Integrated Design Flow Methodology for Open-Source Innovations in Smart Transportation: Empowering Accountable AI and Cybersecurity

Alper Kanak [1,2*] [0000-0003-2541-7753], Salih Ergün [1,2] [0000-0001-7070-9726], Ali Serdar[3] [0000-0002-9513-2481], Ahu Ece Hartavi Karcı[4] [0000-0003-4956-4651], and Baran Çürüklü[5] [0000-0002-5224-8302]

[1] Ergünler R&D Co. Ltd., Isparta, Turkiye
[2] Ergtech Sp.Z.o.o., Warszawa, Poland
[3] AI4SEC OÜ, Talinn, Estonia
[4] University of Surrey, Center of Automotive Engineering, Guildford, UK
[5]Mälardalen University, Division of Intelligent Future Technologies, Eskilstuna, Sweden

alper.kanak@{ergtech.eu, erarge.com.tr},salih.ergun@{ergtech.eu, erarge.com.tr}, ali.atalay@ai4sec.eu, a.hartavikarci@surrey.ac.uk, baran.curuklu@mdu.se.

Abstract. Spearheading the adoption of trustworthy AI and cyber security across the triad of automotive, transportation, and logistics embark on rigorous evaluations of applications rooted in open hardware/software ecosystems. Open innovations at chip, embedded and/or system level foster research on security-, safety-, privacy-, and accountability-by-design. Cyber-physical system of systems amplifies the fortress of AI-powered solutions and cyber-physical security, with particular emphasis on open-source frameworks. This paper presents an integrated design flow methodology that incorporates chip, embedded and system-level design and development. The proposed methodology is aligned with the recent trends and state-of-the-art dealing with open-source hardware and software development.

Keywords: Integrated Circuits (IC); Trustworthy AI; Accountability; Cyber Security; Smart Transportation; Open Source Innovations.

1 Introduction

As digital transformation is getting widespread, dependency on Information and Communication Technologies (ICT) has covered a wide range of technological advancements starting from chips to the cyber-physical system (CPS) of the systems level. The automotive, transportation and logistics sectors, as they are getting smarter and more connected, cooperative, and autonomous, cities are evolving to more digitised, AI- and IoT-powered. However, this transformation can be blocked due to serious problems like chip shortages, supply chain inefficiencies, multimodal transportation inadequacies, cyber-physical threats, accidents and safety challenges.

In this technology battlefield, open-source hardware and software (OSH and OSS) innovations present a game-changer strategy to reduce research, development, testing and consequently manufacturing or deploying costs and durations. For instance, RISC-V [1] had been promoted as a free and open-source computer instruction set architecture first introduced in 2010 and it has become a new trend in the open HW domain to develop secure processors and AI-on-chip solutions. Within the scope of emerging needs and trends of the semiconductor industry, the RISC-V open instruction set architecture (ISA) has been well-positioned, as a pioneering open innovation initiative, for the transportation, mobility and automotive market. Similarly, formalizing organizations' approach to OSS management and strategies to increase the uptake of open-source tools is increasing to boost the efficiency in development processes and minimize the risks of implementation and dependency on expensive solutions in the market.

This paper first aims to present an overview of the current status of the research and innovation strategies, like the European Chips Act, or global OSS strategies led by the Linux Foundation. Then, the paper will introduce a design flow methodology that can be used in CPS design starting from the chip level and covering the embedded level towards the implementation of a system of systems.

2 Current Status Update

The projected growth of the global automotive software (SW) and hardware (HW) market is about 6% CAGR from 2019 to 2030 and is expected to reach $460 billion [2]. This is parallel with the rising importance of open-source automotive SW and HW in the overall market. As vehicles become more advanced and integrated, the complexity of the systems involved also increases substantially. Managing this complexity and ensuring the seamless operation of numerous on-board systems, such as electronic control units (ECUs), as well as the transportation/mobility grid infrastructures in premium vehicles is a significant challenge faced by automakers. A well-designed Electrical/Electronic (E/E), or cyber-physical in general, architecture ensures efficient and trusted communication and coordination between different on-board and external systems. This is crucial for the proper functioning of advanced features like autonomous driving capabilities, infotainment systems, safety mechanisms, and recent third-wave AI solutions enabling the trustworthy, efficient and fair use of AI [3].

