NEUROPULS

Deliverable 6.1

Metrics Definition

Start date of the project: 1st January 2023

Duration 48 months

Funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them.

Document Classification

Document Title	D6.1 Metrics Definition
Author(s)	P04 – POLITO – Roberta Bardini, Stefano Di Carlo, Alfredo Benso, Paolo Prinetto, and Alessandro Savino P09 – NKUA – George Papadimitriou, Dimitris Gizopoulos P05 – INESC-ID – Luis Guerra y Silva P12 – UNIVR – Mariano Ceccato, Alberto Lovato, Niccolò Marastoni P1.1 – ECL – Alberto Bosio P15 – TUB – Jean-Pierre Seifert, Ulrich Ruhrmair P1 – CNRS – Fabio Pavanello
Work Package	WP6 – Benchmarking of the secure low-power system
Dissemination Level	PU = Public
Nature	R = report
Doc ID Code	24_03_29_NEUROPULS_D6.1.doc
Keywords	Metrics, computing architectures, physical unclonable functions, simulation

Document History

2023-08-03	Table of contents and structure defined	P4 POLITO – A. Savino
2024-02-28	First draft	All contributors

Document History

2024-03-29	Finalized	All contributors

Document Validation

Project Coordinator	Fabio Pavanello
Date	2024-03-29

This document contains information that is proprietary to the NEUROPULS consortium. The document or the content of it shall not be communicated by any means to any third party except with prior written approval of the NEUROPULS consortium.

Document Abstract

This report aims to comprehensively examine metrics for evaluating the NEUROPULS Horizon Europe project (Grant Agreement n° 101070238) accelerator and their simulation counterparts and the security of the phase-change material PUFs provide. The report provides an in-depth analysis of the key performance indicators, methodologies, and tools utilized in assessing the efficiency and efficacy of the revolutionary computing platforms proposed in the project. By defining a standardized set of metrics and evaluation practices, this document aims to foster cross-comparisons, facilitate advancements, and guide researchers, developers, and industry stakeholders in harnessing the true capabilities of neuromorphic computing.

This deliverable explores the challenges of benchmarking photonic chips, considering speed, power efficiency, and scalability factors. Additionally, we emphasize the importance of establishing a robust framework for comparing simulation results with physical implementations, enabling researchers and engineers to gain meaningful insights into the capabilities and limitations of these cutting-edge technologies.

Through thoroughly examining established evaluation metrics and emerging standards, this document aims to provide a comprehensive guide for researchers, developers, and industry professionals engaged in evaluating and benchmarking photonic chips and their simulation models. By addressing the unique challenges and opportunities in this field, we strive to contribute to the ongoing dialogue surrounding the advancement of photonic computing and foster a standardized approach to benchmarking that ensures meaningful and reliable assessments.

Table of contents

1. Introduction

Developing advanced hardware accelerators and supporting software tool ecosystems presents many challenges and opportunities in the dynamic and evolving landscape of neuromorphic computing. As part of our ongoing efforts in this domain, the NEUROPULS team has dedicated substantial resources towards defining comprehensive metrics that holistically evaluate the quality and performance of neuromorphic chips and the encompassing tool ecosystem.

1.1 Objectives

This deliverable outlines a set of metrics designed to gauge various aspects of the work, ensuring that our advancements not only meet but exceed the current standards of neuromorphic technology.

The essence of our initiative is to establish a robust framework for evaluation that spans across multiple dimensions of our project:

- **Chip-Related Metrics**: Recognizing the critical importance of hardware efficiency, we have developed metrics to assess the neuromorphic accelerator's performance, mainly focusing on its bandwidth, latency, and power consumption. These metrics are pivotal in understanding the chip's operational efficiency and potential impact on broader system performance.
- **Simulation-Related Metrics**: With an eye toward the future, our team has also focused on metrics that validate the functional likelihood of our models. These metrics are instrumental in assessing the scalability of our simulations, especially in the context of next-generation hardware that may present sizes and complexities not currently available. This foresight allows us to anticipate and address potential challenges in hardware evolution, ensuring our models remain relevant and adaptable.
- **Security-Related Metrics**: In an era where digital security is of paramount importance, securing architectures is essential. Our security-related metrics are designed to evaluate the integrity and robustness of the security features provided by Physical Unclonable Functions (PUFs) and the overall secure architecture of our envisioned system. These metrics offer a quantitative measure of security performance, essential for building trust and reliability in neuromorphic computing systems.

Integral to our evaluation framework is the inclusion of reference counterparts for all performance metrics. This addition is designed to facilitate future benchmarking efforts, providing a quantitative perspective that enhances the comparability and competitiveness of our neuromorphic solutions. Establishing these benchmarks lays the groundwork for continuous improvement and innovation in the field.

1.2 Scope and Limitations

Our evaluation metrics aim to contribute to the field's advancement, setting new benchmarks for quality and performance in neuromorphic chips and tool ecosystems.

The current set of metrics proposed in this document will be updated and refined iteratively along with project activities to evaluate performance in a meaningful way with a close link to our use-cases.

1.3 Organization of the Deliverable

The deliverable is composed of three parts:

1. Section 0 serves as a pivotal resource for evaluating the performance and energy consumption of the Neuromorphic Chip by means of well-defined metrics. Through meticulous analysis of these metrics, stakeholders can gain deeper insights into the chip's functionalities, enabling informed decision-making regarding its utilization in various applications.
2. Section 3 focuses on the thorough assessment of the simulation tool. The primary objective is to facilitate an effective comparison between the simulated models and their physical counterparts. Additionally, the section emphasizes benchmarking the scalability of the simulation tool, particularly in scenarios that need rapid prototyping and design exploration. By elucidating the tool's scalability and fidelity in modeling real-world scenarios, stakeholders can ascertain its suitability for diverse applications and expedite the development process.
3. Section 4 investigates the metrics for evaluating the quality of the Physical Unclonable Function (PUF) architectures developed in NEUROPULS. PUFs play a pivotal role in enhancing the security of hardware systems by leveraging unique physical characteristics for authentication purposes. Through rigorous analysis of various metrics, including reliability, uniqueness, and robustness, the section aims to provide valuable insights into the efficacy and performance of such architectures. By exploiting these metrics, stakeholders can make informed decisions regarding the deployment and integration of PUF-based security measures in their hardware systems, thereby fortifying their resilience against unauthorized access and cybersecurity threats.

2. Evaluating Neuromorphic Chip Accelerators

2.1 Performance Metrics

The architecture of the NEUROPULS final prototype is depicted in Figure 1. It is divided into two main components: (i) the host microcontroller (FPGA Board) and (ii) the Photonic Integrated Circuit (PIC).

