Evaluating an open-source hardware approach from HDL to GDS for a security chip design — a review of the final stage of project HEP

Tim Henkes, Steffen Reith, Marc Stöttinger
FB Design Computer Science Media
RheinMain University of Applied Sciences
Wiesbaden, Germany
tim.henkes | steffen.reith | marc.stoettinger@hs-rm.de

Norbert Herfurth, Goran Panic
IHP - Leibniz-Institut für innovative Mikroelektronik
Frankfurt (Oder), Germany
herfurth | panic@ihp-microelectronics.com

Julian Wälde
ACE
Fraunhofer SIT
Darmstadt, Germany
julian.waelde@sit.fraunhofer.de

Fabian Buschkowski, Pascal Sasdrich
Chair for Security Engineering
Ruhr-University Bochum
Bochum, Germany
fabian.buschkowski | pascal.sasdrich@rub.de

Christoph Lüth, Milan Funck
Deutsches Forschungszentrum
für Künstliche Intelligenz
Bremen, Germany
christoph.lueth | milan.funck@dfki.de

Tuba Kiyan
Security in Telecommunications
Technische Universitaet Berlin
Berlin, Germany
tuba.kiyan@tu-berlin.de

Arnd Weber
Bollschweil, Germany
arnd.weber@alumni.kit.edu

Detlef Boeck
Elektrobit Automotive GmbH
Erlangen, Germany
detlef.boeck@elektrobit.com

René Rathfelder
Software Powertrain
IAV GmbH
Berlin, Germany
rene.rathfelder@iav.de

Torsten Grawunder
Swissbit Germany AG
Berlin, Germany
torsten.grawunder@swissbit.com

***Abstract*—The project "*H*ardening the value chain through open-source, trustworthy *E*DA tools and *p*rocessors (HEP)" uses open-source, free components and tools for the production of a prototypical security chip. A design flow using only free and open tools from the abstract description in SpinalHDL via OpenROAD down to the GDS-file for tape-out has been established, and first ASICs produced at IHP. The prototypical hardware security module (HSM) produced in this way provides, among other things, a processor based on VexRiscv, a cryptographic accelerator and masking of cryptographic keys. The open development tools used in the process were integrated into a common environment and expanded to include missing functionality. Subsequently, the whole tool chain and its peripherals are wrapped into a new Nyx container. The easy accessibility of the used process significantly reduces the learning curve for chip design. Additionally, we provide tools for formal verification and masking against side-channel attacks in our design flow. Interest in the results of project HEP has been shown in publications in which industrial partners participated, such as Elektrobit, Hensoldt Cyber, IAV, Secure-IC and Swissbit Germany.**

***Index Terms*—open-source, hardware, side-channel, formal verification, RISC-V, PDK, open EDA, security, HSM**

This research has been supported by the German Federal Ministry of Education and Research (BMBF) as part of the ZEUS programme in the Leitinitiative *Trustworthy Electronics*.

I. Introduction

The open-source approach to digital hardware design is relatively new. Current industrial practice follows a closed business model, with information about the hardware design process available only under NDAs and protected by patents. This contrasts sharply with software development, where is has become clear that a free operating system is an advantage, given that suitable development tools (*e.g.*, compilers and editors) are available. By access to the source code, entire generations of developers coming through universities, schools or self-taught, acquired the necessary know-how to drive new developments forward. The advantage of this development model is that knowledge and development costs can be shared, in order to tackle large projects together.

Almost 40 years after the founding of the GNU project and the start of the Free Software Foundation, it is becoming clear that a similar approach is feasible in the field of hardware development. The first steps have been free development tools for programming commercial FPGAs and their extension, such as NextPNR and Yosys. At the same time, the open Instruction Set Architecture (ISA) RISC-V has been developed. It has given rise to a large number of RISC-V implementations (of

varying quality, size and efficiency), available under open-source licences. There are plenty of examples of the adoption of the RISC-V ISA in industry:

- Western Digital is developing the RISC-V implementation SweRV [1], integrating it into their products, and even giving their developments back to the community.
- SiFive licences IP for high-end RISC-V applications to Renesas [2].
- OpenTitan [3] plans to build an open-source silicon root of trust, as well as, more recently, the Caliptra initiative [4], using the SweRV/Veer design [1].

