CHIP DESIGN GERMANY

AG Roadmap

# Research Topics

## Chip Design Germany Working Group Roadmap

Authors
Oliver Bringmann, University of Tübingen
Wolfgang Ecker, Infineon Technologies AG and Technical University Munich
Jan-Hendrik Oetjens, Robert Bosch GmbH
Norbert Wehn, RPTU University Kaiserslautern-Landau

### Authors

Oliver Bringmann, University of Tübingen

Wolfgang Ecker, Infineon Technologies AG and Technical University Munich

Jan-Hendrik Oetjens, Robert Bosch GmbH

Norbert Wehn, RPTU University Kaiserslautern-Landau

### Contributors

Mourad Elsobky, Robert Bosch GmbH

Max Georg Fend, Robert Bosch GmbH

Michael Graf, Robert Bosch GmbH

Husni Habal, Infineon Technologies AG

Göran Jerke, Robert Bosch GmbH

Felix Lang, Robert Bosch GmbH

### Working Group Office

Andreas Vörg, edacentrum GmbH

# Table of Contents

# Executive Summary

This document has been prepared by the Chip Design Germany (CDG) Working Group Roadmap to guide research and development priorities in chip design, including IP development, between 2025 and 2030. This roadmap summarizes, explains, and motivates 17 priority topics.

This roadmap deliberately excludes the widely discussed, acknowledged, and already significantly funded topics of agentic and generative AI, chiplet technology, and quantum computing. Also not covered is the open-source aspect, as studies on that topic have recently been published. Also important to mention is that the proposed topics target innovation beyond classical and state-of-the-art design flows and IPs, be it commercial or open source.

The first topic cluster covers novel, automated methods to speed up the design process and simplify the integration of chips in a customer's product. These topics include *Customizable Design Automation, Overall and General Design Productivity Boost, Toolchain for Customized Signal Processing with a Focus on the Use of Open Architectures, and Customization & Mixed Deployment*. This cluster is complemented by the design topics of *Low Power Design Techniques and Valuable Verification and Validation*, with or without automation.

The second topic cluster addresses the compute and firmware design perspective, focusing on a paradigm shift *From Compute-Centric to Communication-Centric System and Architecture Design, Methods and Architectures for Software-Defined Systems,* as well as *Highly Efficient Firmware*. IPs are also essential for improving chip design, but with the emphasis on creating value. Therefore, the following topics are also recommended for future innovation: *Valuable IP, Automated Technology Migration and Automated Analog Design*, as well as *Compact High-Performance Sensors*.

Specific to semiconductors from Germany are innovations in the area of *Robust Design against Fast Transient Interference, High-Voltage Electric/Electronic Architecture at the ASIC Level, System-Level Design for Innovative Electric/Electronic Architectures*, and *System-Level Co-Design for Analog/RF Systems in Small-Node Technologies*.

Lastly, Data-Driven Test Time Optimisation and DFT Design Improvement are essential, as they cover not only design-related aspects but also manufacturing and live-cycle aspects, and as they impact a double-digit percentage of the manufacturing cost, i.e., billions of Euros.

Each topic is summarized on one page, describing the motivation (why), the targeted improvement (what), and a potential solution idea (how). Each topic covers a wide range of applications that are particularly important in Germany, including automotive, industrial, medical, avionics, space, defense, home automation/ IoT, and more. A table at the end of the document summarizes and categorizes the research focus topics, describing them according to the addressed circuit and design domains.

These topics were selected because they offer significant value for chip design in Germany, by offering unique selling points for products or by having the potential to dramatically shorten design time.

# Introduction

Semiconductors are the foundation of modern technology, embedded in everything from cars and industrial machines to smartphones and household appliances. Indeed, no electrical device can function without integrated circuits, and **chip design** is the essential first step in creating these microelectronic components. From Industrie's perspective, strengthening semiconductor innovation and design capabilities is of paramount importance for Germany's economy, security, and technological sovereignty. Recent global chip shortages and geopolitical shifts have highlighted Europe's dependence on external sources. In response, Germany and the European Union are significantly investing in the semiconductor sector to reduce vulnerabilities and capture greater value in the supply chain. For example, the European "Chips Act" initiative aims to **double Europe's share of global semiconductor production from roughly 10% to 20% by 2030**. Achieving this goal will require not only expanding fabrication capacity but also developing world-class design expertise domestically to create the next generations of chips.

This document is **the strategic roadmap that summarizes, explains, and motivates 17 focus topics in chip design,** including IPs. This document has been prepared by the *Chip Design Germany (CDG) Working Group Roadmap* – a Federal Ministry of Technology, Research and Space (BMFTR) funded network uniting industry, academia, and research institutes – to guide research and development priorities in chip design between 2025 and 2030. The main contributors to this document are from **Industry and Academia**, each with deep, broad know-how in real-world **chip design**.

The CDG network was launched with the conviction that expanding Germany's chip design expertise will bolster the nation's innovation and competitiveness and contribute to Europe's technological sovereignty. It serves as a platform for pre-competitive collaboration and recommends to the BMFTR on all questions of chip design and microelectronics strategy. This roadmap is one of its key outputs: it identifies priority themes, challenges, and opportunities in semiconductor design, intended to inform upcoming research funding calls and initiatives by BMFTR and other stakeholders.

This document presents topics for a comprehensive 3-5-year plan developed by the CDG Working Group Roadmap, outlining critical technology topics and innovation directions in chip design. It is intended for stakeholders in research funding and policy, providing a strategic view of where Germany and Europe can focus R&D efforts to enhance competitiveness and technological sovereignty in microelectronics.

The proposals target innovation beyond classical and state-of-the-art design flows and IPs, be it commercial or open source. The guiding principle of the roadmap topics was to cover areas that boost Germany's companies, whether semiconductor providers or users.

**Goals and Scope:** The roadmap covers a broad range of chip design challenges – from cross-cutting design methodology improvements to domain-specific technology needs. It emphasizes **design automation, productivity, low-power design, verification, testing, system architecture, software-hardware codesign, analog/mixed-signal techniques, and sensor integration**. This roadmap deliberately goes beyond currently widely discussed and already significantly funded topics (agentic AI/generative AI, chiplet, quantum computing, etc.). Also not explicitly considered are open source and AI, as they are ways to make automation (or a solution that searches for its problem) and ways to manage R&D and dissemination. In rare cases, they are named as a solution option.

Instead, the focus is on areas where targeted research investments over the next few years could have a high impact, fill gaps, and leverage Germany's strengths (such as automotive electronics, industrial automation, and high-quality engineering). The roadmap's scope aligns with national and European initiatives—for example, fostering open design platforms and talent development—and is closely integrated with the goals of the **European Chips Act** and related programs.

**Structure of the Document:** This introduction is followed by the 17 sections of focus topics. Each topic is elaborated: (1) the motivation for **why** it is strategically important, (2) a description of **what** the topic is and includes with relevant technical background and state-of-the-art context, (3) key research directions and ideas on **how** to address the challenges. At the end of the document, a table is embedded that summarizes and categorizes the proposed research topics by circuit and design domains. The proposed research topics have a broad impact across all application domains and disciplines important to German industry – automotive, industrial, consumer, medical, defense, space, avionics, and others. Finally, a glossary of key terms and acronyms is provided.

**Audience and Tone:** This document is written in a formal, informative tone appropriate for research funding agencies, policymakers, and R&D decision-makers. It balances technical depth with high-level rationale to convey both *how* specific innovations can be pursued and *why* they matter for Germany's strategic position, in a real-world context. Ultimately, the roadmap intends to guide coordinated actions – **collaborative projects, funding programs, talent development, and infrastructure investments** – that will enable Germany's chip design community to thrive and deliver world-class innovations in the next 3–5 years. By focusing on design capabilities (in addition to semiconductor manufacturing), Germany can reinforce its industrial competitiveness and contribute to Europe's technological self-reliance.

# 1. Customizable Design Automation

### Why

When looking at today's design techniques, we observe that we have been designing digital and analog hardware IP in the same way for decades: RTL design on the one hand and schematic/ layout-based design on the other hand. Only some incremental enhancements have been made.

Very specific automation is available, e.g., via high-level synthesis for digital design or via layout generators for inductors for analog design. But all these techniques are very domain- and application-focused and do not bring improvement across a wider area. **Customizable design automation on top** of **existing** and **established flows** would help **broaden the automation approaches beyond RTL** and schematic through a wide range of very specific tools. Examples are **application-specific platform composition** and the **generation of complete IPs** that go beyond the generation of register interface and safety/security-related code modifications. As increasingly specialized or general-purpose cores become part of the design, design automation is also needed for firmware, i.e., software that closely interacts with hardware (HW). HW debug is an additional important aspect that requires improvement.

