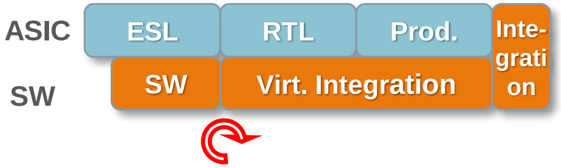
- In traditional design flows **HW-SW-integration** starts with first **engineering samples**
- Fixing issues cause **long and costly iterations**



- Starting **HW-SW-integration** based on **RTL model** make iterations more local
- Subsequent HW-SW-integration on ES is shortened significantly
- RTL **simulation performance** is limiting factor



- Starting HW-SW-integration based on ESL model make **iterations** even more **local**
- Subsequent HW-SW-integration on ES was shortened dramatically
- ESL **simulation performance** is appropriate for HW-SW-integration



The functional model

A detailed executable model based on the specification for automated transformation to gates

ESL model

- **Untimed model** optimized to simulation performance; timing is part of additional design constraints
- **Timed model** is closer to RTL but still does not implement all timing details
- **High Level Synthesis** (HLS) transforms ESL model into gates (or RTL); lot of additional guidance and design constraints are required
- **Formal sequential logic equivalence** checkers prove equivalence of ESL and gates

RTL

- **Clock cycle accurate** model implements complete timing behavior
- **Simulation performance** at least one order of magnitude slower than ESL
- **Synthesis** does deterministic transformation into gates; only a few additional design constraints (e.g. clock frequency) are required
- **Formal logic equivalence** checkers prove cycle-by-cycle equivalence of RTL and gates

Verification of ESL and RTL models vs. **requirements/specification** is major challenge!

Verification of functional models

Proving the correctness of the functional model vs. the requirements/specification

- Main goal is to ensure that the **ASIC fulfills** the **requirements**

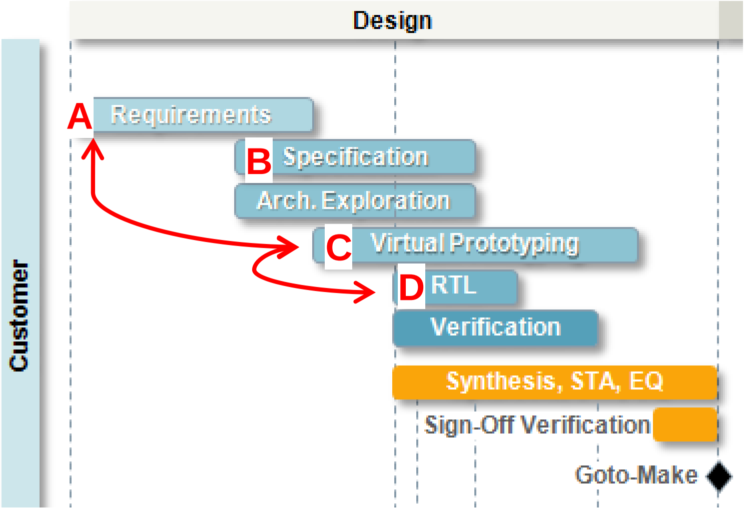
→Functional RTL model D ↔ requirements A

- **Virtual prototype** C used for HW-SW integration **shall behave like the ASIC**

→Virtual prototype C ↔ functional RTL model D

Set-up a **verification environment** which

- verifies the features derived from A (specified in B)
- verifies first the Virtual Prototype C
- supports refinement of the environment to also verify the RTL model D



Verification of functional models

There are various verification methodologies in place

Simulation based verification drives stimuli and checks results

- **Directed**: Simple approach to drive a defined set of stimuli; appropriate for very simple DUTs
- **Random**: Brute force approach to drive a DUT with random data and check the results with a reference model
- **Constraint random**: Guides the random stimuli generation with constraints; much more efficient than pure random
- **Metric driven constraint random**: Metrics (coverage items) drive stimuli generation; state-of-the-art for complex DUTs