However, all these examples are carried out by bigger companies with an established design flow, most likely using commercial EDA tools. While this adoption of the open RISC-V ISA in the industry is welcome, the wider adoption of open hardware is blocked by enormous licence costs of EDA tools. Yet, this wider adoption is necessary to stimulate effects similar to those observed over the last 40 years in the software world.

This is where the HEP project is setting a new milestone in the domain of open-source hardware. The goal of the project is to design a real use-case driven ASIC, based on an open-source RISC-V design, with an EDA tool-flow that utilizes open-source tools wherever it is possible, from the definition in a high-level hardware description language down to the fabrication at IHP's MPW and prototyping service. Furthermore, formal and functional verification as well as the semi-automated implementation against selected side-channel attacks are integrated to point the direction into an industrial adoption for the evaluated fully open-source approach. Thus, the project HEP significantly expands the practical application of open-source frameworks in the domain of secure embedded integrated circuit design. This paper reports the achievements and contributions from the HEP project from both design and tooling perspectives, highlighting its substantial impact on pushing forward the frontiers of what is possible with open-source hardware development and therefore its contribution to the rising EU open hardware ecosystem.

II. Use Case and Demonstrator

Although secure communication is a fundamental requirement in IT security, the current landscape lacks a standard for implementing secure communication and ensuring the secure implementation of cryptography and key handling. Consequently, companies find themselves compelled to develop unique solutions to address these challenges. Hardware security modules (HSMs) are specialized hardware devices designed to provide such a solution.

HSMs provide a secure environment for key generation, storage, and cryptographic operations. Notable implementations of HSMs include Microsoft's *Pluton*, AMD's *PSP*, and Google's *Titan*. As the demand for secure environments in applications continues to rise, there is a simultaneous increase in the demand for HSMs. Since 2023, efforts by the Chips Alliance to develop the standardized HSM framework *Caliptra* signify a move towards establishing industry-wide standards.

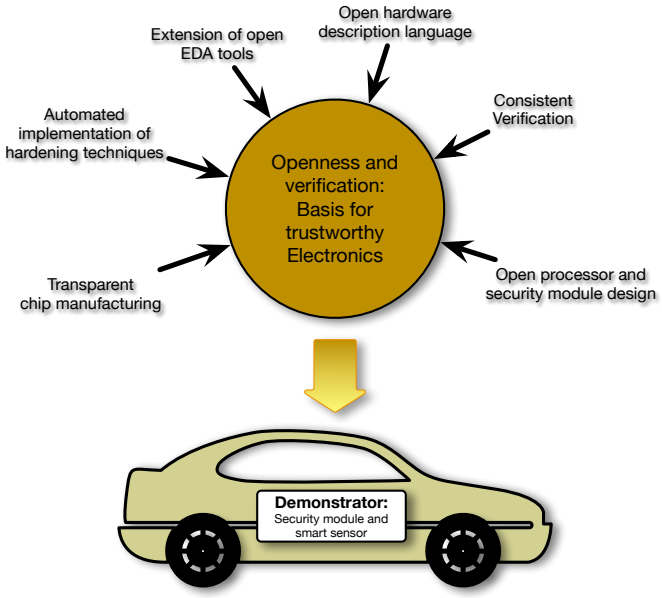

Fig. 1. The HEP demonstrator

For these reasons, the HEP project chose an HSM as its primary industrial use case, to demonstrate that a realistic, industrially viable hardware module can be developed using open-source technology. The main application area for HEP is the automotive industry (see Fig. 1), but as requirements in other application domains, *e.g.* memory controllers, might differ, we aim for a flexible and adaptable implementation, ensuring a robust and tailored approach to security. To improve the quality of the HSM, formal verification as well as hardening against side-channel attacks have to be implemented. The project partner have published a requirements report [5]. Ongoing developments, including the introduction of Caliptra and collaborations with Google/Skywater, align with the HEP project's foundational ideas and practical objectives.