As evident from the chip shortages and the indispensable side-effects of the COVID-19 outbreak, the chip industry has faced with serious problems listed below that cannot be solved by self-contained and private investments of large semiconductor companies: i) Integration and Validation; ii) Budget Allocation; iii) Reliability and Safety; iv) Scalability; v) Cybersecurity and Trustworthiness; vi) Collaboration and Standards. Establishing industry standards, especially for OSS and OSH architecture promotes interoperability and facilitates the development of compatible components and systems. Navigating these challenges requires proactive supporting acts to encourage stakeholders to continuously adapt to new technologies, industry standards, AI-powered smart systems and cybersecurity practices to ensure the seamless integration of SW and HW in modern vehicles and transportation infrastructures. For instance, the European Chips

Act [4] identified RISC-V as one of the key disruptive technologies in which Europe should invest. The RISC-V strategy is expected to be a key market driver for the new generation Integrated Circuits developed for more digitised services in the automotive and logistic domain, i.e., connected and/or autonomous vehicles, communication infrastructures, SW-defined cars, connected and collaborative mobility and smart logistic services enhanced with System-on-Chip.

Similarly, formalizing an organization's approach to OSS management, and strategies to increase the uptake generated through open source will most likely boost the efficiency associated with product development. At the same time, it will minimise the risks of locking future developments into specific technologies that are not controlled internally. For example, The European Commission has already adopted the benefits of OSS by approving the new OSS Strategy 2020-2023[5].

Although the automotive sector is quite conservative to open-source solutions, a tremendous effort can be observed to utilise the OSH and OSS as evident from the recent advancements in autonomous driving and smart transportation and mobility. Recent approaches orient technology providers to open HW and open-source technologies, for instance, mixed-criticality multi-OS architecture for automotive ECUs [6] or open cooperative driving automation in transportation and mobility [7]. The development of commercial embedded AI and AI accelerators is continuously rising, but in exchange, they are very expensive. On the open-source and affordable HW side, championed by RISC-V, will bring forth a new generation of open-source chipsets designed for specific ML and DL applications at the edge. The High-Performance Embedded Architecture and Compilation (HiPEAC) Vision 2023 [8] report classifies the embedded architecture goals around "the race for" the "next web", AI, sovereignty, and sustainability.

The cyber-physical security is a paramount need in new-generation OSH [9] or OSS [10]. RISC-V-originated HW solutions are still open to HW or architecture-related vulnerabilities. From the HW perspective, the core RISC-V design carries some specific security improvements, in particular, relative to the long-established ARM RISC design, but is lagging in practical security technique implementations due to its competitive disadvantage in terms of R&D up-front investment compared to, for example, ARM implementations by commercial ARM developers [11]. This relates in particular to an incomplete cryptographic instruction set, access security, HW-assisted security extension units, instruction set architecture security extensions and the memory and side-channel attack protection of silicon implementations of the RISC-V design.

3 Proposed Design Flow Methodology

The paramount take-away message of both state-of-play and state-of-the-art is that the design and implementation of a CPS supported with OSS and OSS is based on a complex life cycle. To tackle the challenges addressed in Section 2, a three-step agile methodology is proposed for the industry so that the developers can formulate their work and assess and verify the design and implementation steps throughout the development life cycle (As depicted in **Fig. *1***).

Step-I: FPGA/ASIC/Hybrid Chip Design: Since ASIC design itself requires too much effort and time, and might be expensive, FPGA-based design can also be implemented in parallel for IC prototyping. The ASIC design process (digital chip) includes function description, module division, module code input, module-level simulation verification, system integration, and system simulation verification, synthesis, STA (static timing analysis), formal verification. Similarly, the complete FPGA design process includes functional description, circuit design and input, functional simulation, synthesis optimization, post-synthesis simulation, implementation and placement and routing, timing simulation, board-level simulation, verification, debugging, and loading configuration. FPGA verification is an important part of ASIC design. After that, it is necessary to introduce the ASIC version source code, insert IO PAD, DFT, power consumption estimation, and perform other back-end processes. It can be said that 50-80% of the entire ASIC process is completed by completing FPGA verification.