### What

Conceptually, the **development of own domain-specific tools** shall be **supported**, whether for **architectural design**, **chip design**, **prototyping,** or **ramping the chip in the lab**. This includes a mix of **customized design automation** and IP reuse techniques, which are tailored to meet the **specific requirements** of **in-house design flows** and dedicated **application domains**. In this context, there is a need for tools to facilitate **application-specific system composition** and **parameterization**, based on company-specific **building blocks** for compute, memory, and communication. It is essential to combine this with an efficient setup of CI/CD pipelines along with automated documentation and verification processes.

### How

From today's perspective, two general directions may lead to making customizable design automation and deployment flows a reality. The first direction is to **provide a toolbox** for **creating** these **design tools**. This **toolbox includes internal models** and their **formal definitions**, tools for **model composition** and **optimization**, and **transformation tools for lowering** (reducing the abstraction level). It should also support reading and writing using standard formats. Specifications and parts of the toolbox – if not the complete toolbox – should be made open source.

The second direction is to use **specifically trained LLMs** to **support** the **EDA tool generator** to a wide extent. As this might not be achievable with today's available technologies, a mixed approach consisting of a configurable building block and LLM generation, including the use of LLM technology for specific tasks, such as automated creation of (customer) documentation, may be the best way to follow.

Both directions must be supported by automated validation and verification of the tools. If this is not possible, an independent second path must be built to ensure the tools can be used in safety-critical designs.

## 2. Overall and General Design Productivity Boost

### Why

**Productivity is a key competitive factor** for highly developed economies such as Germany, which rely on an educated workforce and highly specialized engineering talent. While complex and efficient methodologies are well established in software engineering, the **productivity gap between software and hardware design** remains substantial. For instance, digital hardware design is still largely based on Register Transfer Level descriptions. Moreover, incremental tool improvements are often neutralized by new challenges such as increased verification complexity, tighter power-performance-area (PPA) constraints, and heterogeneous system integration. **Bridging this productivity lag is essential** to maintain technological leadership and address increasing system complexity.

### What

It is necessary to **rethink and restructure the design processes** across all relevant domains (analog, digital, sensors, firmware, non-electrical components, packaging, etc.). Traditional workflows are still dominated by low-level specifications, which require significant manual effort and limit the scalability of design processes as system complexity grows. The entire design chain — from specification and early architectural planning to implementation and verification — must be considered. To resolve this, design productivity must no longer be viewed as a local or tool-specific issue, but as an overall concern. Furthermore, design approaches must be restructured to ensure alignment with higher system-level objectives – such as functionality, reliability, and energy efficiency – by creating a tighter link between low-level implementation and high-level architecture and system design.

### How

A significant increase in microelectronic design productivity requires **integrating advanced design methodologies, domain-specific automation, and AI-enhanced workflows**. Next-generation EDA tools have to offer **higher levels of abstraction**, enabling early design space exploration with automated refinement towards implementation, supporting holistic, cross-disciplinary design methodologies with hardware-software co-design and cross-verification, and co-optimization of analog and digital blocks. **AI techniques, in particular Large Language Models (LLMs),** are emerging as productivity accelerators across the design flow. LLMs support context-aware HDL and constraint generation, can optimize tool parameters, suggest architecture and circuit variants, and detect inconsistencies early. Domain-specific and hierarchical methodologies remain essential to manage complexity. AI-based co-pilots can be embedded into structured workflows to provide interactive, abstraction-level-specific assistance during specification, verification, and implementation.

## 3. Low Power Design Techniques

### Why

The need for **low-power design in microelectronic systems is rapidly growing** due to multiple converging factors. As Moore's Law slows, gains in transistor energy efficiency no longer keep pace with increases in transistor density. This imbalance leads to higher power densities and thermal challenges, especially in densely integrated circuits. In parallel, the carbon footprint of compute infrastructure has become a major concern, particularly in the context of climate targets and sustainability efforts. Low-power design is therefore also a key enabler for greener electronics. Additionally, the proliferation of battery-powered devices from IoT sensors and wearables to edge AI accelerators demands ultra-low-power solutions to extend operational lifetimes and enable energy-autonomous operation.

### What

To address these challenges, **power optimization becomes a fundamental concern throughout the entire design flow**. This includes integrating power and energy awareness from the earliest stages of system-level and architectural exploration, treating power consumption as a primary constraint alongside performance and area. Design methodologies must support automated insertion of low-power mechanisms at every level—from system and RTL levels through gate-level down to physical implementation. Such strategies must be applicable across a broad range of SoC platforms, covering both ultra-low-power edge systems and performance-oriented central compute nodes. This requires techniques that scale across different power and performance envelopes. The overarching goal is to enable efficient **power-performance-area (PPA) trade-offs across the full stack in a scalable, systematic manner**.

### How

Achieving this requires methodological advances beyond standard low-power techniques such as clock and power gating, dual-Vth design, and dynamic voltage and frequency scaling. **Design methodologies and corresponding research must focus on developing advanced power-aware design and analysis methods that operate across abstraction levels**—from single transistors and gates to full SoCs—and across disciplines, capturing interactions in the thermal, timing, and electrical domains. Area and power optimization must be deeply embedded in the complete Architecture-To-GDSII flow. Furthermore, **AI offers new capabilities** for automating optimization, predicting power bottlenecks, guiding architectural decisions, and accelerating verification. Such AI-based methods can enable more intelligent and adaptive power-aware design workflows.

# 4. Valuable Verification and Validation

### Why

As already stated in the 1999 ITRS roadmap about 25 years ago, verification effort consistently exceeds design effort. Despite the continuously increasing design complexity, e.g., resulting from continuously decreasing design nodes, **the fraction of verification in the overall design process still increases**. One reason is that verification often causes some aspects to be covered very often, i.e., one million times, and others are covered only to a small extent, if at all.

There are other reasons why bugs escape: they are specification bugs, since specifications are often unverified. Another reason is that all used coverage metrics are "negative" metrics, i.e., everything that is not covered is a potential source of an error. However, a specific degree of coverage does not guarantee the absence of bugs.

Finally, verification is not only functional verification of digital systems. Analog and non-electrical properties must also be verified, including their interactions with the digital part, the software, and other system components.

### What

Generally, **verification must be made more efficient**. Methods are needed to avoid over-verification and identify potential holes. **Ideally, methods are needed that identify the unknown** (what has not been verified). Further **debugging must be accelerated**, as this accounts for the largest fraction of verification time. Finally, **verification engines** such as linters, checkers, formal verification tools, or simulators **must run faster**. Especially, formal verification must be able to handle more than ~100 million gates.

When bringing to mind the above statement, "identifying the unknown" or sloppy knowing the unknown," it becomes obvious that disruptive breakthrough innovation is needed, which is not visible from today's perspective.

### How

As this innovation may never – or not in the next 10 years – be found, **finding a variety of domain-specific solutions may be the way to go**.

Similarly, the solution of challenging sub-topics may help, or even provide, the improvements needed in sum. First to mention is research on **new, more meaningful coverage metrics** that **prevent over-verification** of special pieces. Next is the speed improvement in the verification engines, e.g., through automated checkpointing and recursive branching across different verification streams. In this sense, formal methods must be enhanced to handle 10x larger/more complex components and 10x longer time windows, e.g., via **massive parallelization**. Low-overhead **FPGA-based accelerated formal verification** algorithms via an FPGA PCI card plug-in or co-processor is another way that might help forward. Further, **creating verification artifacts must be automated**, or verification artifacts must be reused from other domains, e.g., FW/SW for hardware verification and vice versa.

It is essential that verification engineers can think like designers, and that designers can think like verification engineers. This may be supported by better and broader education in both fields, whether as part of master's studies or lifelong learning. Lastly, we should strive to develop tools that identify unknown verification gaps.