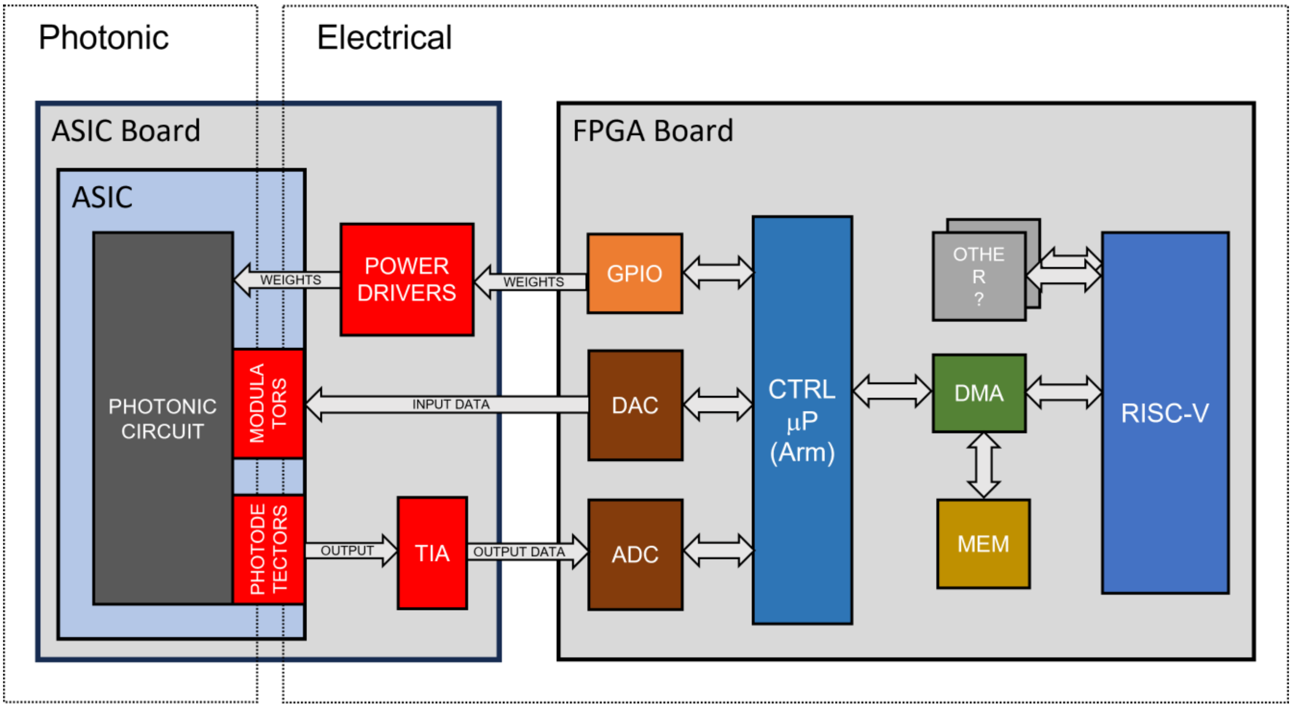


Figure 1: Neuromorphic Chip Preliminary Accelerator architecture. Figure does not include for simplicity the laser and its DC supply.

The host RISC-V microcontroller deploys specific computational kernels defined as Inference Operations (IOs) to the photonic circuit. The IO deployment requires a controller (CTRL) responsible for fetching the data from memory (MEM), sending them to the PIC, using the DAC to convert digital to analog electrical signals, and then leveraging the ADC to convert back the analog to digital electrical signals corresponding to the IOs outputs. In the ASIC board, specific components are responsible for the encoding of the electrical signal on the light carrier (MODULATORS) and for the detection of the optical signals and their optical-to-electrical conversion (PHOTODETECTORS) as well as transimpedance amplifiers (TIA) for the conversion from current to voltage to provide suitable signals to the ADC.

2.1.1 Latency

In this context, we define Latency as the time (in seconds) required to execute a given operation. Depending on the granularity, we can list the following metrics:

- **System-Level**: the latency is the time required to run the whole application. This metric includes all the hardware components involved during the computation.
- **RISC-V level**: the latency is the time required to run a set of IOs on the photonic circuit. This metric includes the CTRL, DMA, DAC, ADC, POWER DRIVERS, MODULATORS, and PHOTODETECTORS components involved in the computation.
- **CTRL level**: the latency is the time required to fetch data from memory, convert them from digital to analog (DAC), then encode the analog electrical signals onto the optical carrier (MODULATORS), and detect the optical signals and convert back to digital the results (PHOTODETECTORS followed by TIA and ADC) of a single inference.
- **ASIC-level**: the latency is the time required to encode the signals from the electrical domain (at the entrance of the power drivers) onto the optical carrier and at the output of the TIAs after being detected and converted into the electrical domain.
- **PIC-level**: the latency is the time required for the signals at the entrance of the MODULATORS to be detected and converted in electrical signals at the output of the PHOTODETECTORS.

Concerning the instruments used to measure the latency at the above levels, we plan to add specific components at the FPGA, ASIC and PIC to measure the five latency levels. In case of problems, we can still resort to indirect measures. For example, we can subtract the ADC/DAC times from the CTRL-level latency to obtain a good approximation of the latency at the ASIC level.

2.1.2 Memory Utilization and Data Transfer Efficiency

When evaluating the data transfer in a system that connects an FPGA-based microprocessor with external accelerators, it is essential to consider several key metrics to assess the efficiency and the performance of the data exchange. Data transfer is involved at the RISC-V level and the CTRL level. The data transfer rate will have the following definition:

- **Throughput (or bandwidth)**: Measure the amount of data transferred per unit of time (in seconds). This metric indicates the system's efficiency in moving data between the microprocessor and external accelerators.

While this metric can support an efficient measurement of the compatibility between the photonic part and the electrical counterpart, memory utilization challenges benchmarking as it will link with the CTRL-level management. For this reason, we expect to investigate the memory utilization by providing the following metric:

- **Buffering and Flow Control Overhead**: Examine the effectiveness of buffering and flow control mechanisms in managing data flow between the microprocessor and accelerators. It is the time between storing the output from the photonic accelerator in the shared memory and the time the RISC-V microprocessor can access it.

This is crucial for preventing bottlenecks and optimizing overall system performance.

2.1.3 Reliability

Reliability is the probability that the system properly works (i.e., it provides correct outputs) at a given time [1]. In our context, we will consider the inference accuracy loss induced by the presence of hardware faults as impact on reliability. The considered hardware faults are transient faults induced by external perturbation. Such faults can affect the electronic components and are mainly modeled as single-bit flips in memory cells (i.e., main memory, registers, single flip-flops) [1].

Since faults in the physical systems can only be simulated [2], the expectation is to carry out light fault injection campaigns based on the benchmarking applications, where an extra software layer modifies the input data to introduce soft errors on a single event-based methodology. The intricate nature of cutting-edge devices renders exhaustive Fault Injection (FI) campaigns practically unfeasible, often exceeding computational capabilities. A viable approach involves adopting statistical FI campaigns, allowing a reduction in the requisite number of experiments by selectively injecting a small, carefully chosen portion. Under specific conditions, statistical FIs ensure a precise understanding of the issue, albeit with a diminished sample size. Presently, challenges revolve around determining the optimal sample size, fault locations, and accurate interpretation of statistical assumptions [3]. For all those reasons, the injection of the faults will follow two primary objectives: firstly, to apply the correct specification of statistical FIs for Neural Networks (NNs), and secondly, to resort to a data analysis methodology that significantly diminishes the number of FIs required for achieving statistically meaningful results, all while upholding the integrity of the proposed approach [3]. Those results will be able to highlight the actual accuracy loss, i.e., only when the expected output-based decision deviates from the expected one.

