

Ahead of Time Generation for GPSA Protection in RISC-V Embedded Cores

Abstract

State-of-the-art hardware countermeasures against fault attacks are based, among others, on control flow and code integrity checking. These integrities can be asserted by Generalized Path Signature Analysis and Continuous Signature Monitoring. However, supporting such mechanisms requires a dedicated compiler flow and does not support indirect jumps. In this work we propose a technique based on a hardware/software runtime to generate those signatures while executing unmodified COTS RISC-V binaries. The proposed approach has been implemented on a pipelined rv32i processor, and experimental results show an average slowdown of $\times 1.82$ compared to unprotected implementations while being completely compiler independent.

Introduction

Because of their nature, embedded systems are prone to physical attacks. Several works have demonstrated that a well-designed cryptographic application, whose implementation is considered safe, can be compromised with fault injection attacks (eg., laser, EM, clock or power glitch), which induce an incorrect behavior of the victim processor or a data leak [1].

Countermeasures against such faults can be implemented both in software or in hardware. Software countermeasures, often inserted at compile time, consist of duplicating part of the instructions to detect and counter fault injections. This type of countermeasures has reached its limits with the emergence of attacker models allowing for several faults happening in a single execution. On the other hand, hardware countermeasures rely on a modified processor microarchitecture which ensures some form of Control Flow Integrity (CFI) and code integrity. Among the numerous existing techniques, Generalized Path Signature Analysis (GPSA) and Continuous Signature Monitoring (CSM) [2] happen to provide the best trade-off between sensitivity and area/performance overhead.

GPSA/CSM relies on cryptographic signatures to ensure integrity. Throughout the execution, the processor computes a signature based on previously executed instructions. The dynamic signature is verified against a reference signature at each branch and patches are used to correct the signature when executing branches. Additional instructions are therefore needed to load signatures during the execution. Besides, patches and reference signatures must be computed ahead of time and inserted in the executable.

In GPSA/CSM, the processor datapath is considered as being protected against faults, for example through error-detecting codes in both pipeline stage

registers and data/code memory.

This technique is implemented in The SCI-FI RISC-V core [3], along with an additional mechanism that protects pipeline control signals through some form of redundancy.

SCI-FI and other existing approaches share common limitations: i) the target application needs a custom compilation flow to embed signature and patches; ii) indirect branches cannot be handled without strong assumptions on the possible targets; iii) function calls, returns, and interrupts require to store/restore signatures which increases attack surface.

Previous work [4] overcomes these limitations with a runtime environment for the generation of GPSA values. This solution comes with a high cost, in both time and area. Only relying on an interrupt mechanism and a routine to handle all the program GPSA values.

In this paper, we present a method to mitigate the high overheads induced by the method of Savary *et al.* [4]. Our approach relies not only on a runtime environment, but also on a GPSA value generation when deploying the program. Our runtime also transparently handles indirect branches, function calls, interrupts, and context switches.

We have designed a proof of concept implementation based on the Comet RISC-V processor [5]. In our implementation, the pipeline is modified to check signatures on control flow instructions and trigger an interrupt to update patches and signatures whenever an indirect jump with missing signature is executed.

Our approach has been validated through fault injection simulations to ensure that protection was effective. The experimental study also shows that the average performance slowdown factor due to dynamic analysis is $\times 1.82$.

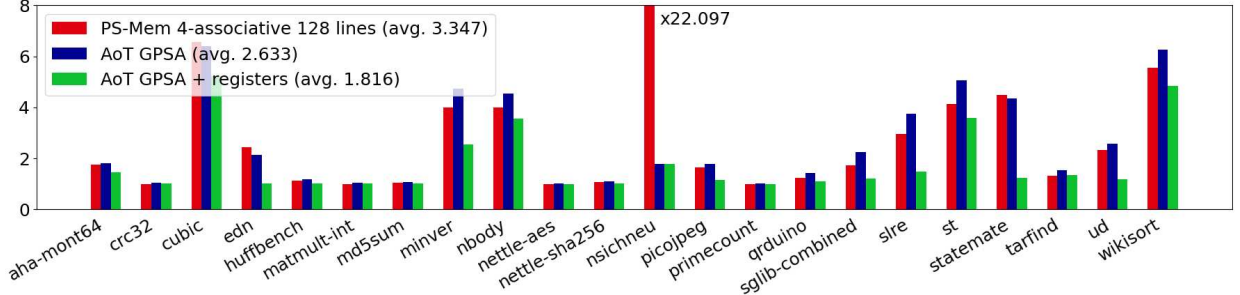


Figure 1: Runtime overhead on Embench-IoT. The first column corresponds to [4] with a ps-mem main memory of 128 lines, 4-associative. GPSA deploy corresponds to the solution presented in this paper. registers correspond to two sets of 16 128bits register for indirect jumps, and other instructions, gpsa values.

Ahead of Time Analysis for GPSA

In order to apply GPSA protection on COTS binaries, but with less overheads than existing approaches, we propose to compute the GPSA values ahead of time. These values are then stored in data memory.