# 5. Data-Driven Test Time Optimisation and DFT Design Improvement

## Why

The testing of manufactured chips, both during production and in the field, significantly impacts the cost and quality of semiconductors. In fact, **testing can account for anywhere from 10% to 50% or more of a product's overall cost.** However, companies typically do not disclose detailed testing costs, as they are often bundled with manufacturing expenses. Several factors contribute to the high cost of testing. For instance, in industries like automotive, rigorous testing is required to ensure extremely low defect rates. This involves subjecting chips to various stress tests, including temperature and voltage fluctuations, which can lead to complex fault models and a higher likelihood of defects. Additionally, some testing methods, such as trim and repair, require re-programming or even physical modifications to the chip using lasers or other tools. Other challenges in chip testing include the relatively low frequency of interface signals, which can result in slower test data transfer rates due to the length of wires on the so-called load board. Furthermore, analog testing presents its own set of difficulties, as it requires evaluating analog behaviour in both the time and frequency domains. Moreover, sensors often require consideration of non-electrical signals, such as pressure, light, or other environmental factors.

## What

First, the **complete chain from Automatic Test to Equipment to the transistor level must be addressed**. This also includes consideration of in-system test (on the PCB and in application) and health monitoring in addition to manufacturing (wafer and post-packaging) test. In this sense, all design disciplines, such as analog, digital, and non-electrical, **shall be covered**.

Second, **fault models, i.e., the models of potential failure mechanisms, must be rethought.** There are still escaping faults despite continuously increasing features in fault models as the already supported cell-aware faults. This is one reason why the number of test vectors to be executed continuously increases. Fault models must be developed that have relevance, i.e. correlate with actual defects. Data for DFT improvement and inefficient – or highly efficient – tests may be gained from production tests and fed back to the test pattern generation. In addition, test models shall support an increase of test speed.

## How

The headline over solution potentials is leverage synergies means **make use of items that have not been utilized or that have been separately considered and implemented**. One example is to utilize test data and detect more and less probable errors, potentially using AI methods. More probable errors are to be tested early, potentially leading to early error detection, which allows for stopping the test at this point. Another test-data-driven improvement lies in **identifying and removing inefficient tests**. Over that, machine learning might be used for error prediction and dynamic, risk-based test optimisation.

Further synergies can be achieved in the field of design for reliability, security, safety, and debug.

Finally, **analysing production and test data to identify inefficient or redundant test structures is a promising data-driven approach.** Similarly, outside synergetic approaches are questioning the standard test approaches based on scan and the development of efficient self-test strategies.

A totally new aspect is the increase in yield and quality by analysing and correcting systematic errors.

## 6. Robust Design against Fast Transient Interference

### Why

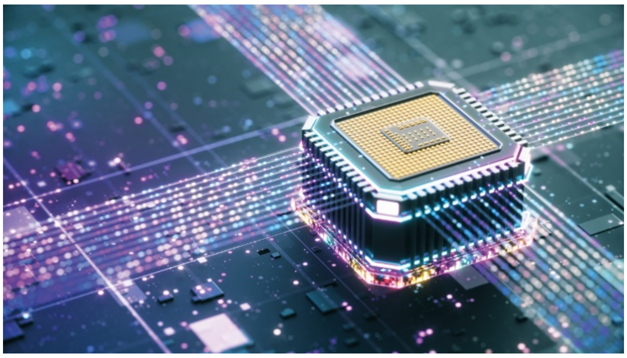
As the complexity of microelectronic components increases, so does their sensitivity to external disturbances during operation. Coupled pulses can reach **safety-relevant circuit components** via conduction, indirect coupling, or air discharge, potentially leading to functional faults. In the worst-case scenario, this could result in central control units being switched off while driving or airbags being triggered incorrectly. Such malfunctions could have **catastrophic consequences** for end users, especially in the context of fully automated driving. Standardizing these scenarios is extremely challenging due to their high application specificity and has not yet been achieved. A systematic analysis of **vulnerabilities in integrated circuits** has not yet been carried out, largely due to the wide variety of applications. This work would represent a first step toward addressing this gap.

### What

The overarching goal is to identify potential weaknesses in electronic components used in selected applications and to develop concepts to improve their robustness against external disturbances. The challenge lies in the diversity of applications and the range of possible fault mechanisms that may occur in each environment. To avoid deriving recommendations and improvements that apply only to isolated cases, it is essential to first establish a **comprehensive overview of components at risk**. Circuits selected as test vehicles should be widely used across domains. Typical examples of such IPs include **power supply units** and **communication interfaces** such as CAN, LIN, or xBASE-T1 (Ethernet).

### How

To systematically assess and enhance the robustness of integrated circuits (ICs) against external disturbances, a **structured methodology** is proposed. The process begins with the definition of representative disturbance scenarios, including the types of stress pulses and the operating conditions under which ICs are most vulnerable. These scenarios serve as the basis for reproducing fault behaviour in a controlled laboratory environment, enabling the development of **reliable and repeatable test procedures**. Once the test conditions are established, the next step involves identifying sensitive functional blocks within the ICs. This analysis combines empirical testing with design-level insights to uncover the underlying mechanisms of failure. To support this investigation, a **dedicated simulation environment** is developed, allowing for the reproduction and analysis of fault behaviour through fault injection and behavioural modelling. Based on the findings, **targeted design improvements** are proposed for the most critical blocks. These measures are then implemented and verified using custom test structures, including test chips that incorporate both baseline and improved versions of the circuits. The effectiveness of these improvements is evaluated through **comparative analysis**. In parallel, the validity of the test procedures is assessed to ensure they reflect real-world conditions. Finally, the insights gained are consolidated into general design guidelines for robust ICs and modules operating under powered conditions and exposed to external disturbances.



## 7. High Voltage Electric/Electronic Architecture at ASIC Level

### Why

The automotive industry is undergoing a significant transformation as it increasingly shifts towards **electric drives**, computing, and additional features for the 'Smartphone on Wheels'. This leads to a higher power consumption of various units, which causes significant energy losses if the conventional 12V cable harnesses are not scaled up in diameter. The consequence would be a notable increase in vehicle weight and higher material costs, especially due to the use of copper in larger cables. To address this, the industry is moving toward **compact and efficient high-voltage 48V E/E architectures**, ideally implemented at the **ASIC level**, which serve as the core supply components for ECUs and subsystems. **Raising the voltage from 12V to 48V reduces the current by a factor of four for the same power**, allowing thinner cables and avoiding the need for heavier wiring. As this evolution progresses, the demand for high-voltage (HV) ASICs, particularly in BCD (Bipolar-CMOS-DMOS) technologies, is increasing – driven by the need to improve performance, efficiency, and reliability. However, this shift also introduces challenges at the ASIC level, such as physical limitations of on-chip devices, thermal management, electromagnetic compatibility (EMC), and the integration of multiple subsystems. These issues must be addressed to enable a cost-effective and scalable implementation of future vehicle electronics.

### What

To tackle these challenges, the focus is on developing **robust and efficient key modules** for a **48V E/E architecture at the ASIC level**. This architecture must meet the stringent requirements of modern and future vehicles, ensuring high performance, reliability, and optimized power management. It must also support stable and reliable in-car communication systems to handle the increasing complexity of vehicle functions and increasing data rate needs. This architecture will serve as the **foundation for future vehicle platforms**, enabling the integration of new technologies and functionalities that will define the next generation of automotive innovation.

### How

Developing effective concepts for a 48V E/E architecture at the ASIC level calls for a focused approach to the increasing power demands, weight, and cost constraints of conventional 12V systems, and the physical and thermal limitations of integrated circuits. The process starts with analysing the transient load behaviour of 48V actuators (e.g., electrical motors or eCompressor), maximum conditions, and dynamics under fault conditions, and thermal stress on power MOSFETs and integrated drivers. Based on this, ASIC concepts are developed that integrate high-voltage stages, regulators for efficient down-conversion of the higher voltage, galvanic isolation, and protected circuits for gate drivers and PHYs like CAN, LIN, or FlexRay, and on-chip diagnostics for thermal and electrical faults. These are validated through mission-profile-based simulations and early silicon to ensure performance, EMC compliance, thermal robustness, and cost-effective scalability. Close collaboration with Tier 1 and Tier 2 suppliers ensures that the developed concepts are compatible with existing vehicle platforms and can be standardized across applications. Through iterative refinement, testing, and stakeholder feedback, the most promising solutions are matured into **robust, efficient, and scalable ASIC-based architectures**. These form the foundation for future vehicle electronics, enabling high-performance, reliable, and sustainable mobility systems.