2.2 Power Efficiency and Energy Consumption

As for the latency, the power and energy consumption will be defined at different granularity.

- **System-Level**: the power consumed [Watts] to run the whole application. The energy [Joule] is the power times the latency at this level. This metric includes all the hardware components involved during the computation.
- **RISC-V level**: the power consumed [Watts] to run a set of IOs on the photonic circuit. The energy [Joule] is the power times the latency at this level. This metric

includes the CTRL, DMA, DAC, ADC, POWER DRIVERS, MODULATORS, and PHOTODETECTORS components involved in the computation.

- **CTRL level**: the power consumed [Watts] to fetch data from memory, convert them from digital to analog (DAC), then encode them onto the optical carrier (MODULATORS), and detect the optical signals and convert back to digital the results (PHOTODETECTORS followed by TIA and ADC) of a single inference. The energy [Joule] is the power times the latency at this level.
- **ASIC-level**: the power consumed [Watts] to encode the signals from the electrical domain (at the entrance of the power drivers) onto the optical carrier and at the output of the TIAs after being detected and converted into the electrical domain. The energy [Joule] is the power times the latency at this level.
- **PIC-level**: the power consumed [Watts] for driving the MODULATORS and the PHOTODETECTORS.

All the powers and related energy consumptions discussed above shall also include the laser power with its wall-plug efficiency.

As for the latency, we plan to add specific components at the FPGA board and ASIC board levels to measure the different power and energy metrics. In case of problems, we can still resort to indirect measures. For example, from the CTRL-level power/energy, we can subtract the ADC/DAC power/energy to obtain a good approximation of the ASIC level.

3. Simulation of Neuromorphic Chip Accelerators

3.1 Introduction to Neuromorphic Chip Simulation

Focusing on achieving significant energy efficiency and security enhancements relies heavily on two key components: the photonic neural network (NN) accelerator and photonic security primitives (Physical Unclonable Functions - PUFs). A comprehensive system-level toolchain for modeling and simulation is essential to fully harness the potential of these photonic accelerators and seamlessly integrate them into a functional and programmable computing platform. NEUROPULS aims to design and implement a comprehensive simulation infrastructure to facilitate the utilization of photonic accelerators, such as NNs and PUFs, alongside other Phase Change Memory (PCM)-based modules. These tools will empower the creation of system-level models that precisely capture the functionality and performance metrics of the photonic modules, including performance (timing) and power consumption.

A pivotal aspect of this initiative is the development and implementation of simulation modules. This entails developing a simulation engine that efficiently implements complex and accurate system-level models tailored for different design and architecture exploration stages. We are constructing a simulation infrastructure that will model the hardware structures of a complete computing system comprising CPU cores (based on RISC-V ISA), memory hierarchy, and photonic NN accelerators. Interfaces between CPU cores and accelerators will be aligned with the accelerator implementation to ensure seamless integration.

The simulation platform will consider the impact of all computing stack layers, including hardware, system software (such as the operating system), and application software. To achieve this, the framework will be built upon gem5, a state-of-the-art microarchitectural simulator. This foundational infrastructure, known as gem5-MARVEL, has been developed on top of the gem5 in the context of the NEUROPULS project. gem5-MARVEL supports RISC-V-based systems modeling with diverse accelerators and flexible interfaces. Detailed information regarding gem5-MARVEL can be found in the related deliverable D5.9 of the NEUROPULS project; therefore, we omit the detailed explanation in this deliverable.

3.2 Neural Network Models for Simulation

The gem5-based simulation infrastructure comprises two core components of the NN accelerator: the Compute Unit and the Communications Interface. The Compute Unit represents the custom accelerator's datapath, while the Communications Interface facilitates memory access, control, and synchronization through memory access ports, Memory-Mapped Registers (MMRs), and interrupt lines. The memory access ports allow parallel access to different memory types like scratchpad memories (SPMs) and register banks (these two types of memories occupy the most significant part of the area of many accelerators). MMRs consist of configurable status, control, and data registers, enabling low-level device configuration and facilitating communication between the accelerator and the host as well as between multiple NN accelerators in a cluster. The host can utilize the provided interrupt signals for synchronization without constant polling by treating the NN accelerator as a memory-mapped device.

Additionally, the gem5-based infrastructure includes Direct Memory Access (DMA) devices and custom memories that can be seamlessly integrated into accelerator designs, enhancing its versatility. Figure 2 shows the SoC architecture. Specifically, our tested accelerator designs are loosely coupled and communicate with the host CPU via MMRs and DMA transactions. The CPU writes the input and output memory addresses to the accelerator MMRs and directs the accelerator to start the computation. The accelerator transfers the data to its SPMs or Register Banks via DMA, performs the required calculations, and transfers it back to the system memory. After task completion, it notifies the host via a pre-defined interrupt.

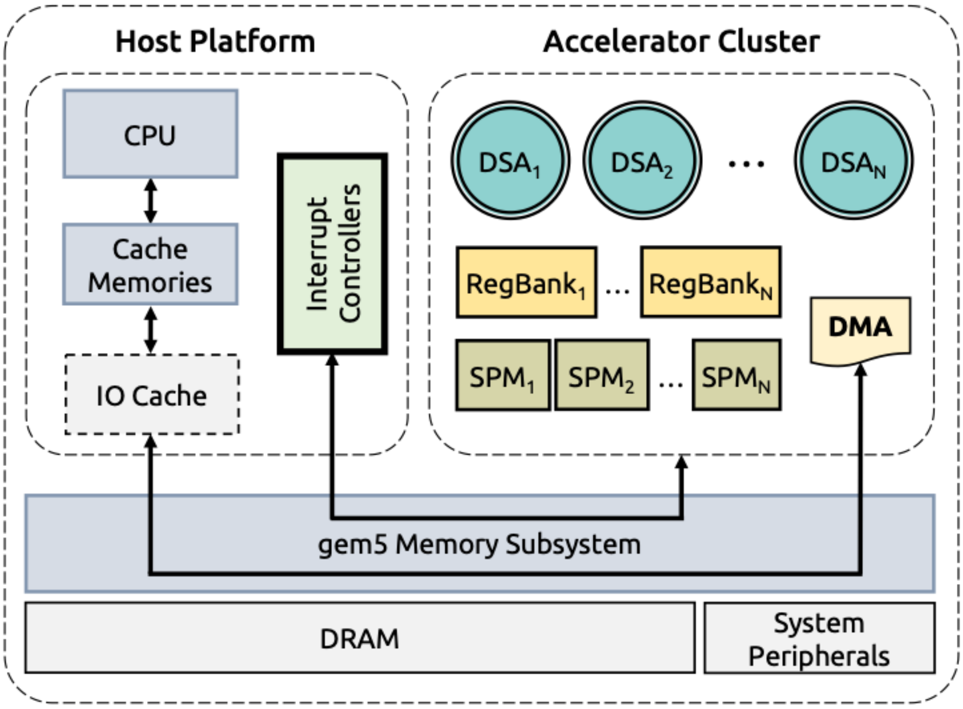


Figure 2: gem5-based SoC architecture and interconnection.