During execution, we need the hardware to easily access the GPSA values of the executed instruction. To do this, these values are stored with the following structure: a list of tuple, each corresponding to a control flow instruction, sorted by PC. To ease the data cache fetching in memory, the tuples addresses are align to the data cache line size. A register is also added to the core, pointing to the values of the next control flow instruction to be executed.

To browse the list of tuple in constant time, it is sorted by PC and a fourth value is contained in the tuples: the address offset. In the tuple corresponding to an instruction a , the offset is the difference between the address of this tuple and the address of the tuple corresponding to the control flow instruction following the target of the instruction a . With this structure, when the CCFI component processes a control flow instruction, it loads the data cache line containing the values of this instruction. With these values, it verifies the dynamic signature. If the branch is taken, the signature is updated and the address of the tuple corresponding to the next instruction is obtained by adding the offset to the current tuple address. Otherwise, the corresponding tuple is the following tuple in the list, because it is sorted by PC.

Concerning patches for indirect jumps, as their targets cannot be known ahead of time, their computation is left to an interrupt mechanism, similar to the one from Savary *et al.* [4].

Experimental study

We implemented our solution on the Comet RISC-V processor. The overall area overhead has been evaluated thanks to an HLS tool and is presented in the Table 1.

The figure 1 shows the slowdown between the previous solution from Savary *et al.* [4] and our solution

Core	area(μm^2)	overhead
PS-Mem [4]	150311	126.6%
AoT GPSA	74856	12.9%
AoT GPSA + registers	86130	29.9%

Table 1: Area overhead of different solutions. PS-Mem refers to solution from [4] with a 4-associative 128 lines main memory. AoT GPSA is the solution presented in this paper. Registers represents two sets of 16 128bits registers for GPSA values.

on the Embench-IoT benchmarks [6], normalized on the performances of an unmodified Comet. Results show a slow-down factor between 1.0 and 5.23, with an average of $\times 1.82$.

Conclusion

In this paper, we propose a method to apply GPSA and CSM protections on unmodified binaries, with an average runtime overhead of $\times 1.82$ and area overhead of 30%. As far as we know, this is the best hardware/software implementation for GPSA without compiler dependence and allowing integrity properties to hold while handling indirect jumps, function calls, interrupts as well as context switches.

References

- [1] J. Laurent et al. "Fault Injection on Hidden Registers in a RISC-V Rocket Processor and Software Countermeasures". In: DATE'19.
- [2] M. Werner et al. "Protecting the Control Flow of Embedded Processors against Fault Attacks". In: *Smart Card Research and Advanced Applications*. Springer International, 2016.
- [3] T. Chamelot et al. "SCI-FI: Control Signal, Code, and Control Flow Integrity against Fault Injection Attacks". In: DATE'22. IEEE.
- [4] L. Savary et al. "Hardware/Software Runtime for GPSA Protection in RISC-V Embedded Cores". In: DATE'25.
- [5] S. Rokicki et al. "What You Simulate Is What You Synthesize: Designing a Processor Core from C++ Specifications". In: ICCAD 2019. IEEE.
- [6] David Patterson et al. *Embench: Open Benchmarks for Embedded Platforms*. <https://github.com/embench/embench-iot>.

Accelerating LWE-Based Post-Quantum Cryptography with Approximate Computing

Diamante Simone Crescenzo¹, Emanuele Valea¹, Alberto Bosio²

¹Univ. Grenoble Alpes, CEA, List, F-38000 Grenoble, France

²Institut des Nanotechnologies de Lyon, École Centrale de Lyon, Lyon, France

Abstract—Conventional cryptographic algorithms rely on hard mathematical problems to ensure an appropriate level of security. However, with the advent of quantum computing, classical cryptographic algorithms have become vulnerable. For this reason, Post-Quantum Cryptography (PQC) algorithms have emerged, as they are designed to resist quantum attacks. Most PQC algorithms rely on the Learning With Errors (LWE) problem, where generating pseudo-random controlled errors is crucial. A well-known solution is the use of hash functions followed by error samplers, implemented according to specific error distributions, whose implementation is challenging. This paper provides a proof of concept demonstrating how Approximate Computing (AxC) can be exploited in LWE-based cryptographic algorithms to alleviate this implementation bottleneck. The main idea is to use AxC circuits to perform certain operations of the algorithm, introducing the required error for free thanks to the approximation. Our key contribution is demonstrating how AxC techniques can be effectively applied to LWE-based algorithms, highlighting a novel approach to generating and introducing the error. This concept has proven effective in an approximate implementation of the FrodoKEM algorithm, showing up to 2.19% improvement in performance.