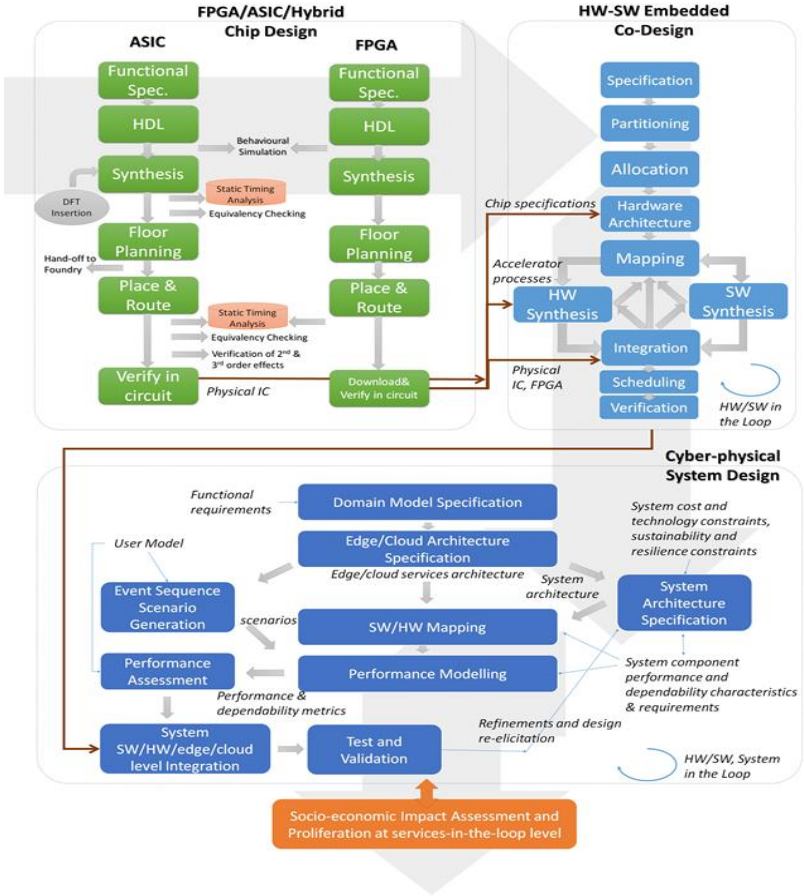

Fig. 1 Proposed Design Flow Methodology

Step-II: HW/SW Embedded Co-Design: The co-design methodology encounters the following steps to come up with an embedded device (such as telemetry devices, gateways, ECUs, Embedded AI devices, etc.): i) Specification: consists of a collection

of metrics, both functional and non-functional, which provide a precise description of the top-level system attributes and requirements. ii) Partitioning is the action of breaking the system functionality into small domain-independent, concurrent and interacting/communicating processes. iii) Allocation is the action of assigning each process to either the HW or SW domain. Communication bandwidth alternatives/limitations between HW and SW should be considered. iv) HW Architecture means describing what HW components should be used and how they should be connected to support the execution of the processes. v) Mapping is the selection of specific HW components and mapping the processes onto parts of the HW architecture. vi) Synthesis is the implementation of the HW/SW processes for the selected HW. vii) Integration is the recombination and testing of processes and interfaces after implementation. viii) Scheduling is the assignment of resources to all system processes such that their execution requirements are satisfied including interprocess communication dependencies.

Step-III CPS Design: This approach is useful to distinguish among a variety of alternatives, both good and bad, assess the impact of architectural choices, predict potential bottlenecks, and size of the HW components, and evaluate if a proposed architecture will meet the performance requirements under the expected workload. The main thrust of this iterative method is to ensure, by successive refinements, that the architecture meets the performance goals set forth in the requirements analysis and specification phase. There are five basic inputs to the method (shown in dark blue) functional requirements, user model, performance requirements, system cost and technology constraints, and system component performance.