3.3 Challenges in Simulating Large-Scale Neural Networks

Simulating accelerators for large-scale NNs poses several significant challenges due to the size and complexity of these systems. Some of the key challenges include:

1. **Computational Resources:** Large-scale NNs may require vast computational resources to be simulated effectively. As the number of neurons and connections increases, the computational demands grow exponentially, requiring many computational resources to simulate effectively large-scale NNs, especially in cycle-level simulation, such as gem5 (the baseline simulator used in the NEUROPULS project).
2. **Scalability**: Ensuring that simulation platforms can scale efficiently with the size of the NN is also a significant challenge. Scalability issues regarding computational performance and memory usage can arise, requiring innovative parallelism and distributed computing approaches. For example, to evaluate the reliability of an accelerator design, the gem5-MARVEL framework we have built in the context of the NEUROPULS project utilizes multiple systems and/or CPU cores to speed up the assessment, turning the simulation time problem into an infrastructure scale one.
3. **Synchronization and Communication Overhead**: In distributed or parallel simulations, managing synchronization and communication overhead between computing nodes of CPU cores running different scenarios becomes critical. Minimizing latency and ensuring efficient data exchange are essential for maintaining simulation accuracy and performance.
4. **Parameter Tuning and Optimization**: Large-scale NNs often have numerous parameters that must be tuned and optimized for effective performance. Parameter tuning and modeling of a specific NN can be time-consuming and computationally intensive, particularly when exploring a large parameter space.
5. **Validation and Verification**: Validating the accuracy and reliability of simulation results for large-scale NNs may be challenging for specific accelerator designs due to the absence of ground truth data. Rigorous validation and verification methodologies are necessary to ensure the fidelity of simulation outcomes. For example, gem5-MARVEL (see details in the deliverable D5.9) runs the exact algorithm of a specific accelerator design on both the accelerator and the CPU to compare the results. In such a case, our simulation infrastructure ensures the validity of the results.

To address these challenges, advanced simulation techniques, efficient algorithms, and simulation-scalable computing architectures are essential for enabling the simulation of large-scale NNs and unlocking their full potential for scientific research and practical applications.

3.4 Metrics

Simulators, especially gem5-based simulators, which are the primary vehicle of this project, inherently provide some information with respect to performance and other metrics abovementioned. However, in the context of the NEUROPULS project, we also offer several different metrics, which will provide essential information about the workload and accelerator design executions on a simulation platform. Below, we present some crucial metrics regarding performance, power, and reliability of the workloads and NN accelerator designs that our simulation infrastructure will provide or metrics already implemented in our baseline simulation infrastructure.

3.4.1 Performance and Area of NNs

gem5-MARVEL is based on gem5-SALAM [4], which uses an advanced dynamic graph execution engine based on LLVM [5]. gem5-SALAM instruments the low-level virtual machine (LLVM) IR (Intermediate Representation) to model domain-specific accelerators (DSAs) using C descriptions of their functionality. gem5's tight integration enables seamless and intricate interaction between the accelerator and other system modules, including the CPU and the memory subsystem. Its high level of integration within gem5 allows for complex interaction between the accelerator and other system modules, such as the CPU and the memory subsystem.

gem5-MARVEL also offers a range of performance metrics to users after simulation. Within the device setup, it specifies the cycle time required for each LLVM IR instruction to execute in the compute queues. Users can define hardware device latency and clock speed within the accelerator. This allows for accurate modeling and exploration of their impact on accelerator models' cycle counts, runtime, and functional unit occupancy.

During dynamic runtime simulation, our infrastructure records the scheduling or in-flight status of instructions for each cycle. This additional data, coupled with configurable hardware resources, provides a detailed analysis and exploration tool for examining occupancy levels within the system.

The area estimation model utilizes parameters specified in a hardware profile and a device configuration file, which users can customize. The hardware profile contains details regarding different hardware components used by the accelerator (e.g., Multiply-Accumulate Units). The device configuration file allows users to define and adjust the values according to the real characterization measurements and refine the allocation of hardware components. Using these parameters, the framework calculates the area estimations based on LLVM IR analysis of the C description of the accelerator inferring the datapath of the design, similarly to how high-level synthesis works in Register Time Level (RTL) design.

3.4.2 Static and Dynamic Power

gem5 offers a robust platform for modeling and simulating intricate computing systems, facilitating thorough analysis of computational performance and power attributes. We will focus on integrating key power-related metrics into our foundational gem5-MARVEL framework.

In static power evaluation, our emphasis will be on estimating the power consumption of neural network (NN) models by considering architectural features and configuration parameters, irrespective of runtime behavior. This entails modeling the power usage of individual hardware components, such as CPU cores, memory hierarchy, and accelerators, alongside their interactions within the system. Through analyzing static power characteristics, valuable insights can be extracted into the base power consumption of the system across varied operational conditions and setups.

In dynamic power evaluation, our emphasis will be on the power consumption of NN models during runtime execution. It entails monitoring the power usage of hardware components as they process input data and execute computational tasks. Dynamic power evaluation will capture the fluctuating power consumption patterns stemming from diverse computational workloads, data dependencies, and hardware utilization levels. By examining dynamic power behavior, we can explore the energy efficiency of various NN designs and architectures, identifying potential obstacles for power optimization. This contributes to the simulation-based design space exploration, a core aspect of the NEUROPULS project.

3.4.3 Reliability Evaluation

gem5-MARVEL evaluates both the Architectural Vulnerability Factor (AVF) and the Hardware Vulnerability Factor (HVF) and provides accurate evaluation results using statistical fault injection for both metrics. A hardware structure's Hardware Vulnerability Factor (HVF) is the fraction of faults in the structure that are either activated within the hardware layer or exposed to a higher layer [6]. A hardware-visible fault is exposed to the user program once it reaches a software (or architecture-visible) resource [7].

gem5-MARVEL employs two vulnerability evaluation methodologies of different layers: the HVF assessment [6] and the AVF assessment, providing the partial microarchitecture-dependent vulnerability and the full cross-layer vulnerability, respectively. For accelerator designs, where the faults target the scratchpad memories of each design, the HVF and AVF analyses are identical. The reason is that the architecture of a domain-specific accelerator differs from that of a general-purpose CPU.

In an accelerator design, any fault is eventually visible unless the fault hits an invalid or unused cell of the scratchpad memory. In that case, the fault is characterized as masked. The HVF analysis considers Benign faults, those faults that eventually get masked by a microarchitectural operation (e.g., a misprediction), and thus, the fault occurrence never reaches the commit stage of an out-of-order microprocessor (i.e., the fault is not architecturally visible). On the other hand, any fault that reaches the commit stage (i.e., architecturally visible) is considered a corruption and participates in the total HVF measurement.