# 8. System-Level Design for Innovative Electric/Electronic Architectures

## Why

Modern electronic systems in sectors like automotive, avionics, space, and industrial automation are rapidly increasing in complexity and adaptability. However, current design methodologies and EDA tools are not evolving at the same pace. While effective at the RTL level, they struggle with system-level challenges. A key issue is the lack of **architectural flexibility** and **lifecycle adaptability** in early design stages, making it difficult to assess whether systems can support future updates or product variants without costly redesigns. Another significant gap is the fragmented integration of analog and RF components, which are often excluded from early models, leading to integration issues later. **Security and safety** aspects – such as isolation, redundancy, and minimizing attack surfaces – are rarely addressed early, resulting in reactive solutions. Additionally, architecture choices increasingly affect thermal, mechanical, and cost factors, yet current tools lack a **unified framework** to assess these interdependencies. Furthermore, the increasing heterogeneity of system components – ranging from sensors to embedded controllers – requires a more integrated and scalable design approach that can accommodate diverse functional and non-functional requirements from the outset.

## What

The goal is to develop a **next-generation system-level design methodology** that enables early, accurate, and holistic evaluation of complex architectures. It should allow designers to assess **flexibility**, **security**, and **long-term adaptability** from the outset. The methodology must support the co-design of analog, digital, and software components and address integration issues early. It should embed **formal validation** of safety and security properties, such as fault tolerance and isolation, before implementation. To manage complexity, it must enable **AI-assisted**, multi-domain optimization across electrical, thermal, mechanical, and software constraints. The methodology should support scalable abstraction—from high-level exploration to subsystem refinement—within a unified framework. Finally, it should provide **traceability** from requirements to architectural decisions, enabling continuous validation throughout development.

## How

To realize this vision, several methodological building blocks are proposed. **Formal modelling and simulation** will support early evaluation of architectural flexibility. **Mixed-signal co-simulation** with abstract analog and RF models will help detect critical domain interactions. Safety and reliability will be addressed through **FMEA**, fault injection, and model checking. **AI-based exploration**, using reinforcement learning and surrogate models, will guide trade-off decisions across performance, cost, and safety. To support early-stage decision-making, the methodology will include fast feedback-driven evaluation loops that allow iterative refinement of architectural alternatives under evolving constraints. **Executable traceability** will link requirements, design decisions, and validation artifacts, enabling continuous verification. It supports model-based development for effective cross-domain integration.

## 9. From Compute-Centric to Communication-Centric System and Architecture Design

### Why

Historically, **designing and optimizing a chip** meant **focusing on compute units**. ALUs, CPUs, and specializations thereof: DSPs, GPUs, TPUs were dominant as well as their PPA figures (Power-Performance-Area). Increasingly, the bottlenecks are in moving data, be it off-chip as well as on-chip. The evolution started with simple buses, went over multi-layer buses to crossbars, and finally to network-on-chips (NoCs). As these interconnects are all proven and generic, they need plenty of memory operations to communicate on data-word abstraction or consider on-chip communication at the packet level, which results in **substantial PPA overhead**. However, today applications are dealing more and more with **multi-data-stream handling** coming from different or similar sensors, such as audio, radar, video, and gyroscope, or from real and virtual sensors. These data streams need to be processed and merged. Conclusions must be taken and actions computed. The data streams are **heterogeneous in data rate, dimensionality (1D-4D), representtation order, and other properties,** such as compression type. Early-stage design must therefore consider communication aspects more strongly. This must be related to the way and order in which data is processed, read, or written.

### What

Unfortunately, there is only limited design support for communication-centric designs, neither at the early analysis level nor at the implementation and IP levels. To name only a few challenges: Communication-centric system modelling is based on queuing models, which run quickly but ignore content and functionality. SystemC, the dominant language for system modelling, is used bus-centrically, especially when using the TLM2 standard. There are proven pseudo-standard ways to model a bus that have serious difficulty breaking out of the standard bus model to address the upcoming application-specific needs of multi-data-stream handling. When optimizing in a compute-centric way, time estimation remains limited to the area required by the compute unit. **Communication-centric design needs to focus on novel application-driven streaming architectures capable of cross-device pipelined execution**. This requires consideration of off- and on-chip wire lengths, which must account for the SoC's floorplan. Remains the design and trade-off challenge: Even though the focus lies on communication, the interaction with computation remains an issue. Consequently, the trade-off analysis will be even more complex.

### How

The most important thing is to raise awareness of the topic of Communication-Centric System and Architecture Design, and that it requires quantum leaps in design, which cannot be achieved through small, incremental improvements with today's technology. Ideally, physical and system design must be considered together, meaning the floorplan and interconnect are designed in parallel and used for both PPA and throughput estimation. **Interlinked estimation methods** for SoC wire delay and communication cost must be provided with **an error of ~10% to get valuable feedback**. **Estimation must be fast** to **evaluate many architectural and microarchitectural trade-offs and to support "what-if" analysis of performance, area, power, and throughput**. The **definition and trade-offs of possible architectures**, with a focus on heterogeneous, application-streaming approaches, must be simplified to allow focus on architecture-variant analysis. This also requires the **availability of appropriate IPs**, including collaterals covering physical, RTL, behavioural, **and SW views**. Finally, SystemC-based system modelling needs an **agreed practice to model off-and on-chip communication** at various levels at a reasonable cost. In addition, a **common view for HW with SW** must be established.

# 10. Valuable IP

## Why

Europe, especially Germany, is a high-cost region. It depends on products that deliver value to customers—individuals, companies, or authorities. This cost structure requires building valuable products that offer value through quality and innovation. Two aspects are important: high quality remains relevant but is increasingly secondary to time-to-market. Time-to-market is addressed in section 0. Innovation can come from how we integrate components and configure/program them, as addressed in section 0. Valuable IP components are another pillar of a product's unique selling point. **Valuable IP refers to components that offer technological differentiation, economic leverage (e.g., reuse, licensing), outstanding PPA (power, performance, area), or critical system-level advantages such as efficiency, security, or configurability.** RISC-V has been in focus in this area for a while, and recent R&D projects targeting ecosystems or core variants have accelerated its adoption. Unfortunately, the RISC-V ISA and most of the underlying technologies are not new. However, the scalability and extensibility give a wide range of opportunities to bring value beyond the commodity RISC-V ISA defined in the non-privileged specification. The availability of a base ecosystem can catalyse novel approaches by providing a starting point. RISC-V serves both as a flexible baseline for integrating custom IP and as a fertile ground for architectural innovation, especially beyond general-purpose computing.

## What

Conceptually, we need **IP innovation with RISC-V—not on RISC-V—and also innovation beyond it.** This includes architectures, IP-to-system integration, new compute paradigms beyond CPU and GPU, and non-compute paradigms, as highlighted in section 0. Innovation is needed—and it requires knowledge across horizontal (multi-domain, multi-disciplinary) and vertical (from ideation to commercialization) stages. **Human inherent striving for innovation must be freed and accelerated**. This topic complements section 0 by contributing foundational IP blocks that enable system-level innovation, accelerate time-to-market, and support domain-specific optimization.

## How

The challenge is to strive for the unknown. Therefore, actions must be taken to prepare and widen the know-how. **Digital design techniques beyond CPUs** must be structured, formed into a discipline, and prepared for learning and knowledge sharing. To unlock creativity from this foundation, **novel forms of R&D projects and programs must be explored,** ambitious goals set, and failure accepted as a real option. **Valuable IPs must be considered and realized along the value chain: Design automation (see section 2), IPs, chips**, and chip-based products must be envisioned together. **Measures must be taken to foster the know-how exchange along this chain**. To give examples, novel engines, such as state-machine-driven programmable control-flow engines or boolean satisfiability engines, as well as decision-tree engines, must be built hand in hand with design automation for both constructing and using the accelerators. Easy-to-use, highly efficient, and therefore low-cost, high-security IPs are another example – useful, for instance, in home automation, novel ways of integrating memory and digital logic are another.

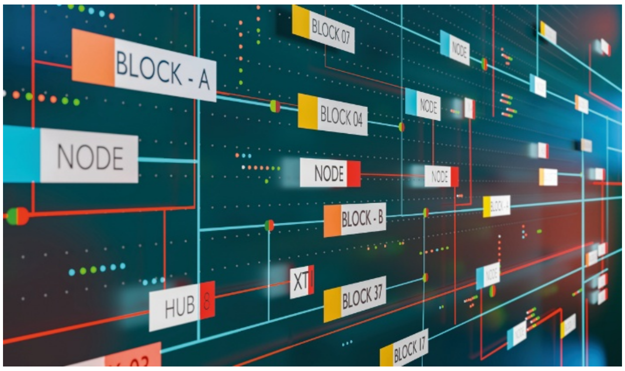
# 11. Toolchain for Customized Signal Processing with a Focus on the Use of Open Architectures

## Why

DSP development for automotive mixed-signal sensor ASICs is currently characterized by fragmented toolchains and proprietary solutions, which introduce inefficiencies, increase costs, and limit flexibility across the design cycle. At the same time, the demand for **advanced DSP algorithms** is rising due to the proliferation of **ADAS (Advanced Driver Assistance Systems)** and the evolution toward **software-defined vehicles (SDV)**. These trends highlight the need for **flexible, modular, and integrated DSP toolchains** that support rapid iteration, component reuse, and maintainability. While open hardware platforms such as **RISC-V** offer promising deployment options, their application in DSP remains underexplored. This research topic addresses the need to advance DSP toolchain design and deployment strategies, with a focus on reduced reliance on closed ecosystems.