Formal and hybrid methodologies

- **Model checking** with native frontend (PSL, etc.)
- Model checking with application specific frontend (e.g. check of toplevel wiring, clock domain crossing check
- **Hybrid** methodologies simulate a certain number of clock cycles to “set-up” the DUT and then continue with formal methods

Verification IP

- Commercial IPs for standard interfaces (e.g. certified compliance tests)
- In-house verification IPs which result from silicon proven implementations

Verification Management

Verification is most time consuming task and requires a effective management

Verification Planning

- Detailed technical planning of **how** to verify which feature in an **automated** way
- Detailed project management to keep verification from **critical path**
- Ensure the appropriate **level of verification** for pre-trial, trial and final netlist handover

Execution of verification

- **Communication**: Ensure a targeted information of the status of bugs for affected bodies
- **Regression**: Automated execution of verification and related reporting
- **Balancing of resource**: Management of license and server usage

Controlling and reporting

- **Timely** availability of **actual status** of verification
  - Feature coverage achieved vs. plan
  - Status of bugs over time
- **Automated** frequent reports to document the progress and history

Quality of verification

A high quality verification minimizes the risk of costly re-designs

Coverage

- **Code coverage**: mandatory but not sufficient
- **Functional coverage**: Identification of all features to be covered is crucial
- Coverage is a strong indicator for verification progress

Reference models

- **Verification IP** for standard interfaces
- Reference models based on **design pattern** in dedicated verification languages (e.g. SV, e)

Mutation based assessment of verification environments

- Methodology to check if verification environment checks the same features implemented in the DUT
- Tool introduces mutations in source code of DUT and checks if verification environment is able to trigger the mutation and to identify propagated misbehavior
- No way to uncover missing features or badly implemented features if behavior of DUT and verification environment is identical

Synthesis
The transformation of functional models into gates

HLS will substitute RTL in the upcoming future like RTL synthesis did it two decades ago

Starting from ESL → HLS

- Input: **ESL model**
- Output: **RTL**
- ESL model implements a large degree of freedom compared to clock cycle accurate RTL
- **Additional input required** to guide HLS to
  - Optimize for performance
  - Optimize for resources and power
- **Formal logic equivalence** check is tricky (SLEC)
- Excellent flow for **short design cycles**
- There is still some room for further **optimizations**
- **Subsequent RTL synthesis** required
- **Risk**: RTL is slightly different from ESL!

Starting from RTL → Synthesis

- Input: **RTL model**
- Output: **Gatelevel netlist** of library primitives
- RTL is **clock cycle accurate** → netlist exactly matches verified RTL
- RTL synthesis has long history and is a very **mature** technology
- **Formal equivalence** check proves correctness of transformation completely
- **Correlation** of timing and area in synthesis with final layout results is at very high level
- **Design constraints** are consistently used for subsequent digital layout steps

Synthesis
Starting from RTL

Two EDA vendors are dominating the market – Synopsys and Cadence

Tool chains

- **Synopsys** Design Compiler is tightly integrated with IC Compiler (digital Place&Route)
- **Cadence** Genus (former RTL Compiler) is tightly integrated with Innovus (former Encounter)
- **Data format standards** ensure interoperability of tool chains to a large extent

Synthesis strategy

- Several **hundreds of commands and variable** to guide synthesis tools
- Threshold voltage **library mixing** adds another degree of flexibility/freedom
- Wisdoms like “It’s easy to add gates but very hard to get rid of them…”

Constraints

- Specify **timing, power and area requirements**
- **Consistently** used starting from synthesis downstreams
- **Completeness** of constraints is crucial
- Consistency from subchip level to toplevel is critical for one-pass flow