III. Toolchain and Design

While the HEP project started in March 2021, the digital open-source tools of OpenROAD as well as Google's Sky 130 process design kit (PDK) had already been released. Back then, Skywater's multi-project-wafer (MPW) service was not that well established. On the other hand, the European-based digital tool flow *Coriolis* was not as prominent as it is today. Therefore, the HEP consortium decided to rely on the OpenROAD-based OpenLane flow in combination with the MPW service offered by the Leibniz-Institute IHP, which has already been established for decades. As the HEP project started, no Europen Open Source PDK was available. Therefore the project integrate IHP's proprietary PDK SG13G2 into OpenLane and in parallel initiated the developments for the IHP-Open130-G2, the first European open-source PDK (https://github.com/IHP-GmbH/IHP-Open-PDK).

On the design side, we selected the RISC-V core VexRiscv as the best option to realize the HEP goals. VexRiscv is

based on a relatively new meta hardware description language, SpinalHDL. The decision to opt for SpinalHDL over SystemVerilog was primarily driven by the lack of stable tooling for SystemVerilog with open-source hardware synthesis tools such as Yosys. While there are additional arguments in favour of SpinalHDL, including its user-friendliness and the accolade of VexRiscv as the winner of the RISC-V soft-core CPU contest, a key consideration is the ability to leverage the expressiveness of the programming language Scala, as SpinalHDL is written in Scala.

The utilized tools can be summarised as follows:

1) Specification and Architecture: The design is based on the VexRiscv core, establishing the initial specifications and architectural foundation.
2) RTL Design and Verification: The core is configured and extended in SpinalHDL, along with the usage of Verilog black boxes. Test benches in SpinalHDL are employed for RTL-level design testing.
3) Simulation: Verilator is used for simulation, validating the design at the Register-Transfer level.
4) FPGA Prototype: The full design is tested on an FPGA, ensuring functionality and performance in a hardware prototype environment.
5) ASIC Configuration: Minimal modifications between FPGA and ASIC configurations are required, involving a switch between the generated GDSII SRAM cells for the ASIC and the available RAM blocks on the FPGA.
6) Transpilation and Processing to GDSII: The design is first transpiled from SpinalHDL to Verilog, and then further processed to GDSII using the OpenLane flow. To insert a scan chain, commercial EDA tools have to be used in an intermediate step of the OpenLane flow.
7) PDK Configuration: The OpenLane flow is configured to use KLayout due to the lack of a PDK configuration for Magic.
8) Padframe Generation: Padframe generation is manually added to the OpenLane flow, leveraging a padframe generator in OpenROAD.
9) Conclusive Validation Steps: For DRC and LVS, proprietary tools are employed due to the absence of DRC and LVS configuration for KLayout in the PDK.
10) Metal Filling: Metal filling, not available in OpenLane, is performed by the fab using proprietary tools before production. Open-source tooling for metal filling is planned for incorporation in the next tape-out.

As a first result, the interoperability between open-source tools and commercial EDA tools works seamlessly. Furthermore, the proprietary DRC and LVS check on the GDSII file generated using the open-source tools did not reveal any failures.

IV. Tape-Out

The goal of the first two tape-outs was to design a working ASIC with an open-source design and open-source tools with a proprietary PDK. Therefore, the design was neither optimized for area nor speed. As such, the numbers presented in the following will not hold up to comparison with any optimized design, regardless of the toolchain used.