## What

The research focuses on the development of a flexible, modular, and integrated DSP toolchain for implementation on platform-independent hardware targets such as those based on the open **RISC-V instruction set architecture (ISA)**. The envisioned toolchain supports a wide range of DSP use cases and aims to improve development efficiency, portability, and long-term maintainability. A core aspect is the integration of reusable standard algorithms for common DSP tasks, based on open and resilient technologies to reduce dependency on proprietary IP and enhance strategic autonomy. The toolchain is designed to enable consistent development and validation flows, facilitating smooth transitions from algorithm design to hardware deployment. Special attention is given to **automotive safety compliance**, balancing functional safety requirements with time-to-market and chip-level performance, power, and area (PPA) considerations.



The research investigates how DSP functionality can be efficiently mapped onto RISC-V-based platforms, including the design and evaluation of **custom instruction set extensions** to meet performance and efficiency goals.

## How

The research explores modular, open-source-based toolchain architectures tailored to the requirements of automotive sensor ASICs. A key approach involves the design of standardized, reusable DSP building blocks, optimized for platforms based on open ISAs such as RISC-V, to support efficient development and scalable deployment. To bridge the gap between high-level algorithm modelling and hardware implementation, the toolchain incorporates **semi-automated flows** from environments such as MATLAB/Simulink to deployable implementations, including automated synchronization of design views and traceable model-to-code transformations. The methodology includes safety-compliant verification strategies, such as reusable testbenches and formal validation flows, aligned with automotive-grade functional safety standards. Early prototyping on **FPGAs and test ASICs** enables iterative performance tuning and architectural exploration. The research is grounded in a **requirements-driven approach**, informed by real-world use cases, safety constraints, and performance targets.

# 12. Methods and Architectures for Software-Defined Systems

## Why

The rise of **software-defined vehicles (SDVs)** is fundamentally reshaping the role of processor architectures in embedded systems. In contrast to traditional automotive platforms, SDVs demand highly **flexible, upgradeable, and software-centric computing environments**. This evolution places unprecedented demands on processor ecosystems, where the integration of **heterogeneous processor IPs** – often from different vendors with incompatible toolchains, interfaces, and performance profiles – has become a major bottleneck. The **lack of standardization** across these components complicates interoperability, reuse, and long-term maintainability, which are essential in automotive systems with lifecycles exceeding a decade. Moreover, **safety-critical applications** such as autonomous driving and ADAS require **real-time performance**, fault tolerance, and functional safety at the processor level. Without structured methods to evaluate, compare, and integrate processor IPs, development becomes inefficient, error-prone, and vulnerable to late-stage architectural failures.

## What

Structured **methodologies for the design, evaluation, and integration** of processor architectures in SDVs are needed. A central goal is to enable **transparent, data-driven selection of processor IPs** based on standardized criteria such as performance, safety capabilities, compatibility, and lifecycle support. Architecture exploration tailored to processor-level design, using **modelling tools** that simulate realistic SDV workloads and provide early feedback on performance, power efficiency, and safety trade-offs, is another challenge that needs to be solved. A key pillar is the **standardization of processor interfaces and toolchains**, particularly within the RISC-V ecosystem, to reduce integration complexity and promote modular, reusable processor components. Additionally, research is needed to integrate **fault-tolerant mechanisms** – such as lockstep execution, error detection, and redundant cores – directly into processor architectures.

## How

To realize these goals, several solution paths are proposed. First, **processor IP evaluation** shall be formalized through benchmarking frameworks that define uniform metrics for performance, safety, and integration readiness. Second, **architecture exploration tools** shall be developed to simulate and optimize processor configurations under SDV-specific constraints. These tools shall support multi-objective trade-offs and enable early design validation. Third, the research must promote **interface and toolchain standardization** within open processor ecosystems like RISC-V, enabling interoperability and IP reuse across vendors and platforms. Finally, **safety-oriented design patterns** – including redundancy, error correction, and fail-operational modes – shall be embedded early in the processor design process to ensure compliance with **automotive safety standards** such as ISO 26262.

## 13. Highly Efficient Firmware

### Why

Firmware is the piece of software that is embedded in hardware and directly interacts with it. **Firmware must deal with hardware properties – especially resource constraints, power consumption, safety, and security – and must be very efficient.** The **demand for firmware has grown** with the expanding embedded systems market and the proliferation of smart devices across consumer electronics, automotive, communication, healthcare, and industrial automation. **Rising hardware customization and Edge AI applications emphasize the need for highly efficient firmware**. One of the key challenges in firmware development is programming specialized processors such as DSPs or, more generally, application-specific cores. As programming has increasingly shifted towards higher levels of abstraction (e.g., 4GL languages and beyond), **firmware development has received decreasing attention in academia**, causing minimal technical advances and a strong reduction of firmware-related content in education. This gap is particularly critical as **firmware plays a central role in enabling hardware-software co-design**, which is increasingly important in domains like AI accelerators, chiplets, inter and intra SoC communication, and real-time control systems.

### What

**Firmware development must be brought back into focus**, as it is essential for making hardware functional – and thus **is critical for addressing major challenges such as AI, decarbonization, and digitalization**. Efficiency in firmware development and methods to improve firmware performance must be advanced. Additionally, **ways of providing high-quality firmware at a reasonable cost are needed**. Beyond methods, suitable tools, libraries, and skilled developers are required. Another key aspect is the **formal description of the entire hardware system** – not just the processor – that **the firmware interacts with**. Standardized hardware structures (e.g., core + peripherals) enable reuse and simplify ASIC design, as functionality can be defined via firmware. This also **allows for late-stage logic changes through firmware updates**, often requiring only metal mask modifications instead of full redesigns. This approach supports platform-based design strategies and can significantly reduce time-to-market and non-recurring engineering (NRE) costs.

### How

**Firmware should be established as a third core pillar in semiconductor education**, alongside digital and analog hardware design. Research, innovation, and education must be strengthened. To lay the foundation, formalisms and databases with rules on the hardware/firmware interface, the syntactic and semantic structure of firmware, and good practices are needed. Building on this foundation, explicit methods – **such as generator- or transformation-based approaches, AI-supported techniques, or combinations thereof** – shall be developed. **Automation** should include **verification, documentation**, and **key development artifacts in the hardware context**. Detailed research and innovation aspects include assembler-focused IDEs – also supporting special instructions, **DSLs for assembler programming, and AI assistants** trained in best practices for **assembler development** – including coding guidelines and methods for analysing quality and PPA (power, performance, area) effects of firmware. In addition, research should explore how **firmware development** can be **integrated into agile** and **DevOps-like workflows**, including continuous integration and automated regression testing for embedded systems. It is essential to develop industry-grade methods and to focus on high-demand domains such as DSPs and on-chip communication programming. Emerging areas such as **RISC-V-based custom cores, chiplet architectures**, and **reconfigurable computing platforms** also present **new opportunities** and requirements for **firmware innovation**.

# 14. Customization & Mixed Deployment

## Why

**Application-specific hardware** is **essential** to **successfully address** the **diverging requirements across different application domains**. These requirements arise from differences in sensor configurations, the number and type of processing tasks, as well as real-time, resource, cost, and energy constraints. Therefore, hardware platforms need to be heterogeneous, highly scalable, and tailored to the specific needs of the application domain. Typically, these platforms are composed of standard compute elements (MCUs, DSPs, embedded GPUs, NPUs) along with custom compute units, memory components, and peripherals. However, **leveraging** the **full potential** of **customized platforms** requires **advanced software support**. **Without appropriate tooling** for **hardware customization** and **mixed deployment** of **software** and **AI models**, **development** becomes **cost-intensive** and **inflexible**. Efficient deployment, particularly in but not limited to edge AI scenarios, demands seamless retargeting capabilities from server-grade GPUs to embedded devices. The success of these platforms, therefore, depends on the availability of an efficient, extensible, and optimized deployment toolchain.