Synthesis
Starting from RTL

Close alignment of flow with ASIC vendor is a key success factor

Interaction with ASIC vendor

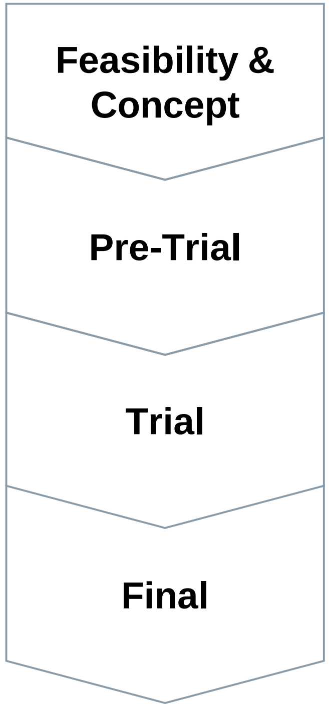
- Tool **versions** and library versions
- Targeted threshold voltage **library mix**
- **Optimizations** in synthesis not supported downstreams in digital layout
- Floorplan
- **Design constraint** “architecture”
- Hierarchical layout flow
- **Modes and corners** for STA (e.g. functional only)
- Pinning

DfT

- Pre-requisites and DRC at RTL and synthesis netlist level
- DfT interfaces at subchip level and toplevel
- Customer access to DfT infrastructure (e.g. RAM BIST)

A 3-step flow for 40nm and beyond

A 3-step design flow is absolutely required for physical synthesis



- Assessment of technology nodes
- Extract timing, power and area characteristics
- Evaluate trade-offs between power and functionality

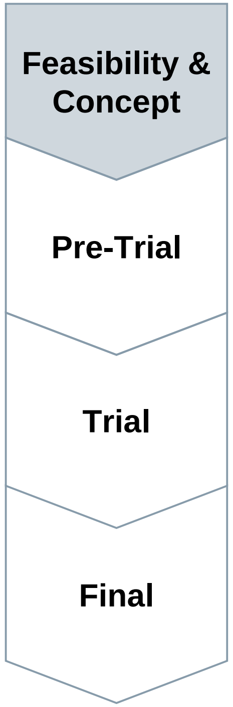
- Initial netlists (85%) for **subchip floorplaning** and design analysis
- Initial toplevel netlist for **toplevel floorplaning**

- Trial netlists (95%) for **complete layout runs**
- Assessment of additional optimization potentials

- Final netlists (100%) for complete layout runs
- Full **post layout verification**

A 3-step flow for 40nm and beyond
Feasibility and concept phase

A detailed feasibility phase is essential for a RFQ with high confidence level



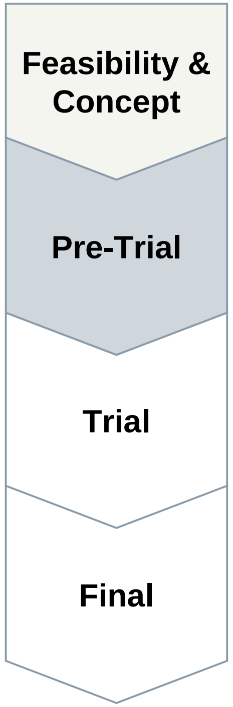
- Select representative design modules
- **Pipe-clean** the synthesis setup
- A series of synthesis runs for
  - Variation of frequency to assess timing performance and technology “sweet-spot”
  - Impact of synthesis strategy on performance vs. power
- **Trial layout** to assess layout gatecount overhead

- **Estimation/calculation power** and gatecount of several design scenarios

- In parallel
  - Architecture exploration and design specification
  - Virtual prototyping
  - RTL implementation and verification

A 3-step flow for 40nm and beyond
The pre-trial layout phase

The key success factor for physical synthesis is the floorplan



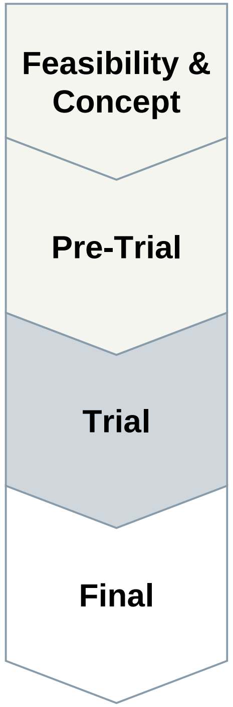
- **Early netlist** for each subchip for **floorplanning**, initial layout and **test insertion** steps
  - >85% of gatecount represented
  - Main connectivity implemented
  - All clocks and resets implemented
  - All RAMs and vendor IPs implemented