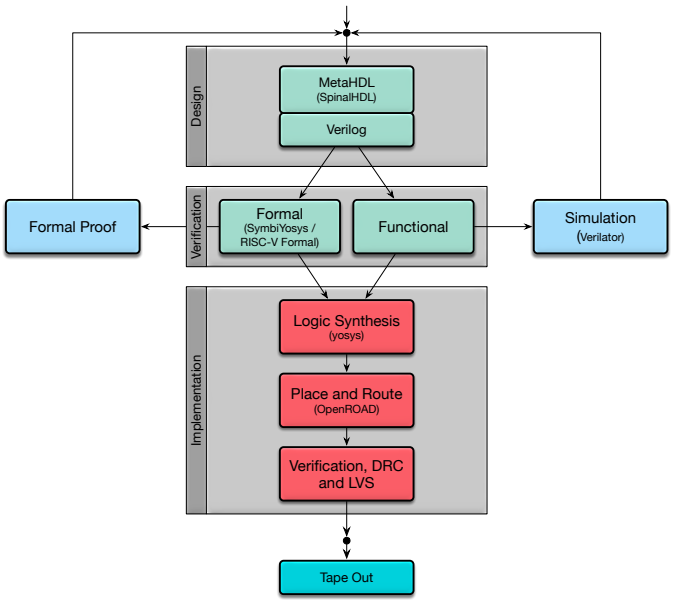

Fig. 2. HEP tool flow

A major obstacle in the timing and area optimization process was a bug in OpenLane or one of its components with our SRAM cell that made it necessary to disable the resizer — an important part of OpenLane that selects standard cell sizes to optimize for both area and timing at various points in the flow. Thus, we decided to target a rather low, fixed frequency until the bug is fixed or can be avoided. The core specifications are as follows:

- Area: 11.5mm²; Vex: 1.3mm²; SRAM: 10.2mm² (2nd TO: 13.7mm²/2.9mm²/10.2mm²)
- SRAM: 256kB RAM (8x32kB)
- Target frequency: 25MHz
- Peripherals: UART, SPI, JTAG, GPIO (2nd TO)
- AES accelerator on the ABP3 bus, masked AES (2nd TO)
- Implements the RV32I ISA with multiplication extension (excluding hardware division)
- Multiply&accumulate custom instructions for big integer arithmetic as part of a MulPlugin that provides 32x32 integer multiplication

Non-volatile memory and a trustworthy random number generator have not been integrated because no suitable open-source components are available; the project aims at designing these in the future.

This significant amount of SRAM is required because in contrast to common microcontroller setups where the firmware is often held in external non-volatile memory, our design is built to hold the firmware in SRAM as well. Only about 1.3 mm² of that area is taken up by the VexRiscv and its extensions. The HEP ASIC architecture is schematically shown in Fig. 3.

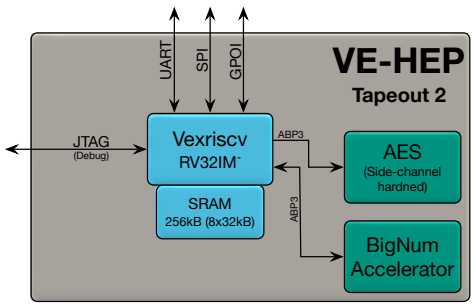

Fig. 3. Architecture of the HEP chip

V. Increased Security by Transparency

Being capable of fully looking into a system design enables novel approaches and transparent verification. It becomes possible to implement robust defences against side-channel attacks seamlessly. The inherent transparency in the hardening process allows a thorough examination of security gaps, ensuring a comprehensive and effective approach to system security.

A. Formal Verification

Formal verification is a crucial part of this project, ensuring the correct functionality of the produced chip. Moreover, formal verification works well with the open-source nature of the project: if the source code of the chip is available, we can formally verify it. In general, we have adopted the standard approach to formal verification in HEP, using assertions at the SpinalHDL level (see Fig. 2), which are translated down to Verilog and passed to the SymbiYosys tool for bounded model checking using SMT provers such as Z3.