3.4.4 Performance-Aware Comparisons

Regarding the reliability evaluations, AVF (Architectural Vulnerability Factor) is a pure reliability metric that does not provide any information about the system's performance. AVF alone cannot provide any insights into the tradeoff between the performance and reliability of a chip. To this end, gem5-MARVEL can compute a new simple reliability metric, Operations per Failure (OPF).

OPF is the number of times a workload is executed before a system failure happens. It is computed using the following formula: OPF = OPS / AVF, where OPS (Operations per Second) is the number of operations (i.e., tasks) the compute unit can perform during 1 second. Assume, for example, the Matrix Multiplication algorithm, which performs 2 x N^3 operations, where N is the size of the matrices. Thus, OPS = 2 x N^3 / Exec_Time.

The OPF metric enables a combined performance and reliability analysis into a single metric. For the same workload that runs on different platforms (a CPU or an accelerator in our example), larger OPF values indicate a better tradeoff between reliability and performance (larger number of correct executions over time).

4. Security evaluation of NEUROPULS-based PUFs

While developing Strong Physical Unclonable Functions (Strong PUFs), researchers often face challenges in assessing their security. One underlying problem is that security evaluation methods can sometimes be vague, subjective, hard to grasp, and technology-dependent. In the context of Strong PUFs, we would like to achieve the unpredictability of the Challenge-Response-Pairs (CRPs) with respect to modeling attacks [8]. It must be unfeasible for an attacker to calculate unknown Responses to specific sets of Challenges. A fundamental advantage of such a property is that the interface to access the PUF can be publicly available [8].

What sounds straightforward is indeed hard to achieve and even harder to measure. As in many areas of computer security, we may only define necessary conditions that Strong PUFs should fulfill. Even though the fulfillment of sufficient conditions would lead to an overall acceptance of the security and unpredictability of Strong PUFs, we have no such metrics. Thus, including various metrics necessary to fulfill is even more critical. Therefore, we work on different approaches with which we will achieve a holistic picture of the behavior and security of novel PUF designs.

Since standardized and easy-to-use metrics have yet to exist in Strong PUF research, we are also developing novel security metrics. In the first section, we discuss a new metric based on the visualization of the CRPs of Strong PUFs. Such visualizations are easy to use, comparable to other Strong PUFs, and usable even in the case of small sets of CRPs. A developed tool for such analysis helps to achieve these desirable goals. In the following section, we also outline other security metrics that should be used and applied in the context of NEUROPULS. These are modeling attacks, randomness testing, and measuring the response sensitivity to challenge variations.

4.1 Development of Novel Visualization Metrics for Strong PUFs

The following metric was initially proposed in [6] and, since then, further developed in the context of the NEUROPULS project. Since the thesis, we have worked on improving compression scores and implementing a tool that automatizes the analysis.

The metric uses a numerical score and an equivalent visualization approach [9]. In the short term, we developed a specific and standardized format to arrange sets of CRPs in a quadratic picture. The standardization of that format has been previously extensively researched. As an example: given a 10-bit Arbiter PUF, 2^{10} = 1024 CRPs in total (please note that such PUF has indeed not enough CRPs for being used as a secure *Strong* PUF given that brute force approaches could easily break it, but it is useful as an example).

Furthermore, we have the whole set of CRPs simulated or measured, such that we have a dataset of 1024 CRPs. In the next step, we arrange the CRPs in a standardized manner.

The pixel (or field) is the corresponding Response to a Challenge. Further specifications of the standardized algorithm are laid out in a paper we are currently writing. Below, we show the results of different PUF architectures. The following images and tables are taken from the yet-unpublished paper. The CRPs are simulated with the tool *pypuf* [10].

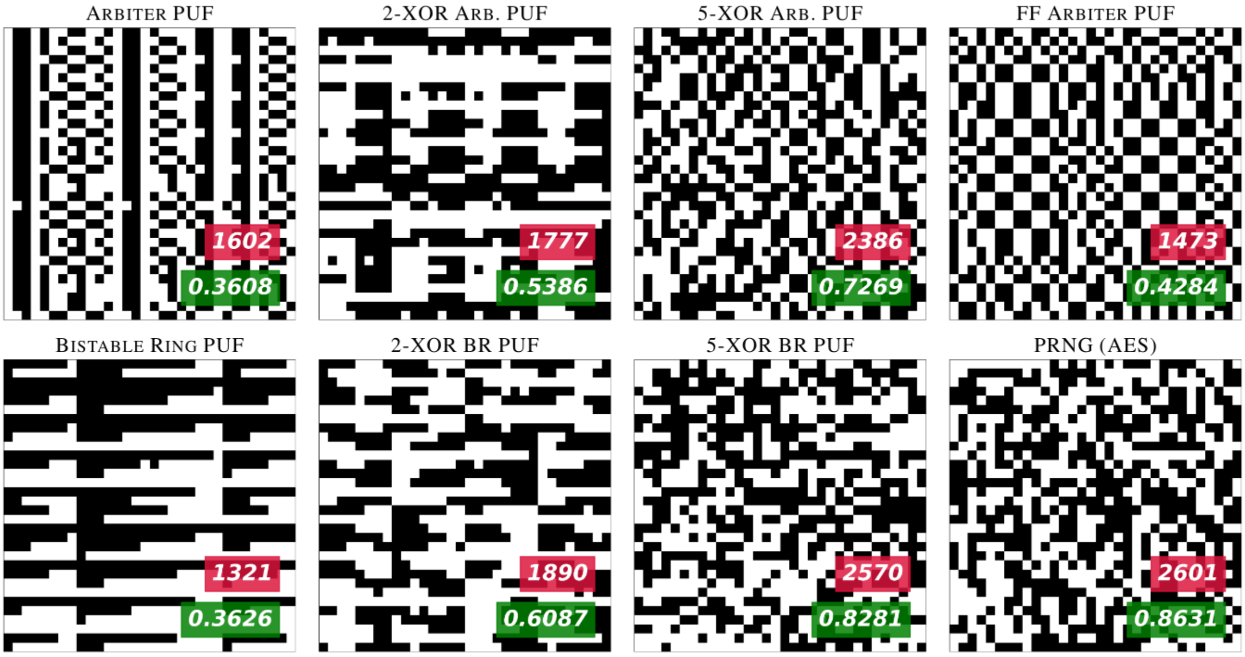
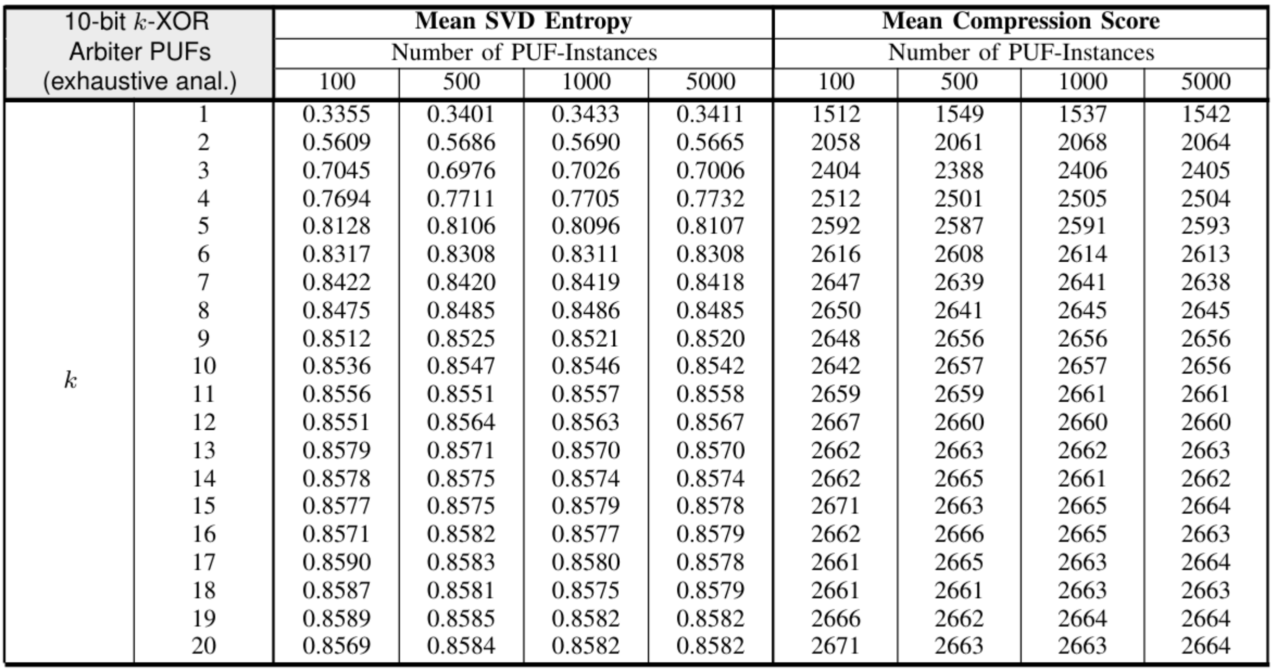