- Early netlist for toplevel with full connectivity and no glue logic
  - **IO locations**
  - Floorplanning based on inter-subchip connectivity
  - Considering the main data paths but also the control path
  - Considering voltage domains

- **RTL optimizations** to overcome layout issues
- **Design partitioning** requires several iteration of subchips and toplevel netlists

A 3-step flow for 40nm and beyond
The trial layout phase

All layout tasks have to be passed and automated



Feasibility & Concept

Pre-Trial

Trial

Final

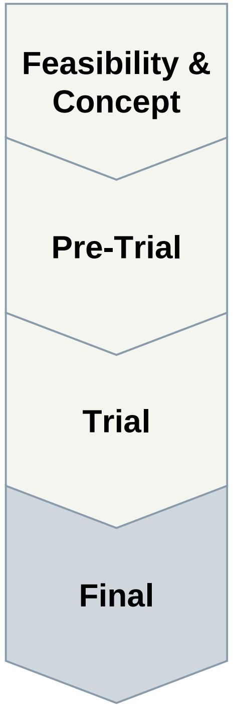
- **Trial netlist** for each subchip
  - >95% of gatecount implemented
  - Almost **complete connectivity** implemented
  - All clocks, resets, RAMs and vendor IPs implemented
  - **Verification** of basic functionality passed

- Trial toplevel netlist with full connectivity and all IOs
  - Fine-tuning of floorplan
  - Lead-frame and **package design**
  - UPF defines voltage domains

- All layout steps have to be **passed and automated** for short final layout phase

A 3-step flow for 40nm and beyond
The final layout phase

Optimized turnaround times for all layout tasks down to mask DRC



Feasibility & Concept

Pre-Trial

Trial

Final

- **Final netlist** for each subchip
  - **Verification closed** and DRC-clean netlist
  - Complete and clean design constraints

- Final toplevel netlist with **complete and clean design constraints**
  - Flat and hierarchical design constraints
  - UPF defines voltage domains

- All layout steps have to be passed
- Timing closure for all corners
- All LVS and final DRC tasks
- Final review with layout engineer

Goto Make
The ultimate milestone to release mask production

This milestone is the gate between ASIC design and ASIC production

- Mask sets are in the **multi-million USD** price range

- The **quality gate** covers **all involved parties**
  - Has complete verification passed successfully?
  - Has synthesis, STA and EQ passed successfully?
  - Has DfT insertion, layout, DRC, LVS and post-layout verification passed successfully?
  - Has configuration management verified the database used and are all results labeled?
  - Is pinning approved by packaging and PCB?

- Goto Make is also an invoicing milestone

Mask production, chip frontend and backend

Chip production typically involves multiple sites

- Large ASIC vendors have their in-house **mask shops** or they rely on 3rd parties

- Chip production is split into
  - **Frontend**: All steps at wafer level (e.g. edging, diffusion, wafer-level testing)
  - **Backend**: All steps at package level (e.g. packaging, package level testing)

- Latest technology nodes require 50-60 masks
- Chip production takes at least one day per mask in average
- In many cases diffusion, packaging and testing are at different sites
- There are about **90 days from Goto Make till Engineering Samples (ES)** available
- Typically **test programs** for ES are not fully up and running
- Higher risk for buggy silicon

Prototype approval – release for mass production

Exhaustive testing of Engineering Samples before start of mass production

Customer

- **ES** from typical, hot and cold process corners are **examined by customer**
- Systematic test of all features in all applications available
- Stress-testing
- Electrical characterization
- Package reliability assessment
- ESD assessment

ASIC vendor

- Bring-up of **all test programs**
- Provide reports for all test programs
- Characterize ASICs and assess yield
- Prepare for mass production