To establish this tool flow, SpinalHDL had to be extended so that assertions formulated in SpinalHDL could be passed down to the Verilog level (initially, we had to formulate the assertions directly in Verilog). A more challenging task was the verification of the pipeline of the VexRiscv processor, which required an induction on the length of the pipeline and a precise definition of valid processor states (*i.e.*, those which can possibly occur while opcodes are passed through the pipeline).

Our verification is based on RISC-V formal [6]. By expanding this approach and implementing a reference formalization of the RISC-V ISA in SpinalHDL, we improved the quality, reusability and depth of the proof. We identified several corner case bugs and architecture gaps (some previously known) that have since been addressed. Furthermore, leveraging our experiences in this and other verification projects, we developed a methodology to systematically create sets of mutation classes. These classes serve to generate mutations tailored to the specific challenges of formal verification. They complement a randomized set of mutations to validate and compare the coverage of formal verification frameworks such as RISC-V formal (ensuring the the accurate classification of mutants as either rejected or accepted) [7]. These results will prove beneficial to other projects utilizing the RISC-V architecture, even if they do not use our toolchain.

Another challenge was to verify the functional correctness of the masked AES. The state space of the AES algorithm, when viewed as a black box, is too vast to handle formally. Thus, we had to split up the implementation into various intermediate steps, each of which is formally verified separately. Through parameterization, we acquire a toolbox capable of generating and verifying various configurations of AES algorithms.

B. Hardening against Side-Channel Attacks

Side-channel attacks have been a major threat to otherwise secure cryptographic implementations ever since their introduction by Kocher et al. [8]. By observing the physical behaviour of a chip, *e.g.*, its runtime or power consumption, an attacker may gain information on sensitive data such as the used cryptographic keys. To protect implementations against this type of attack, several countermeasures have been proposed, of which masking [9] is one of the most popular ones. The core idea of masking is to make the power consumption of an implementation independent of secrets by splitting secret data into multiple randomized parts, so-called shares. While this approach is proved to protect against side-channel attacks, applying it requires modifying existing cryptographic implementations. This task is not only time-consuming but also error-prone when executed manually, even for experienced designers.

As a consequence, (semi-) automated tools that can assist designers with the task of masking cryptographic implementations are needed, both to reduce the time needed to create a secure implementation and to increase the quality of masked implementations. In the course of the project, we developed the tool EASIMask [10], which, on the input of any round-based algorithm written in SpinalHDL, is capable of automatically masking the input at an arbitrary order and outputting the masked design in VHDL or Verilog. While the applied masking techniques are known for years and have been successfully applied manually, EASIMask is the first semi-automated tool to that frees developers from the manual work. We chose SpinalHDL as an input language for our tool for its advantages for designers implementing cryptography, such as its higher level of abstraction compared to Verilog or SystemVerilog, and dedicated functionalities for counters and Finite State Machines. In addition, SpinalHDL can effortlessly be manipulated by automated tools as no parsing of the code is necessary. Instead, tools can operate directly on the objects of the internal data model by modifying or deleting existing objects, or creating new objects.

While we integrated an unprotected byte-serial AES design into the first produced chip of the project, we used EASIMask to protect the AES design and integrated the masked design into the second and third-produced chip. The performance of the automatically masked implementation is on par with that of handcrafted designs in terms of latency, area, and required fresh randomness. Before integrating the masked designs into the chips, we verified their security on an FPGA using

TVLA. For more details on EASIMask and the performed experiments, we refer to [10].