## What

There is currently a **significant gap** in the **availability of adaptable software toolchains** that support **platform customization** and **retargeting across heterogeneous hardware** or for **retargeting** from **server-grade devices** to **application-specific platforms**. Existing solutions are often fragmented, tailored to specific devices, and not easily extensible to support new application requirements or emerging architectures. For example, the rise of transformer models exposed the lack of suitable embedded hardware support for attention mechanisms, showing the inflexibility of today's deployment toolchains. Open-source tool platforms such as TVM were very helpful, but dominant GPU vendors are steering the tool development away from custom hardware to server-grade platforms. A similar trend can be observed in many areas, e.g., control algorithms, DSP, or cryptography. As a result, tool vendors are deprioritizing customizable and open tool support for heterogeneous, domain-specific deployments, leaving developers without comprehensive solutions for mixed deployment and configuration of specialized platforms. **Push-button solutions for hardware configuration and mixed deployment are still not yet in sight.**

## How

To unlock the benefits of application-specific and heterogeneous platforms, there is a need for a **highly customizable, open compiler toolchain** designed for **mixed deployment**. Such a toolchain must support: easy integration of **new operators**, **devices**, and **platform-specific extensions**, **deployment** across **heterogeneous platforms**, with complex memory hierarchies and interconnect architectures, including **streaming channels for pipelined execution, semi-automated generation, and adaptation** of toolchains based on modular building blocks, **advanced memory management**, including ahead-of-time scheduling, static memory allocation, rolling buffers, and memory reuse strategies, and **device-level customization**, such as **slicing compute units** into **execution islands**. It is advisable to build upon existing compiler frameworks, such as **MLIR**, **LLVM**, and **IREE**, and integrate mature kernel libraries like **OpenCL**, **ARM Compute Library**, and **vendor-specific micro-kernels**. An alternative approach may opt for abstracting compute capabilities by providing a stack of configurable functions as use in I/O-stacks. These efforts should aim to extend existing tools beyond single-task deployment (e.g., ML inference), towards orchestrating **complex networks of multiple software functions** distributed across **industrial-grade heterogeneous platforms**.

## 15. Automated Technology Migration/Automated Analog-Design

### Why

Analog integrated circuits, including power and RF components, are critical enablers across industries such as e-mobility, renewable energy, advanced driver-assistance systems (ADAS), AI/data centres, and IoT. These sectors demand analog building blocks with cutting-edge performance. However, **analog (or even RF) IP cannot simply be reused off the shelf. It must be reworked and modified for each product or migrated to a new PDK,** making the design process resource-intensive and time-consuming. Consequently, efficient and effective design of these components is essential for maintaining market competitiveness.

Unlike the highly automated digital design domain, **analog design workflows have remained largely unchanged for decades**. Most tasks are still highly manual, requiring extensive effort, time, and expertise at every stage to produce an IP that meets the required area or power constraints. These tasks are often on the critical path to market readiness, slowing innovation cycles and raising costs. The situation is further exacerbated by a global shortage of skilled analog engineers and the increasing complexity of modern circuits, driven by smaller nodes and advanced packaging technologies.

### What

Efforts and time spent on analog design must be reduced without increasing physical features as area, power, accuracy or linearity. **Design reuse must be fostered, and the development of new IP accelerated**. Manual work must be reduced and automation increased, particularly in layout tasks such as automated placement and routing under constraints, performance and yield optimization, circuit modelling, and technology migration. As full automation may not be achievable, **semi-automated approaches are essential.** These approaches must encode designer intent and integrate seamlessly into existing EDA environments and workflows. In addition to technological advancements, **major organizational and cultural changes are necessary.** Training programs at management, expert, and beginner levels must support the transition from traditional manual methods to programmatic approaches, with a focus on scripting, algorithm development, and machine learning.

### How

Analog circuit and layout generators capable of automating the entire design chain—from electrical specifications to optimized layouts—are a key innovation area. Goals addressing, e.g., power, area, and speed must be specified and prioritized. **Robust frameworks for automating analog technology design and migration need to be developed** and qualified, with a focus on integrating existing PDKs and ensuring compliance with design rules. Particularly, netlists and layouts must be efficiently migrated.

**Predictive simulations** should be incorporated early in the design flow to reduce iteration cycles. Generating **smaller, well-understood building block models** can serve as the foundation for more complex circuits and applications. **Heterogeneous integration**, such as chiplets and advanced packaging, must also be supported to enable analog designs to adapt to emerging integration schemes.

**Advanced AI methods** such as generative AI, transfer learning, and reinforcement learning offer promising opportunities. These methods must support modular workflows and effectively integrate designer expertise. Finally, training programs for analog designers in AI, scripting, and programmatic design frameworks need to be developed and made widely accessible. Strong emphasis should be placed on fostering cross-disciplinary collaboration between analog engineers and software developers.

## 16. System-Level Co-Design for Analog/RF Systems in Small-Node Technologies

### Why

The transition to **advanced technology nodes (40 nm and below)** in analog, RF (e.g., sensing, wireless, and wireline PHY), and mixed-signal IC design introduces significant challenges regarding design complexity, system partitioning, integration, verification, and productivity. As analog, RF, and digital domains increasingly converge in tightly integrated heterogeneous systems, a **holistic system-level co-design approach** across chip, package, and system levels becomes crucial. For example, current EDA tools lack the capability to automatically integrate automotive-grade requirements such as **functional safety** and reliability into design flows. The absence of suitable design and data flows hinders collaboration, especially in safety-critical sectors like automotive and industrial automation—areas where Germany holds a global leadership position. This includes RF-specific challenges like cross-domain verification, mission profile integration, and co-design with safety logic. **AI-assisted design** holds strong potential for automation and decision support but remains fragmented and insufficiently tailored to domain-specific needs. To ensure **technological sovereignty** and competitiveness, long-term interdisciplinary R&D is needed to bridge EDA, AI, and semiconductor design in RF- and safety-relevant workflows.

### What

To effectively address these challenges in analog and RF system design at advanced nodes, targeted R&D efforts are required. Key goals include developing advanced methodologies and solutions for chip- and system-level co-design, enabling **chip-package-system optimization** under thermal, mechanical, and electromagnetic constraints. Design flows must integrate automotive-grade requirements such as **ISO 26262 compliance**, mission profile modelling, and reliability modelling. Furthermore, R&D should focus on **standardized, interoperable frameworks** for system partitioning, prototyping, and verification across chiplets and heterogeneous systems. AI-assisted design must support metadata analysis, reusable design patterns, and **uncertainty-aware requirement handling**. Workflows should include AI-assisted generation, with verification using simulation and formal methods. RF-specific constraints, such as frontend integration and PHY-level modelling, must be addressed early.

### How

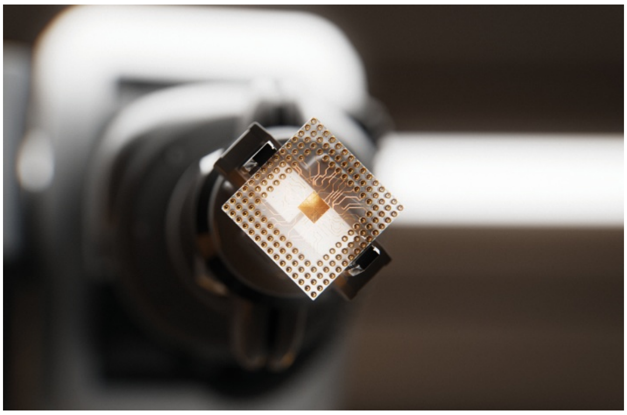
A coordinated set of R&D measures should close technological and methodological gaps in system-level co-design for analog and RF systems in small-node technologies. A key priority is the development of **modular, reusable design flows** supporting constraint-driven schematic capture and parasitic-aware simulation, including models for process variation and aging. AI-assisted design should enable early-stage exploration and **topology generation under uncertainty**, embedded in conceptual workflows to improve first-time-right outcomes. Tools must jointly model thermal, mechanical, and electromagnetic effects across chip, package, and board levels, enabling automated partitioning and interface modelling for chiplet and 3D-IC architectures. Design flows must integrate automotive-grade requirements like ISO 26262, mission profiles, and reliability prediction, with verification workflows supporting **traceability and certification readiness**. Standardized prototyping and verification frameworks should enable reuse and scalability, including reference designs and test benches. Finally, domain-specific ontologies and **metadata repositories** are needed to capture design intent and support semantic search and IP reuse. RF design must be supported by signal integrity and coupling models across system levels.