Final quality gate releases mass production

Synthesis strategy
Some topics to consider for latest technology nodes

- **Physical congestion-aware** synthesis (e.g. Synopsys Design Compiler Topographical/Graphical)
  - w/o floorplan for pre-trial netlist
  - w/ floorplan for trial and final netlist
- SMSC - **Single Mode Single Corner**
- Scan flop insertion
- Optimized **overconstraining** to minimize netlist
  - Correlation of STA (e.g. PT-SI) of subchip vs. toplevel
  - Correlation of layout tool (e.g. ICC) vs. STA tool (e.g. PT-SI)
  - Correlation of synthesis tool (e.g. DC-T) vs. layout tool (e.g. ICC)
  - Clock source uncertainty
- **Lib selection** – Density vs. Performance, standard VT and high VT enabled
- **Leakage** – Optimization enabled for synthesis
- **Clock gating** – enabled
- Handover formats – Verilog netlist + SDC, Tcl constraints, DDC (generic and technology mapped)
- **Optimizations** – remove unloaded sequentials, constant propagation, etc.
- Starting with weak optimizations for pre-trial down to full optimization for final netlist

Pre-layout analysis
Steps before netlist handover to ASIC vendor

- Formal equivalence check
  - RTL vs. netlist is in many cases straight forward
  - **Aggressive optimizations** in synthesis can cause problems
  - Equivalence checking tool and synthesis tool from **different EDA vendors** does not work
  - Before contacting EDA vendors try latest tool release

- Design Rule Checking
  - Many ASIC vendors use DRC as netlist **handover acceptance check**
  - Run early trials to not face surprises at the milestone date
  - Tools generate very useful **design statistics**

- STA
  - **Multi-Mode-Setup**: One setup for each functional mode
  - Generate SDC from pre-layout-STA setup
  - Handover both PT-Tcl and SDC to vendor

Typically DfT is a service of the ASIC vendors

The TAP controller is the central access point

Boundary Scan (IEEE 1149)

- TAP (**Test Access Port**) controller controls whole DfT infrastructure
- Used for board-level connectivity test
- BSDL files is the specification of the boundary scan infrastructure
- TAP is also access point for **debug infrastructure** for on-chip processor cores
- **Access control** for TAP for security reasons
- **Device ID** for traceability of production

Scan

- Covers several **fault models** (e.g. stuck-at, bridging, small delay defects)
- On-chip pattern **generation/expansion and compression**
- At-speed tests
- Clock domain crossing
- Pattern generation and required **coverage**
- **Simulation** for shift and capture phase
- **Diagnosis** for failing patterns

Typically DfT is a service of the ASIC vendors

The TAP controller is the central access point

RAM-BIST

- Algorithms depend on Silicon technology node
- ASIC typically have lots (100+) RAMs – **runtime** of RAM-BIST is an issue
- **Hierarchical infrastructure**: local controller for each RAM clustered to independent groups
- One of the most important test for manufacturing
- **Customer access** to RAM-BIST limited (e.g. start, fail-pass information)

Special **requirements for IPs**

- PLLs: Lock status, frequency monitoring
- SERDES: Lock status and frequency monitoring of internal PLLs
- ADC/DAC: Calibration
- Process monitoring characterize the Silicon process status
- …

Constraining and Post Layout STA

SIEMENS
Ingenuity for life

Constraints are slightly different for different stages of the design flow

Constraining

- **Multi mode**
  - One constraint set for each functional and test mode
  - Merge modes if possible to reduce analysis run-time
- Implement **over-constraining** as a global variable to be easily adjusted
- Generate **IO constraining** from easy-to-review format (e.g. Excel)

STA setup

- **Multi Corner**: Typically 10+ library corners need to be analyzed
- **Clock tree**: Clock tree insertion delay has to be considered for IO constraining

Pre-layout vs. post-layout analysis

- **Clock tree**
  - Pre-layout there are no clock trees inserted
  - Post-layout there is a certain distribution of insertion delay clock tree leaf by leaf
- **Cross talk**: Signal integrity is calculated precisely

There are many steps in the digital layout process

The detailed sequence strongly depends on the Silicon technology node and level of optimization