Once the protected chips are fabricated and shipped, we plan to perform side-channel experiments on these chips to verify that the applied masking is not only secure on an FPGA, but also on an ASIC. This is especially interesting as countermeasures in academia are usually only tested on FPGAs due to the high cost and time-complexity of producing an ASIC. Therefore, we hope to contribute to closing the gap between the rich knowledge about hardware security on FPGAs, and the rather limited knowledge regarding the security of ASICs. In addition to this, HEP's approach also pays off in the area of hardware security. On the one hand, the complete disclosure of HEP results and paths means that everything can be traced, further developed and reproduced with relatively little effort.

C. The Evaluation of Scan Chain Side Channel Attacks

We have evaluated Design for Test structures, with a particular focus on scan chains from a security standpoint. Throughout the HEP project, the intricate balance between Design for Test (DfT) and Design for Security (DfS) became evident. While the use of scan chains in the testing phase significantly reduces testing time, it introduces a trade-off by potentially compromising the device's security through the creation of a backdoor.

The scan chain essentially leverages existing sequential logic, transforming each flip-flop into its scan flip-flop (SFF) counterpart with the addition of a multiplexer. In normal mode, the sequential logic operates as usual; however, in test mode, it transforms into a configurable shift register, facilitating the insertion of a desired state into registers for functional testing. This capability allows the creation of snapshots of an Integrated Circuit's (IC) state at a specific clock cycle and provides access to the inner nodes of the IC, thereby enhancing testing capabilities.

If the scan chain is not properly protected, it can be exploited to retrieve keys from cryptographic cores [11], or, even more drastically, it can be used to reverse engineer a device and steal a company's intellectual properties (IPs) [12]. These attacks assume that access to the scan chain is not restricted, or the scan chain is not obfuscated. In order to counteract the scan chain side channel, access to the test infrastructures can be disabled by blowing the fuses on either side of the scan path. However, this is not a common practice since it prevents on-site debugging capabilities [13]. There are further advanced industrial application techniques such as scan compression or dynamic scrambling of SFFs which are shown to be broken [14]. Another prevalent practice in the industry is known as logic locking, as referenced in [15]. A locked scan chain is implemented to obscure the data within the scan chain, aiming to conceal the chip's functionality during the testing phase. The obfuscation of the scan path is achieved by inserting a specific number of key gates between the SFFs.

We have conducted initial experiments on an ASIC USB-to-SPI bridge produced by IHP and also on an Intel Cyclone IV FPGA where we have implemented a scan chain obfuscation to assess the vulnerabilities introduced by having a scan chain in the design. We investigated possible optical side-channel attack scenarios on scan chains. In this context, our achievements are as follows:

1) We introduce a novel approach to reverse engineer the physical positions of a scan chain's SFFs.
2) After pinpointing the locations of target registers on the chip, we can extract their sensitive data, even in scenarios where access to scan test mode is restricted or disabled.
3) We perform novel optical side-channel attacks based on optical probing against SFFs to break scan chain obfuscation for the first time in the literature.

In our HEP chip, we have designed a scan chain test infrastructure intending to investigate the vulnerabilities introduced by the presence of scan chain structures.

VI. Conclusion

In summary, the HEP project has demonstrated that an industrially relevant ASIC of realistic size and functionality can be developed using open-source designs and tools. We have pinpointed the functionalities still missing in the open-source design flow (DRC and LVS checks), and highlighted the benefits of open-source, in particular increased trustworthiness by formal verification, and increased security by hardening and masking. The project leveraged free components and tools to produce an HSM with features like a cryptographic accelerator and masking. Open development tools were integrated into a cohesive toolchain, adding missing functionalities where needed, to cover the range from hardware description languages to GDSII file input for fabrication. The toolchain involved OpenROAD and German-based PDK and MPW services, aligning with the development of first European open PDK IHP-Open130-G2.