## 17. Compact High-Performance Sensors

### Why

The ongoing shift of sensor ASICs towards **advanced nanometre technologies** presents significant new challenges, particularly for analog designs. This technological progression inherently leads to more process variations, parasitic effects, higher noise levels, and increased design and verification complexity in analog circuitry. In fact, device miniaturization leads to higher **flicker noise** as the probability of charge trapping/detrapping events increases. Also, **substrate noise** due to digital switching activity on the same die couples to sensitive analog circuits. In addition, the transistor threshold voltage becomes a larger fraction of the lower supply voltage, which limits the available headroom for signals and reduces the achievable signal-to-noise ratio (SNR). Furthermore, simulating such accurate analog circuits with complex device models is computationally intensive while including additional corner analysis required for nanometre technologies. These factors necessitate **innovative approaches** to maintain and improve performance in critical sensor applications.

### What



The objective is to develop **compact, energy-efficient, and high-performance readout circuits** specifically tailored for modern sensor systems utilizing advanced manufacturing technologies. This will involve leveraging modern CMOS nodes (40nm and below) to effectively reduce both area and production costs while increasing digital content. The digital area is scaling well with CMOS scaling; however, the analog frontend area, which is the focus here, is not scaling. Therefore, **innovative digital-intensive readout architectures** are potentially advantageous, targeting the same performance of predecessor generations while scaling down the die area. A core focus will be to either save die area and cost or add more functions for the same area. Furthermore, a **seamless integration of both analog and digital functions** into a single chip should be aimed for, while rigorously ensuring testability, reliability, and suitability for demanding automotive but also other applications. The highest automotive functional safety standards are targeted, such as ISO 26262, which imposes another challenge for high-performance automotive sensors in such advanced CMOS nodes.

### How

To achieve these goals, the proposed methodology involves several key steps. Firstly, the development of highly integrated, **noise-optimized, digital-intensive capacitance-to-digital converters (CDCs)** is a key factor to enable a precise and compact sensor readout. Secondly, the integration of **compact high-voltage (HV) actuators** together with sensors is needed, as higher excitation voltage leads to increased sensor signal and hence higher SNR. In addition, we will exploit digital functions like **adaptive calibration**, background trimming, and digital filtering to actively assist and correct analog nonidealities, such as temperature drift or process variations in advanced CMOS nodes. The proposed process should involve comprehensive requirements analysis and application studies to define the precise needs. Subsequently, thorough evaluation and selection of suitable circuit architectures and best-fitting options should be identified. The practical implementation shall involve pre-development, layout design, and **rigorous testing of prototypes** on dedicated test chips to validate the proposed techniques.

# 18. Appendix

## Categorization of Proposed Research Topics According to Circuit Domains and Design Domains

The proposed research topics have a broad impact on all application domains and disciplines important for German industry – automotive, industrial, consumer, medical, defense, space, avionics, and others.

The categorization shows the major intended impact. It shows that Analog, Digital and Mixed-Signal or Digital and Software in the circuit domain often are tagged together. The RF and Mems circuit domains are tagged for few topics but are not of minor importance. In the Design Domain, Software and Architecture Level – with or without System Level as well as Transistor Level and Technology are tagged together as well.

| | | Circuit Domain | | | | | | Design Domain (Abstraction, Disciplin) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Section | Section Title | Digital | Analogue | Mixed-Signal | RF | Mems | Software | Software (Application SW and Firmware) | System Level | Architecture Level | Gate Level | Transistor Level | Technology |
| 1 | Customizable Design Automation | x | | | | | x | x | x | x | | | |
| 2 | Overall and General Design Productivity Boost | x | x | x | | | x | x | x | x | | | |
| 3 | Low Power Design Techniques | x | x | x | | | | | x | x | x | x | |
| 4 | Valuable Verification and Validation | x | x | x | | | x | x | x | | x | | |
| 5 | Data-Driven Test Time Optimisation and DFT Design Improvement | x | x | x | | | | | | | x | x | x |
| 6 | Robust Design against Fast Transient Interference | | x | x | x | | | | x | | | x | |
| 7 | High Voltage Electric/Electronic Architecture at ASIC Level | | x | x | | | | | | | | x | x |
| 8 | System-Level Design for Innovative Electric/Electronic Architectures | x | | x | | | x | x | x | x | | | |
| 9 | From Compute-Centric to Communication-Centric System and Architecture Design | x | | | | | x | x | x | x | | | |
| 10 | Valuable IP | x | | | | | x | x | | x | | | |
| 11 | Toolchain for Customised Signal Processing with a Focus on the Use of Open | x | | | | | x | x | | x | x | | |
| 12 | Methods and Architectures for Software-Defined Systems | x | | | | | x | x | | x | | | |
| 13 | Highly Efficient Firmware | x | | | | | x | x | | x | | | |
| 14 | Customization & Mixed Deployment | x | | | | | x | x | x | x | | | |
| 15 | Automated Technology Migration/Automated Analogue-Design | | x | | | | | | | | | x | x |
| 16 | System-Level Co-Design for Analogue/HF Systems in Small-Node Technologies | | x | | x | | | | x | | | x | x |
| 17 | Compact High-Performance Sensors | x | x | x | | x | | | | | | x | x |

## Glossary

| Term | Definition |
|---|---|
| ADAS (Advanced Driver Assistance Systems) | Electronic systems in vehicles that assist the driver in driving and parking functions. |
| AI (Artificial Intelligence) | AI refers to the ability of machines to perform tasks that normally require human intelligence. This includes learning from experience, understanding natural language, recognising patterns, making decisions and solving problems. AI systems use algorithms and models to analyse data and draw conclusions from it. |
| AI Compiler | A compiler optimized for translating → AI models into efficient executable code for specific hardware platforms. |
| Analog/RF-Design | The design of circuits that process continuous signals (analog) and high-frequency signals (RF). |
| Application software | These are programmes that perform specific tasks for the user, such as word processing (e.g. Microsoft Word), spreadsheets (e.g. Microsoft Excel) or image editing (e.g. Adobe Photoshop). |
| ASIC (Application-Specific Integrated Circuit) | A microchip designed for a particular application rather than general-purpose use. |
| Automated Test Equipment (ATE) | Systems used to test electronic devices for functionality and performance during manufacturing. |
| BCD (Bipolar-CMOS-DMOS) | A semiconductor technology combining bipolar, CMOS, and DMOS transistors on a single chip. |
| Bug | A bug in hardware design refers to an error, flaw, or unintended behaviour in the design of a hardware component or system. These bugs can arise from various sources, such as mistakes in the design specifications, errors in the implementation of the design, or unforeseen interactions between different parts of the hardware. → Debug |
| CAN (Controller Area Network) | A robust vehicle bus standard designed to allow microcontrollers and devices to communicate without a host computer. |
| CDG (Chip Design Germany) | A German initiative to strengthen domestic chip design capabilities. |
| Cell-Aware Fault Models | Advanced fault models that consider the internal structure of standard cells for more accurate test coverage. |
| Chiplet | A small, modular integrated circuit that can be combined with others to form a complete system. |
| Chips Act (EU) | European legislation aimed at boosting semiconductor production and innovation in the EU. |

| Term | Definition |
|---|---|
| CI/CD | Continuous Integration/Continuous Deployment (or Continuous Delivery) |
| Continuous Delivery | Continuous Delivery is similar to Continuous Deployment, but with a key difference: the deployment to production is not automatic. Instead, the code changes are automatically tested and prepared for release, but the final deployment step requires manual approval. |
| Continuous Deployment | Continuous Deployment is an extension of Continuous Integration where the code changes that pass all stages of the production pipeline are automatically deployed to the production environment. This means that every change that passes the automated tests is released to users without manual intervention. |
| Continuous Integration | Continuous Integration is a software development practice where developers frequently integrate their code changes into a central repository. Each integration is automatically verified by building the application and running automated tests to detect integration errors as quickly as possible. |
| Coverage Metrics | Metrics used to evaluate how thoroughly a design has been tested or verified. |
| CPU (Central Processing Unit) | A CPU often referred to as the "brain" of a computer, is the primary component responsible for executing instructions and processing data. It runs the operating system and applications, constantly receiving input from the user or active software programs, processing the data, and producing output, which may be stored by an application or displayed on the screen. |
| Debug | Debugging is the process of finding and resolving defects or problems within a computer program or electronic system that prevent it from operating correctly.<br>→ Bug |
| Development software | This includes tools and environments that programmers use to create new software, such as compilers, debuggers and integrated development environments (IDEs) and EDA tools. |
| DFR (Design for Reliability) | Design practices aimed at ensuring long-term reliability of electronic systems. |
| DFT (Design for Testability) | Design techniques that improve the ability to test a chip after manufacturing. |
| DSP (Digital Signal Processing) | The use of digital computation to process signals such as audio, video, or sensor data. |
| DSP (Digital Signal Processor) | is a specialized microprocessor designed specifically for the efficient processing of digital signals. Digital signal processing involves the manipulation of signals that have been converted from analog to digital form, allowing for various operations such as filtering, compression, and modulation. DSPs are optimized to perform mathematical functions like addition, subtraction, multiplication, and division at high speeds with minimal energy consumption. |