The applied methodology considers the following steps: i) Domain model specification is developed to reflect the interaction among the main system components in order to satisfy the functional requirements. ii) Edge/Cloud architecture specification is used to derive a client/server and edge/cloud SW architecture, which depicts the message exchanges between nodes and services in the system. iii) System architecture specification is used to specify the method, the type, the number of components of each type, and the connectivity used to link them together. iv) Event sequence scenario generation: Event sequence scenarios are created by taking a user model, which provides a detailed description of the user interactions with the system mapping them to the edge/cloud architecture. v) Performance annotation of event sequence scenarios: The event sequence scenarios are further annotated with performance parameters such as request arrival rates, data volumes per request, server processing and I/O requirements per request. vi) HW/SW mapping: The components of the system architecture are assigned performance characteristics (e.g., network segment speeds, router latencies, I/O subsystem bandwidth, processor speeds). Then, the performance annotated scenarios, the HW/SW map, and the system architecture performance characteristics are combined to generate input parameters for a performance model. vii) Performance modelling: The outputs of the performance model include response times and throughputs for each type of request submitted to the system. An analysis of the results of the performance model reveals the possible bottlenecks. If the architecture does not meet the performance objectives, architectural changes at the HW and/or SW level have to take place. viii) Integration is implemented and SW, HW, edge/cloud and system level. ix) Test and validation of the CPS take place by checking the performance and dependability metrics.

4 Conclusion

This paper presents a three-step design flow methodology that can be used to develop FPGA-level or chip-level HW, embedded devices and system-level solutions for the CPSs specified for the automotive, transportation and logistic systems of systems. The paper emphasizes the uptake of the OSS and OSH to present quick and low-cost AI and cyber security solutions in the targeted domains. The proposed methodology has been conducted in two ongoing European projects, ESCALATE and OPEVA, which address the utilization of heavy-duty and lightweight connected electrical vehicles in intercity or urban logistics and fleet management, respectively. Note that this methodology can be adapted to any CPS. The technical details of this use case are left for further study.

Acknowledgement: This work has been supported by the European Union's Horizon Europe and Horizon Europe Key Digital Technologies Joint undertaking, namely ESCALATE and OPEVA projects under grant agreements No. 101096598 and 101097267, respectively.

References

1. RISC-V, Reduced Instruction Set Computer-V, https://riscv.org/
2. McKinsey, Getting ready for next-generation E/E architecture with zonal compute, June 14, 2023, Article, last accessed on July 4, 2023, https://www.mckinsey.com/industries/semiconductors/our-insights/getting-ready-for-next-generation-ee-architecture-with-zonal-compute
3. Kanak, A., Ergün, S., Atalay, A. S., Persi, S., & Karcı, A. E. H. (2022, October). A Review and Strategic Approach for the Transition towards Third-Wave Trustworthy and Explainable AI in Connected, Cooperative and Automated Mobility (CCAM). In 2022 27th Asia Pacific Conference on Communications (APCC) (pp. 108-113). IEEE.
4. Regulation (EU) 2023/1781 of the European Parliament and of the Council of 13 September 2023 establishing a framework of measures for strengthening Europe's semiconductor ecosystem and amending Regulation (EU) 2021/694 (Chips Act), Document 32023R1781, 2023.
5. Open source software strategy 2020-2023, November, 2020, last accessed on July 23, 2023 https://commission.europa.eu/about-european-commission/departments-and-executive-agencies/informatics/open-source-software-strategy_en
6. Cuomo, L., Scordino, C., Ottaviano, A., Wistoff, N., Balas, R., Benini, L., ... & Savino, I. M. (2023, June). Towards a RISC-V Open Platform for Next-generation Automotive ECUs. In 2023 12th Mediterranean Conference on Embedded Computing (MECO) (pp. 1-8). IEEE.
7. Xu, R., Xiang, H., Han, X., Xia, X., Meng, Z., Chen, C. J., ... & Ma, J. (2023). The opened open-source ecosystem for cooperative driving automation research. IEEE Transactions on Intelligent Vehicles.
8. https://www.hipeac.net/vision/2023.pdf
9. Baehr et.al. (2022, August). Open-source hardware design and hardware reverse engineering: a security analysis. In 2022 25th Euromicro Conference on Digital System Design (DSD) (pp. 504-512). IEEE.
10. Nisar et.al.. (2020). A survey on the architecture, application, and security of software-defined networking: Challenges and open issues. Internet of Things, 12, 100289.
11. Lu, T. (2021). A survey on RISC-V security: Hardware and architecture. arXiv preprint arXiv:2107.04175.