The design utilized the VexRiscv core in SpinalHDL, tested at RTL-level with Verilator, validated on FPGA, and transpiled to GDSII using the OpenLane flow. Notably, SpinalHDL and VexRiscv were modified and improved. For an easy reproduction of the project results, the tool chain, firmware builder and more have been wrapped into a Nyx container and published on the projects git repository (https://github.com/HEP-Alliance/VE-HEP-HW-SW). Formal verification employed SymbiYosys to ensure the correctness and security of the design. The team also developed EASIMask, a tool for semi-automated masking of cryptographic implementations in SpinalHDL. The project contributed to closing the gap between FPGA and ASIC security knowledge, emphasizing low-barrier access to the topic of ASIC design. Additionally, the evaluation of scan chain side-channel attacks revealed vulnerabilities and introduced novel optical probing techniques, addressing the intricate balance between DfT and DfS.

Despite facing challenges, including a bug in OpenLane affecting timing and area optimization, the project successfully taped out working ASICs in about 1.5 years. The project showcases the potential of open-source hardware in security

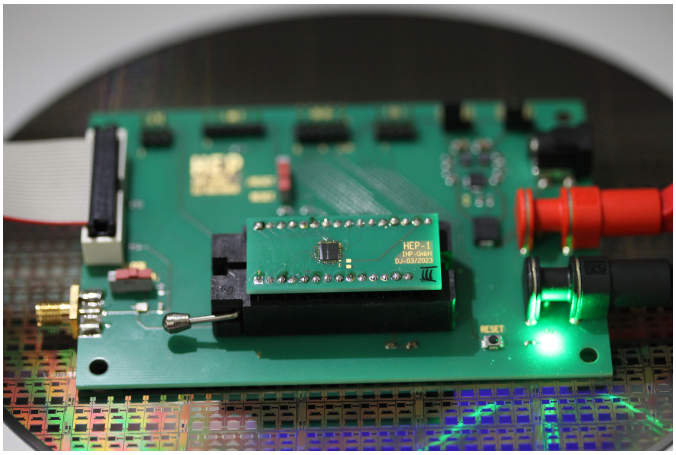

Fig. 4. The HEP chip: ASIC produced by open tools from HDL to GDS II, mounted on a custom-made board.

applications, with a focus on formal verification and hardening against side-channel attacks.

The project's demonstrator is an HSM targeted primarily at the automotive industry, but with an emphasis on flexible design to allow adaption to other application areas as well. The project's impact is evident in the interest it has garnered from industrial partners, including Elektrobit, Hensoldt Cyber, IAV, Secure-IC, and Swissbit Germany. Collaborations have extended to modifying a crypto driver for AUTOSAR to manage a hardware security module. Recognition from the RISC-V community, as evidenced by RISC-V International republishing the project's press release [16], underscores its significance. Alongside partners from CEA-List, LibreSilicon, and UNSW, the project advocates for open, well-verified, and ultimately provably secure components to bolster sovereignty, justifying continued support for both private and public research initiatives [17], [18].

VII. Future Work

In the short term, we aim to finalize the integration of the HSM ASICs into the industrial demonstrator setup. Ongoing efforts include replacing missing open-source components with suitable alternatives. This includes the open-source realisation of scan chain insertion, pad frame generation and metal filling, as well as design rule checks and layout-vs-schematic. Most of this functionality is already available as part of OpenROAD and KLayout, with the exception of scan chain insertion. Furthermore, a fully open-source design tape out with the updated IHP-Open130-G2 PDK is planned. A crucial aspect slated for development is a fully European open-source SRAM generator, expected to see a first draft by late 2024.

In terms of industrial application, the project is exploring integration with the Caliptra framework developed by the Chips Alliance. This framework aims to become a standard for incorporating HSMs, making the integration of the HEP chip a promising approach for industrial applications.

References

[1] https://blog.westerndigital.com/risc-v-swerv-core-open-source/.

[2] https://www.renesas.com/us/en/about/press-room/renesas-and-sifive-partner-jointly-develop-next-generation-high-end-risc-v-solutions-automotive, 2021.