Figure 3: Example of visualizations for different PUFs.

As shown in Figure 3, the statistical behavior of different Strong PUFs is indeed distinguishable by eye inspection. Even more interesting we can see at first glance that the Arbiter PUF, the Bistable Ring PUF, and the Feed-Forward (FF) Arbiter PUF are far from random. However, as expected, the randomness increased when we increased k. Such behavior is well-known and matches previous experience from modeling attacks. As a comparison, we show an image generated with AES.

Furthermore, there are two insets – one in green and one in red. These are numerical measurements with which we aim to capture the visual impression. The green insets are the so-called Singular Value Decomposition (SVD) Entropy of these images [11]. For the SVD Entropy, we empirically determined a desirable average score of 0.8583 for this size of the images. The red one is the memory of the image after compression with PNG. In simple words, a high compression score is desirable. Extending the previously named master thesis and the SVD Entropy, we mainly worked on the compression scores during the last months as they are the most promising ones. In the following table, we computed the scores for 100, 500, 1000, and 5000 k-XOR Arbiter PUF instances for each k. Table 1 shows both the SVD Entropy and the mean compression score.

Table 1: Mean SVD entropy and compression scores for a 10-bit k-XOR arbiter PUF.

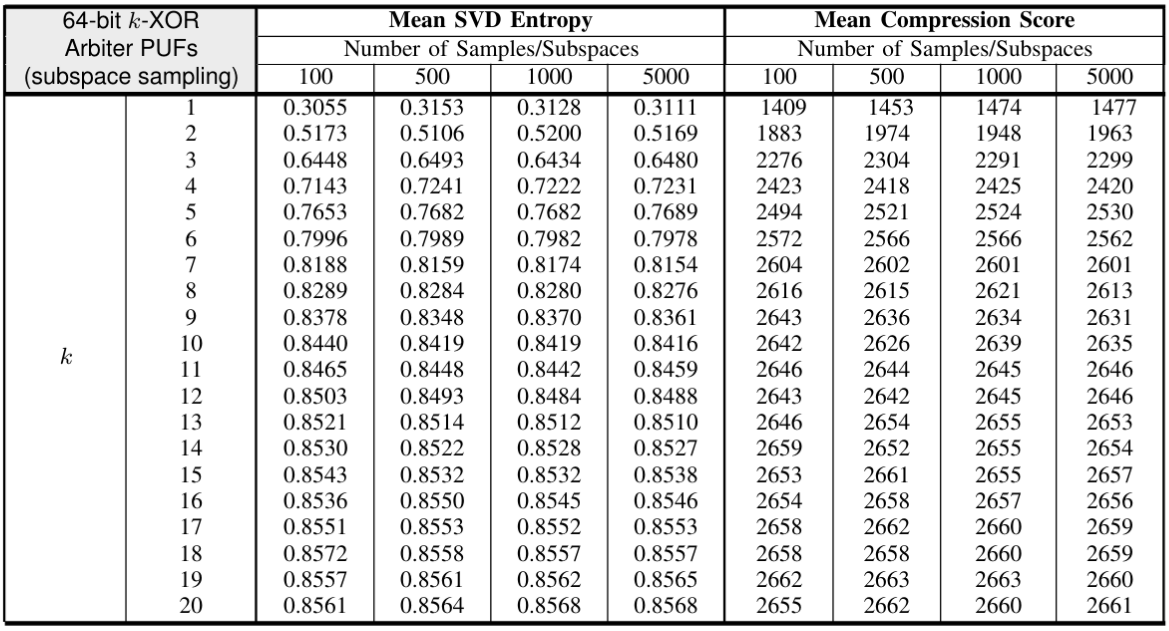


10-bit k-XOR Arbiter PUFs (exhaustive anal.)		Mean SVD Entropy				Mean Compression Score			
		Number of PUF-Instances				Number of PUF-Instances			
		100	500	1000	5000	100	500	1000	5000
k	1	0.3355	0.3401	0.3433	0.3411	1512	1549	1537	1542
	2	0.5609	0.5686	0.5690	0.5665	2058	2061	2068	2064
	3	0.7045	0.6976	0.7026	0.7006	2404	2388	2406	2405
	4	0.7694	0.7711	0.7705	0.7732	2512	2501	2505	2504
	5	0.8128	0.8106	0.8096	0.8107	2592	2587	2591	2593
	6	0.8317	0.8308	0.8311	0.8308	2616	2608	2614	2613
	7	0.8422	0.8420	0.8419	0.8418	2647	2639	2641	2638
	8	0.8475	0.8485	0.8486	0.8485	2650	2641	2645	2645
	9	0.8512	0.8525	0.8521	0.8520	2648	2656	2656	2656
	10	0.8536	0.8547	0.8546	0.8542	2642	2657	2657	2656
	11	0.8556	0.8551	0.8557	0.8558	2659	2659	2661	2661
	12	0.8551	0.8564	0.8563	0.8567	2667	2660	2660	2660
	13	0.8579	0.8571	0.8570	0.8570	2662	2663	2662	2663
	14	0.8578	0.8575	0.8574	0.8574	2662	2665	2661	2662
	15	0.8577	0.8575	0.8579	0.8578	2671	2663	2665	2664
	16	0.8571	0.8582	0.8577	0.8579	2662	2666	2665	2663
	17	0.8590	0.8583	0.8580	0.8578	2661	2665	2663	2664
	18	0.8587	0.8581	0.8575	0.8579	2661	2661	2663	2663
	19	0.8589	0.8585	0.8582	0.8582	2666	2662	2664	2664
	20	0.8569	0.8584	0.8582	0.8582	2671	2663	2663	2664