Example for a Synopsys IC Compiler based flow

1. Setup
2. DFT Insertion
3. Formality
4. Floorplan
5. Initial Placement, place_opt
6. Optimize Placement, syn_opt
7. ICC **timing** for Setup
8. CTS, clock_opt
9. ICC timing for Setup
10. Post place_opt
11. ICC Timing Reports
12. clock route
13. Initial Route (global + detail)
14. ICC **timing**
15. route_opt
16. ICC Timing
17. Filling
18. ICC **timing**
19. RC-XT
20. PT-SI
21. Hold Fixing
22. Leakage optimization
23. Power integrity
24. DRC fixing, LVS in ICC
25. LVS and DRC in sign-off tool
26. Final RC-XT and PT-SI
27. Final Formality

Low Power design requires Multi Voltage Domain design flows

There are standards in place to specify the power intent for the design

Multi Voltage Domains result from

- **Power down** of parts of the chip to even save leakage power
- **Dynamic voltage scaling** to minimize power dissipation

Standards in place

- UPF – Unified Power Format: Driven by Synopsys
- CPF – Common Power Format: Driven by Cadence

Impact on design process

- RTL: Logic simulators **infer** level-shifters and isolation cells defined in UPF/CPF
- Synthesis: **Inserts** level-shifters and isolation cells defined in UPF/CPF
- Formal Logic Equivalence Check, MVDRC and MV simulation for verification

Functional Safety defines additional requirements also for ASICs

Functional Safety has impact on function, implementation and design process

In many application domains (e.g. mobility, healthcare, industrial machinery) safety is enforced by **regulations**

- Industrial IEC61508, Automotive ISO26262, etc.
- Standards impact **function, implementation and design process**

Some examples

- If technology node is qualified for less than 3 years a additional timing margin of x% is required
- Full traceability of requirements throughout the design process
- Hierarchical design approach
- Verification of each model transformation (e.g. synthesis with formal equivalence check)
- Relevant history of tools used
- Checklists defined in standards
- Independent power supplies for independent functions
- Clock monitoring to identify common cause issues
- Test functions to increase diagnostic coverage

ASIC vs. FPGA
Technology and costs drive the decision

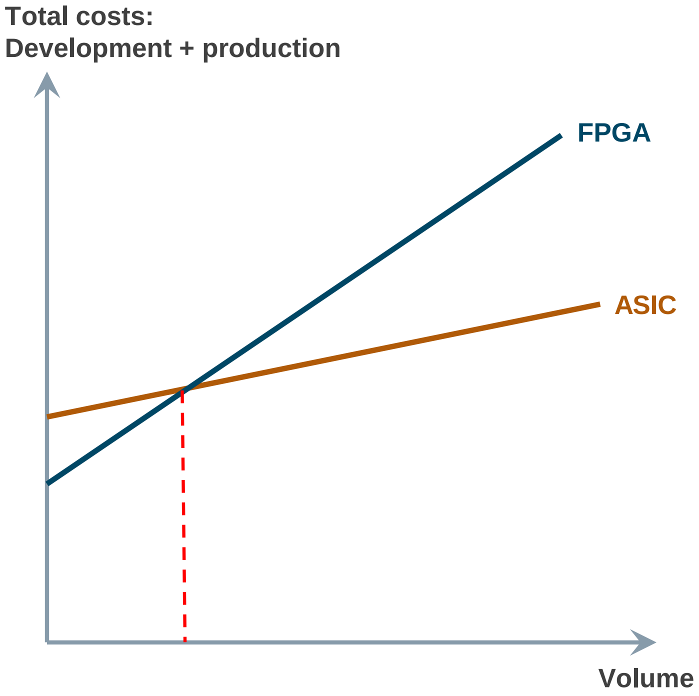
Commonalities and differences between ASIC and FPGA

FPGAs are generic types of ASICs

- The **generic cell/routing infrastructure** of FPGAs requires approximately 10x transistors vs. a custom ASIC implementation
- The overhead results in additional **costs** per devices
- The overhead results in additional **power** dissipation

The design process for FPGA is significantly cheaper and shorter

- The FPGA synthesis and **bitfile generation** is much less effort than three digital layout iterations for ASIC
- For FPGA there is **no mask generation** required