[3] S. Johnson, D. Rizzo, P. Ranganathan, J. McCune, and R. Ho, "Titan: enabling a transparent silicon root of trust for cloud," in *Hot Chips: A Symposium on High Performance Chips*, vol. 194, 2018.

[4] B. Kelly, A. Lagar-Cavilla, J. Andersen, P. Jayana, P. Kwidzinski, R. Strong, J. Traver, L. Ferraro, I. Agarwal, A. Parthasarathy, B. Pillilli, V. Soni, M. Schilder, S. Mathane, N. Nadarajah, and K. Nielsen, "Caliptra: A datacenter system on a chip (soc) root of trust (rot)," Open Compute Project, Tech. Rep., 2022.

[5] M. Böhner, F. Buschkowski, M. Funck, T. Henkes, V. Herdt, N. Herfurth, T. Kiyan, N. Lahr, C. Lüth, R. Rathfelder, S. Reith, P. Sasdrich, M. Ulbricht, J. Wälde, and A. Weber, "Requirements analysis for an HSM, EDA tools and a demonstrator setup," http://hep-alliance.org/, 2021.

[6] "RISC-V formal verification framework," https://github.com/YosysHQ/, 2020.

[7] M. Funck, S. Ahmadi-Pour, V. Herdt, and R. Drechsler, "Identification of ISA-level mutation-classes for qualification of RISC-V formal verification," in *Forum on Specification & Design Languages (FDL 2023)*, 2023.

[8] P. C. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," in *CRYPTO 1996*, 1996.

[9] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi, "Towards Sound Approaches to Counteract Power-Analysis Attacks," in *CRYPTO 1999*, 1999.

[10] F. Buschkowski, P. Sasdrich, and T. Güneysu, "EASIMask - Towards Efficient, Automated, and Secure Implementation of Masking in Hardware," in *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2023, pp. 1–6.

[11] B. Yang, K. Wu, and R. Karri, "Scan based side channel attack on data encryption standard," Cryptology ePrint Archive, Paper 2004/083, 2004, https://eprint.iacr.org/2004/083. [Online]. Available: https://eprint.iacr.org/2004/083

[12] L. Azriel, R. Ginosar, and A. Mendelson, "Revealing on-chip proprietary security functions with scan side channel based reverse engineering," in *Proceedings of the on Great Lakes Symposium on VLSI 2017*, ser. GLSVLSI '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 233–238. [Online]. Available: https://doi.org/10.1145/3060403.3060464

[13] N. Limaye, C. Wachsmann, M. Nabeel, M. Ashraf, A. Kanuparthi, and O. Sinanoglu, "Antidote: Protecting debug against outsourced test entities," *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 3, pp. 1507–1518, 2022.

[14] J. Da Rolt, G. Di Natale, M.-L. Flottes, and B. Rouzeyre, "Are advanced dft structures sufficient for preventing scan-attacks?" in *2012 IEEE 30th VLSI Test Symposium (VTS)*, 2012, pp. 246–251.

[15] J. A. Roy, F. Koushanfar, and I. L. Markov, "Epic: Ending piracy of integrated circuits," in *Proceedings of the Conference on Design, Automation and Test in Europe*, ser. DATE '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 1069–1074. [Online]. Available: https://doi.org/10.1145/1403375.1403631

[16] https://riscv.org/news/2023/10/research-consortium-sets-standards-in-the-field-of-open-source-hardware-open-tools-used-for-a-security-chip/.

[17] A. Weber, S. Guilley, R. Rathfelder, M. Stöttinger, C. Lüth, M. Malenko, T. Grawunder, S. Reith, A. Puccetti, J.-P. Seifert, N. Herfurth, H. Sankowski, and G. Heiser, "Verified value chains, innovation and competition," in *2023 IEEE International Conference on Cyber Security and Resilience (CSR)*, 2023, pp. 470–476.

[18] https://www.elektrobit.com/products/ecu/eb-tresos/.