As expected, the scores increase with a larger k. With 5000 samples for each k, we can reliably distinguish different SVD Entropy scores up to k = 15. The SVD Entropy approaches the score 0.8583 and the PNG compression 2664 bits. As the compression score does not rely on an empirically defined desirable score, our research has shown that compression is a more suitable candidate for such a security metric. The results are to be published for other architectures as well. We already computed such tables for the Bistable Ring PUF and the Feed-Forward Arbiter PUF. Therefore, we can compare new PUF architectures with the same challenge length to existing PUF designs.

As noted, a 10-bit Strong PUF with only 1024 CRPs must be more secure. However, can we also apply our metric to larger sets of CRPs, like from a 64-bit Strong PUF? The problem with a 64-bit Strong PUF is that we have a set of 2^{64} CRPs, which we cannot show exhaustively in one image. So, we have to adapt the algorithm slightly. In a standardized fashion, we reduce the set of CRPs to specific *subspaces*. Our research has shown that such reduction does not lead to worse comparability of different k-XOR Arbiter PUFs. Table 2 shows the results of 64-bit k-XOR Arbiter PUFs.

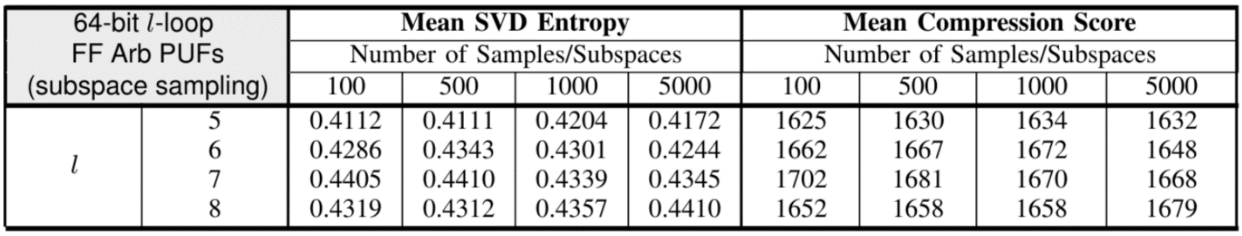
Table 2: Mean SVD entropy and compression scores for a 10-bit k-XOR arbiter PUF.



64-bit k-XOR Arbiter PUFs (subspace sampling)		**Mean SVD Entropy**				**Mean Compression Score**			
		Number of Samples/Subspaces				Number of Samples/Subspaces			
		100	500	1000	5000	100	500	1000	5000
k	1	0.3055	0.3153	0.3128	0.3111	1409	1453	1474	1477
	2	0.5173	0.5106	0.5200	0.5169	1883	1974	1948	1963
	3	0.6448	0.6493	0.6434	0.6480	2276	2304	2291	2299
	4	0.7143	0.7241	0.7222	0.7231	2423	2418	2425	2420
	5	0.7653	0.7682	0.7682	0.7689	2494	2521	2524	2530
	6	0.7996	0.7989	0.7982	0.7978	2572	2566	2566	2562
	7	0.8188	0.8159	0.8174	0.8154	2604	2602	2601	2601
	8	0.8289	0.8284	0.8280	0.8276	2616	2615	2621	2613
	9	0.8378	0.8348	0.8370	0.8361	2643	2636	2634	2631
	10	0.8440	0.8419	0.8419	0.8416	2642	2626	2639	2635
	11	0.8465	0.8448	0.8442	0.8459	2646	2644	2645	2646
	12	0.8503	0.8493	0.8484	0.8488	2643	2642	2645	2646
	13	0.8521	0.8514	0.8512	0.8510	2646	2654	2655	2653
	14	0.8530	0.8522	0.8528	0.8527	2659	2652	2655	2654
	15	0.8543	0.8532	0.8532	0.8538	2653	2661	2655	2657
	16	0.8536	0.8550	0.8545	0.8546	2654	2658	2657	2656
	17	0.8551	0.8553	0.8552	0.8553	2658	2662	2660	2659
	18	0.8572	0.8558	0.8557	0.8557	2658	2658	2660	2659
	19	0.8557	0.8561	0.8562	0.8565	2662	2663	2663	2660
	20	0.8561	0.8564	0.8568	0.8568	2655	2662	2660	2661

A remaining question is: How can we compare the security of PUFs in the context of NEUROPULS with previous architectures? As one promising approach, we compared different architectures with the already well-known and well-studied k-XOR Arbiter PUF. As such, we can differentiate both the visual impression and SVD Entropy scores and compression scores. For example, as it is well known from previous research, the Feed-Forward Arbiter PUF (FF Arbiter PUF) is insecure. Our research has compared the FF Arbiter PUF with k-XOR Arbiter PUFs. The visual impression and the score show that the FF Arbiter PUF lies only between a 1-XOR Arbiter PUF and a 2-XOR Arbiter PUF. Moreover, the number of loops of the FF Arbiter PUF needs to be more relevant. Table 3 shows the scores for 64-bit FF Arbiter PUFs with several loops l:

Table 3: Mean SVD entropy and compression scores for a 64-bit l-loop FF arbiter PUF.



64-bit l-loop FF Arb PUFs (subspace sampling)		**Mean SVD Entropy**				**Mean Compression Score**			
		Number of Samples/Subspaces				Number of Samples/Subspaces			
		100	500	1000	5000	100	500	1000	5000
l	5	0.4112	0.4111	0.4204	0.4172	1625	1630	1634	1632
	6	0.4286	0.4343	0.4301	0.4244	1662	1667	1672	1648
	7	0.4405	0.4410	0.4339	0.4345	1702	1681	1670	1668
	8	0.4319	0.4312	0.4357	0.4410	1652	1658	1658	1679

There are several advantages to employ such a metric. First, the metric works in a black-box manner. That means researchers can find out the underlying architecture of a Strong PUF for analyzing security properties. Second, we achieved desirable results by comparing different architectures. These results match with the previous experience from modeling attacks and other metrics. Third, we use the human perception of randomness combined with a score as a single figure of merit. Fourth, the metric works in the extended version for all challenge lengths. Still, the need for computational power or the size of a data set of CRPs is comparably low. To sum up, the metric is handy at the beginning of developing and designing new PUFs.