Mixed Signal ASIC integration

There is a well established design process for analogue/mixed signal design

Motivation for mixed signal ASICs

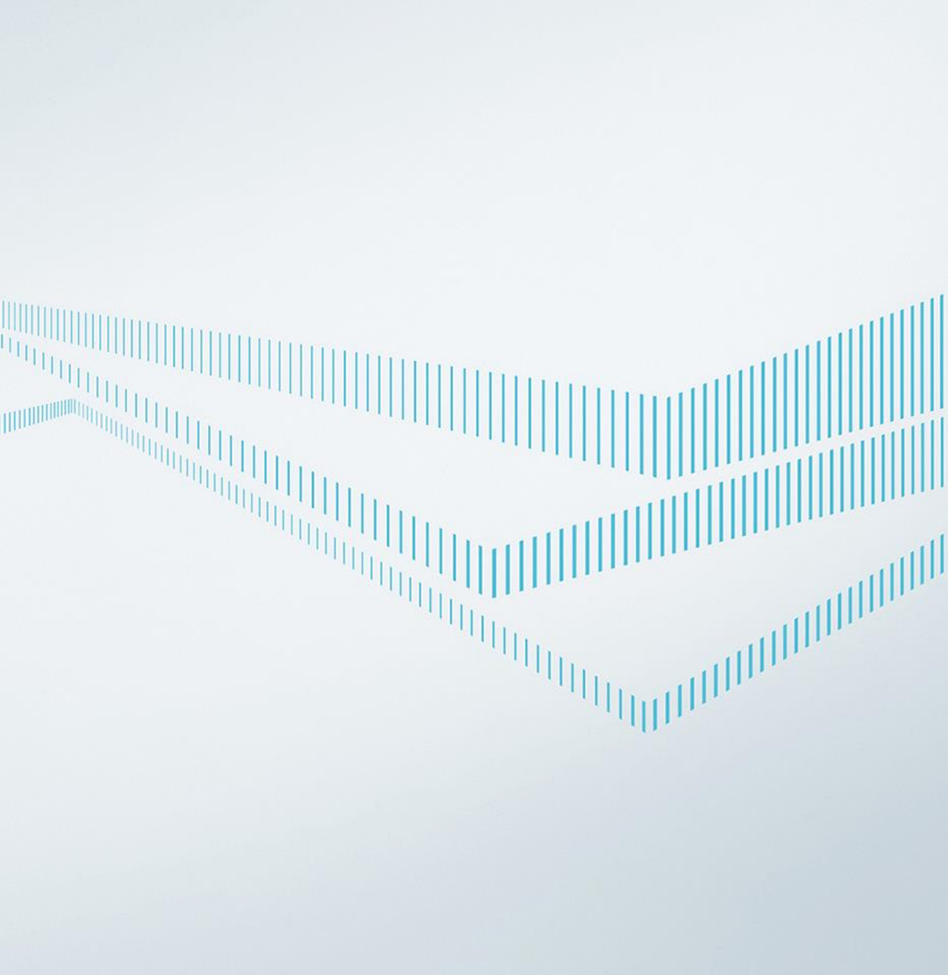
- Limitations of standard interfaces for ASICs as **interface to physical world**
- Ultra high **performance**
- Ultra low **power**

Impact on digital design flow

- Appropriate **floor plan, timing and power models** for analog macro
  - Accurate position of ports and precise shape
  - Fully synchronous interfacing with all timing arcs modeled for all corners
  - Precise power dissipation over time; decoupling requirements
- Appropriate **simulation model** for analog macro
  - Automated extraction of upper level wiring
  - Behavioral model for atomic functions

Digital Chip Design

SIEMENS
Ingenuity for life



Herbert Taucher
Head of Research Group Electronic Design
Corporate Technology

Siemensstraße 90
1210 Vienna

Phone: +43 51707 37626
Mobile: +43 664 80117 37626

E-mail:
herbert.taucher@siemens.com

Internet
siemens.com/corporate-technology