| Term | Definition |
|---|---|
| EDA (Electronic Design Automation) | Software tools used to design and verify electronic systems such as ICs and PCBs. |
| EDA Toolbox | A set of tools and models for building domain-specific design and deployment flows. |
| Firmware | Firmware is a special type of software that is permanently programmed into the hardware of a device to control its functions. It is often located in ROM (read-only memory) or flash memory and is rarely or never changed. Examples of firmware are the BIOS of a computer or the control software of a printer. |
| FlexRay | A high-speed, deterministic, and fault-tolerant automotive network communication protocol. |
| FMEA (Failure Mode and Effects Analysis) | A structured approach to identifying and mitigating potential failure modes in a system. |
| Formal Verification | A mathematical method to prove the correctness of a system's design. |
| FPGA (Field-Programmable Gate Array) | A reconfigurable integrated circuit used for prototyping or specialized computing. |
| GDSII (Graphic Data System II) | A file format used to represent the physical layout of an integrated circuit. |
| GPU (Graphics Processing Unit) | is a specialized electronic circuit designed to accelerate the processing of images and graphics. GPUs are highly efficient at handling parallel tasks, making them ideal for rendering 2D and 3D graphics, performing complex mathematical calculations, and processing large amounts of data simultaneously. |
| HDL (Hardware Description Language) | It is a special programming language that is used to describe and model the structure and behaviour of electronic circuits. HDLs are mainly used in the development and verification of digital systems such as integrated circuits (ICs) and field-programmable gate arrays (FPGAs). |
| High-Level Synthesis (HLS) | A design process that converts algorithmic descriptions into hardware descriptions. |
| HV (High Voltage) | Refers to electronic systems operating at voltages higher than standard logic levels, e.g., 48V in automotive. |
| HW (Hardware) | Physical components and devices that make up electronic systems and circuits. These include, among others: →Integrated Circuits (ICs), →Application-Specific Integrated Circuits (ASICs), →Field-Programmable Gate Arrays (FPGAs), →Analog/RF-Design |
| IC (Integrated Circuits) | These consist of many electronic components such as transistors, resistors, and capacitors, which are integrated onto a single chip. |

| Term | Definition |
|---|---|
| IoT (Internet of Things) | refers to the network of physical objects embedded with sensors, software, and other technologies to connect and exchange data with other devices and systems over the internet. These "things" can range from ordinary household items to sophisticated industrial tools. The primary goal of IoT is to create a seamless and integrated environment where devices can communicate and interact with each other to improve efficiency, convenience, and automation. |
| IP (Intellectual Property) | Reusable design components or blocks used in chip development. |
| IREE (Intermediate Representation Execution Environment) | An open-source compiler and runtime for machine learning models. |
| ISO 26262 | An international standard for functional safety in automotive electronic systems. |
| ITRS (International Technology Roadmap for Semiconductors) | The International Technology Roadmap for Semiconductors (ITRS) was a set of documents produced by a group of leading semiconductor manufacturers and research organizations. The ITRS outlined the technological challenges and opportunities facing the semiconductor industry and provided a roadmap for future developments. It aimed to guide the industry in achieving continuous improvements in semiconductor technology, including advancements in materials, processes, and design methodologies. |
| LIN (Local Interconnect Network) | A low-cost serial communication protocol used in automotive networks. |
| LLM (Large Language Model) | A type of AI model trained on large text datasets, used for tasks like code generation and documentation. |
| MCU (Microcontroller Unit) | is an intelligent semiconductor integrated circuit (IC) that consists of a processor, memory modules, communication interfaces, and peripherals. MCUs are the core components in embedded systems, making up the circuit and enabling the system to interact with the physical world. |
| Middleware | This software serves as an intermediary between different applications or between applications and the operating system to facilitate communication and data management. |
| Mission Profile | A description of the expected operating conditions and stresses a product will experience during its lifecycle. |
| MLIR (Multi-Level Intermediate Representation) | A compiler infrastructure for building reusable and extensible compiler components. |
| NoC (Network-on-Chip) | A communication subsystem on a chip that connects various IP blocks. |

| Term | Definition |
|---|---|
| NPU (Neural Processing Unit) | is a specialized hardware accelerator designed to enhance the performance of artificial intelligence (AI) and machine learning tasks. NPUs are optimized for the parallel processing of neural network operations, making them highly efficient for tasks such as image and speech recognition, natural language processing, and other AI-related applications. |
| Packaging | refers to the process of designing and producing enclosures for electronic devices, ranging from individual semiconductor devices to complete systems. The primary functions of electronic packaging include interconnection, powering, cooling, and protecting semiconductor chips to ensure reliable operation of the system. |
| PCI (Peripheral Component Interconnect) | A hardware bus used for adding internal components to a desktop computer. |
| PDK (Process Design Kit) | A set of files provided by a semiconductor foundry that describes the manufacturing process. |
| PHY (Physical Layer) | The physical layer of the OSI model, responsible for the transmission and reception of raw bitstreams. |
| PPA (Power, Performance, Area) | Key metrics used to evaluate the efficiency and effectiveness of a chip design. |
| RISC-V | An open standard instruction set architecture (ISA) based on established reduced instruction set computing principles. |
| RTL (Register Transfer Level) | A design abstraction that describes the flow of data between registers in a digital circuit. |
| Scan Test | A method used in → DFT to test digital circuits by shifting test data into scan chains. |
| SDV (Software-Defined Vehicle) | A vehicle architecture where functionality is primarily defined and updated via software. |
| SoC (System on Chip) | is an integrated circuit that consolidates most or all components of a computer or electronic system onto a single microchip. This high level of integration typically includes a central processing unit (→ CPU), memory, input/output interfaces, and data storage control functions. Additionally, SoCs may incorporate optional features such as a graphics processing unit (→ GPU), Wi-Fi connectivity, and radio frequency processing. |
| Streaming Architecture | A system design paradigm optimized for continuous data flow, often used in signal processing and □ AI. |

| Term | Definition |
|---|---|
| SW (Software) | The programmes and operating systems that run on a computer or other electronic device. Unlike hardware, which comprises the physical components of a system, software consists of the instructions and data that control the hardware and perform various tasks. There are different types of software, including: → System software, →Application software, →Middleware, →Development software, and →Firmware |
| System software | This includes operating systems such as Windows, macOS or Linux, which control the basic functions of a computer and enable the execution of application software. |
| SystemC | A C++ library and simulation kernel used for system-level modelling and design. |
| SystemC TLM (Transaction-Level Modelling) | A high-level modelling approach in SystemC for simulating communication between components. |
| Technology Migration | The process of adapting a chip design to a different semiconductor manufacturing process. |
| Test Collaterals | Artifacts such as testbenches, scripts, and models used to support verification and validation. |
| Toolchain | A set of programming tools used to perform complex software or hardware development tasks. |
| Toolchain Customization | The ability to adapt and extend development tools to specific hardware or software needs. |
| TPU (Tensor Processing Unit) | is an application-specific integrated circuit (ASIC) developed by Google specifically for accelerating machine learning tasks, particularly those involving neural networks. TPUs are designed to handle the high volume of low-precision computations required for training and running machine learning models, making them highly efficient for these tasks. |

CHIP
DESIGN
GERMANY

AG Roadmap

Chipdesign Germany
Project coordination:

edacentrum GmbH
Schneiderberg 32
30167 Hannover
Germany

info@chipdesign-germany.de
www.chipdesign-germany.de

Working Group Office:
roadmap@chipdesign-germany.de