Since such algorithms are tedious to implement, we have been working on a tool called *pufvis*. *pufvis* implements the above-described properties of the metric. Furthermore, it encapsulates the simulations of *pypuf*. Still, it is open to other CRPs and incorporates a simple-to-use Python interface. Since the design principles of the implementation follow extensibility, further metrics like the ones described below may be easily implemented and integrated as well. The long-term goal is a holistic set of security metrics that helps us understanding the behavior and the security of very different Strong PUF designs and architectures.

4.2 Application of Further Metrics

Such a visualization metric is a first step to secure Strong PUFs. However, we should again emphasize that such a metric is only a necessary condition, not a sufficient one. Therefore, we have to work on further metrics and applications. The most traditional, promising, and essential method is to apply modeling attacks on Strong PUFs [12]. Such attacks try to exploit the missing fulfillment of the unpredictability property. When such attacks are successful, we must accept that such Strong PUFs are not secure. On the contrary, unsuccessful attacks rather speak for the security of Strong PUFs. To get an overall and more comprehensive picture of the security of Strong PUFs, we may also apply further metrics beyond the visualization and modeling attacks. Two metrics suit that case: I) Randomness testing for uniqueness, bit-aliasing, and uniformity [13]. II) Measuring the response sensitivity to challenge variations [14].

Randomness testing is suitable since it has already been applied to many cryptographic applications. Furthermore, when such tests are successful, we may expect better unpredictability of CRPs. Such randomness testing is connected to the previously explained and novel visualization metrics. The latter can show visually when sets of CRPs are not random, as in the case of plain Arbiter PUFs. By comparing existing and simulated Strong PUF architectures, the results matched our previous experience and previous research on already broken PUF architectures.

Measuring the response sensitivity to challenge variations is another easy-to-use and straightforward security metric for Strong PUFs. Developed in 2022 [14], such a metric examines whether small perturbations of challenges impact the response. Imagine, e.g., the case when the first bit of a Strong PUF with a challenge length of 64-bit does not influence the probability of flipping the response. Such behavior would be undesirable since an attacker could eliminate that bit's influence.

Previous research shows that such behavior is visible in the Arbiter PUF design [14]. The first challenge bits have a low impact on the probability of flipping the response bit. On the contrary, the bits at the end of a challenge significantly impact the likelihood of flipping the response bit. Both properties are not desirable since we would like to achieve a probability of 50%. In that case, the attacker could not conclude, from a set of CRPs, the behavior of unknown CRPs. The findings of such probability tests match our research of the visualizations remarkably well. However, to compare different PUF designs and architectures, there is a need for a single figure of merit, which is currently being under investigation.

To sum up, our different perspectives on the security of Strong PUFs have led to a holistic set of metrics that shall be fulfilled. With these metrics' applications on the existing Strong PUF architectures like the Arbiter PUF or the Bistable Ring PUF, we may compare the results of the NEUROPULS project with existing designs. Even though the security degree might need to be improved further, we can precisely show which problems arise with which designs. Therefore, analysis with these metrics will lead to a more profound understanding of the behavior, security, and complex nature of the NEUROPULS PUFs.

Conclusion

The NEUROPULS project foreseen outcomes span from the physical photonic architectures, including the neuromorphic accelerator and the PUF, to the simulation tools that will model it to support a fast prototyping and design space exploration.

To pave the way to a meaningful benchmarking, this deliverable provides a set of metrics to address each aspect of the NEUROPULS products. It includes the most significant performance and energy consumption metrics to assess the low power target of the project. At the same time, the modeled digital twin in the simulation tool finds the definition of useful metrics not only to guarantee the quality of the modeling in terms of accuracy of the measurements, but also to confirm the potentiality to simulate bigger photonic counterparts that will be available in the future.

Eventually, the PUF design and implementation is supported by a set of metrics that ensure the fair comparison with the state of the art and the quality of the fundamental security characteristics that are found in weak or strong PUFs.

Bibliography

[1] G. Di Natale, D. Gizopoulos, S. Di Carlo, A. Bosio and R. Canal, Cross-Layer Reliability of Computing Systems, IET Digital Library, 2020, p. doi:10.1049/PBCS057E.

[2] A. Bosio, S. D. Carlo, M. Rebaudengo and A. Savino, "Toward the hardening of real-time operating systems," in *2022 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, Austin, TX, USA, 2022.

[3] A. Ruospo, G. Gavarini, C. de Sio, J. Guerrero, L. Sterpone, M. S. Reorda, E. Sanchez, R. Mariani, J. Aribido and J. Athavale, "Assessing Convolutional Neural Networks Reliability through Statistical Fault Injections,," in *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Antwerp, Belgium , 2023.

[4] S. Rogers, J. Slycord, M. Baharani and H. Tabkhi, "gem5-salam: A system architecture for llvm-based accelerator modeling," in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2020.

[5] S. Rogers, J. Slycord, R. Raheja and H. Tabkhi, "Scalable llvm-based accelerator modeling in gem5," *IEEE Computer Architecture Letters,* vol. 18, no. 1, p. 18–21, 2019.

[6] V. Sridharan and D. R. Kaeli, "Using Hardware Vulnerability Factors to Enhance AVF Analysis," in *Proceedings of the 37th Annual International Symposium on Computer Architecture, ser. ISCA '10*, New York, NY, USA, 2010.

[7] G. Papadimitriou and D. Gizopoulos, "Anatomy of On-Chip Memory Hardware Fault Effects Across the Layers," *IEEE Transactions on Emerging Topics in Computing,* pp. 1–12, doi:10.1109/TETC.2022.3205808, 2022.

[8] U. Rührmair and D. E. Holcomb, "PUFs at a glance," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Dresden, Germany, 2014.

[9] L. Lerch, "Security Metrics for Strong PUFs based on Effective 2D Visualization," 2023.

[10] N. Wisiol, C. Gräbnitz, C. Mühl, B. Zengin, T. Soroceanu, N. Pirnay, K. T. Mursi and A. Baliuka, "pypuf: Cryptanalysis of Physically Unclonable Functions," *Zenodo,* vol. Version v2, p. doi:10.5281/zenodo.3901410, August 2021.

[11] O. Alter, P. O. Brown and D. Botstein, "Singular value decomposition for genome-wide expression data processing and modeling," in *Proc. Natl. Acad. Sci.*, USA, 2000.

[12] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas and J. Schmidhuber, "Modeling attacks on physical unclonable functions," in *17th ACM conference on Computer and communications security (CCS '10)*, 2010.

[13] A. Maiti, V. Gunreddy, and P. Schaumont, A Systematic Method to Evaluate and Compare the Performance of Physical Unclonable Functions, Cryptology ePrint Archive, 2011. https://eprint.iacr.org/2011/657.pdf

[14] F. Kappelhoff, R. Rasche, D. Mukhopadhyay and U. Rührmair, "Strong PUF Security Metrics: Response Sensitivity to Small Challenge Perturbations," in 2022 23rd International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, USA, 2022.