Integration of IP-Cores for the M^3 Architecture with Low Area Overhead: Accelerator Support Module

Friedrich Pauls, Sebastian Haas, Yogesh Verma
Barkhausen Institut, Dresden, Germany
forename.surname@barkhauseninstitut.org

***Abstract*—Smart digital systems are integral to sectors such as communications, power grids, and autonomous transport. As these technologies increasingly underpin critical societal functions, the need for robust, breach-resistant security architectures becomes paramount. Secure tiled architectures with stringent hardware isolation capabilities have been shown to have a minimized attack surface even when integration of third-party intellectual property (IP) cores is indicated. However, current platforms require costly adaption of IP cores to the security protocols these architectures provide. This paper introduces a new Accelerator Support Module that manages the protocol overhead of the underlying security architecture and offers commonly used hardware interfaces. This module simplifies the integration of accelerators and only slightly increases the area overhead.**

***Index Terms*—Hardware/Software Co-Design, Isolation, Network-on-Chip, Operating System, Privacy, Security, Tiled Architecture**

I. Introduction

The increasing integration of smart digital systems in sectors such as communications, power grids, and autonomous transport highlights the critical need for advanced security and privacy measures in digital architectures. Ensuring the infallibility and resistance to breaches of these technologies, which increasingly support societal functions, is essential. The secure architecture proposed recently, known as M^3, utilizes a micro-kernel operating system (OS) and hardware units called trusted communication units (TCUs) [1, 2]. This design aims to address security challenges in diverse applications such as edge computing and network slicing. The architecture focuses on low-latency, high-performance, and isolation capabilities necessary for safe and secure operations in multi-tenant environments.
The M^3 architecture features the integration of third-party intellectual property (IP) cores that are necessary to meet time-to-market demands. However, not all IP cores come with appropriate hardware/software (HW/SW) interfaces required by the TCU and the M^3 protocol. This paper proposes the accelerator support module (ASM) tile, a novel integration method for hardware accelerators into the M^3 system, that reduces the integration overhead for third-party IP with only minimal area overhead.

This paper is structured as follows. Section I-A introduces the architecture concept and the key challenge that comes with it. Thereafter, in section I-B, we refresh the concept of *isolation-by-default*, which deals with the security challenges. Section II provides a rational and architectural description of our novel ASM. Section III gives implementation details of our hardware platform and presents results. Section IV concludes the paper.

A. M^3 Architecture Concept

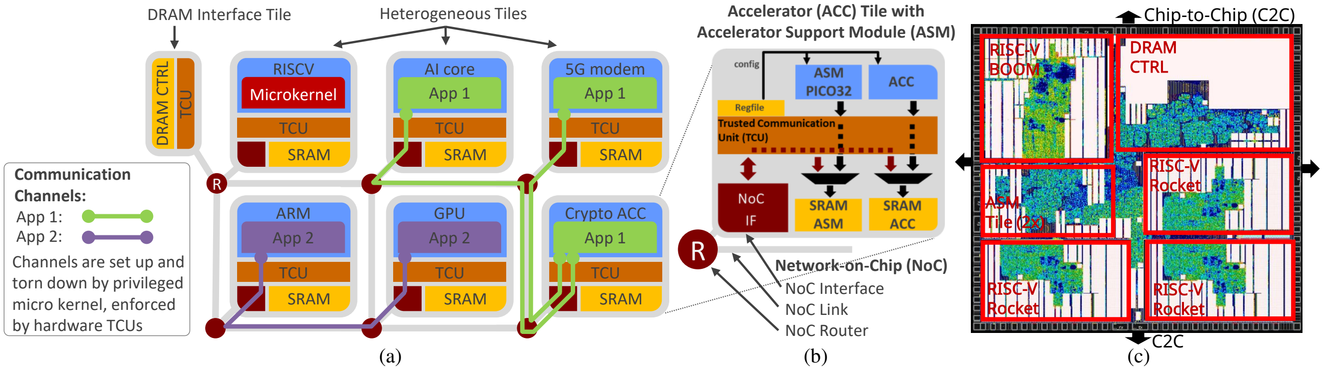
Our architecture integrates heterogeneous processing units into a single system-on-chip (SoC) to meet the performance demands of targeted applications. The platform employs a tiled architecture, which includes a variety of processing tiles such as general-purpose cores, specialized accelerators, and communication modems, each linked by a scalable Network-on-Chip (NoC) [3]. It can be tailored to a specific application scenario. Figure 1 (a) shows an example configuration. A significant challenge arises from integrating third-party intellectual property (IP) cores; although necessary, it increases the risk of security breaches. To tackle this problem, the platform follows the *isolation-by-default* paradigm, explained in the following section.

B. Isolation-by-Default

The cornerstone of our security strategy is the robust isolation of all system components, achieved through the deployment of Trusted Communication Units (TCUs) [1]. These units enforce strict access controls, ensuring that each processing tile can only interact with the system via regulated pathways. Our architecture follows the isolation-by-default principle, where no tile can inherently communicate with another, effectively minimizing potential security breaches. We use the open-source M^3 micro-kernel OS [4]. The OS establishes and tears down communication channels between all components that an application wants to use. Fig. 1 (a) shows two isolated applications (App 1, App 2) as an example. In this HW/SW co-design approach, the OS manages secure communication channels between system components. At the same time, the TCUs enforce these policies in HW, significantly reducing the trusted computing base and enhancing overall system security. Applications operate in discrete, secure environments, isolated from each other and potential threats, even if part of the system becomes compromised. Related security architectures like the NoC-centric SiFive Shield use memory protection units and message passing to create a tiled security architecture [5].

II. Accelerator Support Module (ASM) Tile

Specialized IP cores designed for one specific task (e.g., FFT or encryption) typically do not come with the required interfaces and logic to handle the communication primitives



(a) (b) (c)

Fig. 1: (a) General architecture secure HW/OS platform. (b) Architecture of Accelerator Support Module tile. (c) Post-routing layout of an instance of our secure platform: 3x RISC-V Rocket in-order core, 1x BOOM out-of-order core, 4x Chip-2-Chip IO tiles, 2x ASM Tile (AES, SHA3), 1xDRAM Tile, 22 nm FDSOI, 25 °C, 0.8 V, 100 MHz, DRAM 400 MHz, 11.24 mm^2.

required by the M^3 system. To ease this integration challenge, we present a modular ASM tile that reduces the barrier for system integrators to adopt our secure platform. Fig. 1 (b) shows its general architecture. The ASM tile employs the accelerator (ACC), which has access to local SRAM via a standard memory interface of up to 128-bit data width. Typically, ACCs are configured via configuration registers or memory-mapped IO. Both variants can be handled by the 64-bit configuration interface provided by the TCU. We use a size-optimized PicoRV32 RISC-V core as a protocol handler between M^3 kernel, the TCU, and the ACC. This allows for a flexible SW adaption process while keeping the area overhead minimal. The ASM can access SRAM regions for code and data via a dedicated memory interface. In addition, it can access the ACC configuration registers and possibly interrupt signals via the configuration interface provided by the TCU. The TCU supervises all entities' memory and configuration requests, thus enforcing all security properties and allowing IP integration with minimal security assumptions.

III. Implementation and Results

To evaluate the ASM area overhead, we implemented and synthesized an instance of an M^3 architecture in a 22 nm FDSoI GlobalFoundries process, typical conditions (25 °C, 0.8 V). The architecture consists of 12 tiles: 4xRISC-V tiles (3xRocket, 1xBOOM), 4xChip-2-Chip (C2C) IO tiles, a periphery IO tile, 2xASM Tiles, a DRAM Tile, and a 2x2 star-mesh NoC (bandwidth 16 bytes/cycle). The critical path lies in the DRAM controller, clocked with the required 400 MHz. All other components are synthesized at 100 MHz. All tiles are secured by a TCU.

Fig. 1 (c) shows the post-routing layout of our implementation. The total chip area is 11.24 mm^2 from which 3.84 mm^2 (52%) is SRAM. Areas for 3xRISC-V Rocket tiles, 1xRISC-V BOOM tile, 4xC2C tiles, and NoC are 4.08 mm^2, 1.81 mm^2, 0.67 mm^2, 0.07 mm^2, respectively. A single RISC-V Rocket tile has an area of 1.36 mm^2 from which 1.01 mm^2 is SRAM, which is mainly used as a cache. Within a single RISC-V tile, the areas for Rocket core, TCU, and NoC-IF are 0.87 mm^2, 0.34 mm^2, 0.14 mm^2, respectively.

Our implementation features two ASM tiles with example ACCs for encryption (AES) and hashing (SHA3) with tile areas of 0.42 mm^2, and 0.34 mm^2, respectively. Both tiles have a memory configuration of 4 kb ACC SRAM, and 64 kb/16 kb ASM SRAM for code/data for the ASM PicoRV32 protocol core. In the following, we focus on the smaller SHA3 core for a conservative estimate of the area overhead. For the ASM tile, the TCU doesn't require its full feature set (e.g., virtual memory, context switching, physical memory protection). Therefore, the ASM TCU (0.036 mm^2) is significantly smaller than the one in the RISC-V tiles (0.42 mm^2). About 42% of ASM area is used for memory, 11% for the TCU, and only 1% for the ASM PicoRV32 core. Even for a small accelerator, the security costs in terms of area overhead for TCU plus ASM relative to the total tile size is only about 11%.

IV. Conclusion

Our ASM architecture builds upon the existing adaptable HW platform with hardware TCUs that, together with a microkernel-based OS, effectively isolate applications from each other. Our presented ASM approach enables the secure integration of arbitrary accelerators with only a very low area overhead.

Acknowledgment

This research was partly funded by the German Federal Ministry of Education and Research under grant number 16ME0527 and the European Union's Horizon Europe research and innovation program under Grant Agreement No 101095759.

References

[1] S. Haas and N. Asmussen, "A trusted communication unit for secure tiled hardware architectures," in *2022 29th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2022, pp. 1–4.

[2] F. Pauls, S. Haas, and M. Hasler, "Trust-minimized integration of third-party intellectual property cores," in *2023 20th International SoC Design Conference (ISOCC)*, 2023, pp. 53–54.

[3] G. Fettweis *et al.*, "A Low-Power Scalable Signal Processing Chip Platform for 5G and Beyond - Kachel," in *53rd Asilomar Conference on Signals, Systems, and Computers*, 2019.

[4] N. Asmussen, M. Völp, B. Nöthen, H. Härtig, and G. Fettweis, "M3: A hardware/operating-system co-design to tame heterogeneous manycores," ser. ASPLOS'16. ACM, 2016, pp. 189–203.

[5] J. Prior, "SiFive Shield: An Open, Scalable Platform Architecture for Security," 2019. [Online]. Available: https://www.sifive.com/blog/sifive-shield-an-open-scalable-platform-architecture