I. INTRODUCTION

The *Learning With Errors* (LWE) problem involves recovering a secret vector from linear equations perturbed by small, structured errors, typically drawn from a discrete Gaussian or other bounded distributions. It can be represented by the notation $b = A \cdot s + e$, where A is the coefficient matrix, s is the secret vector, e is the error vector, and b is the constant vector (usually employed as the public key). Its security derives from worst-case lattice problems, such as the *Shortest Vector Problem* (SVP) [1], which remain hard to solve even for quantum algorithms. The difficulty of distinguishing perturbed linear equations from random noise is the basis of LWE-based cryptography.

Error generation poses challenges due to its complex sampling chains: a True Random Number Generator (TRNG) produces a seed, expanded via a cryptographically secure hash function (often from the NIST FIPS 202 [2] standard), then transformed into a Gaussian distribution using an ad-hoc sampler [3]. While dedicated hardware solutions are still emerging [4], *Approximate Computing* (AxC) offers an alternative by trading accuracy for efficiency, reducing area, power, and execution time [5].

This paper explores AxC as an optimization strategy for LWE-based cryptography, replacing explicit error-generation

with approximation in matrix-vector multiplication. The FrodoKEM algorithm [6] directly descends from the LWE problem, but with b , s , and e instantiated as matrices instead of vectors. For this reason, we use it as a case study. We introduce errors via AxC-based digital operators, specifically approximate adders from EvoApproxLib [7], characterizing their error distribution. A proof-of-concept implementation emulates these adders in software to ensure functional correctness, while performance results assume hardware equivalence to precise adders.

Our key contribution is demonstrating that AxC effectively introduces errors in LWE-based cryptography, achieving a 2.04% reduction in FrodoKEM’s key exchange execution time.

II. IMPLEMENTING FRODOKEM WITH AXADDERS

Our approximate FrodoKEM-640 software implementation is directly derived from the reference one¹. We acted on the sections of code involving the error generation and addition in the *key generation* and *encryption* functions. A graphical high-level overview of transitioning to an approximate version of the algorithm is available in Figure 1 for clarity. The diagram represents the standard algorithm implementation, with the greyed-out components highlighting the error generation chain that is removed due to the use of AxADD.

More specifically, all the function calls to the SHAKE128 primitive to generate the matrix e were removed, which has a direct impact on performance. To introduce the necessary error in the scheme, we acted on the inner product $A \cdot s$. Equation 1 shows the computation details for each element b_{ij} of the public matrix b . Each partial product is normally cumulated, with the exception of the last one, which is added to the partial result using the `addu16_0GN` adder from [7].

$$b_{ij} = \sum_{k=0}^n a_{ik}s_{kj} = a_{i0}s_{0j} + \dots + \underbrace{a_{in}s_{nj}}_{\text{AxADD}} \quad (1)$$

Its operation has been emulated by means of a C function call provided by the authors of [7]. Figure 2 compares the error distributions: the original FrodoKEM-640 errors (in blue) versus those from `addu16_0GN` (in orange), along with their respective Gaussian fits. This adder was selected as the most suitable from the EvoApproxLib library based on its error characteristics.

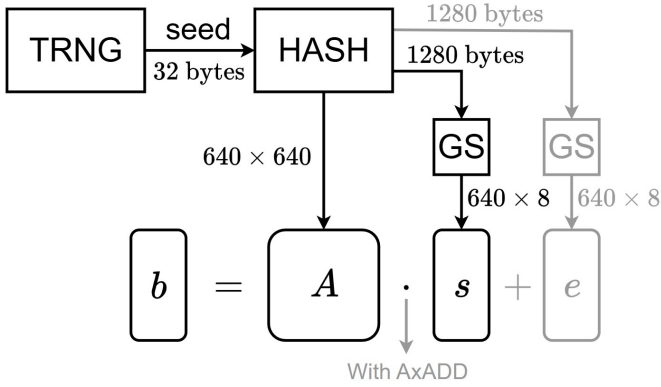


Fig. 1: Removing the error generation chain.

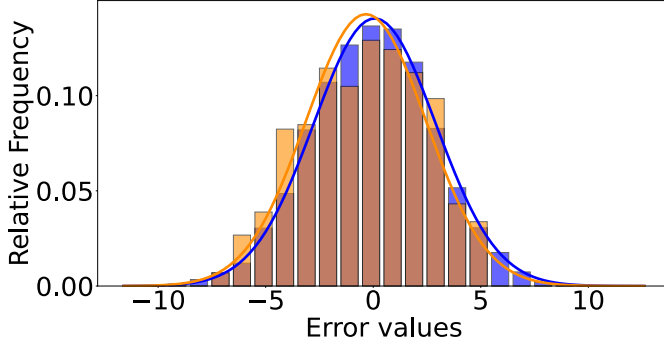


Fig. 2: Direct comparison of FrodoKEM-640 (in purple) and addu16_0GN (in orange) error distributions.

Consequently, the errors no longer require the Gaussian sampling process, resulting in further performance improvements. The rest of the algorithm follows the reference implementation but with less pseudo-random data and optimized data structures. This approximate version of FrodoKEM-640 has been finally tested against the NIST KATs to check its proper functionality and performance.

III. RESULTS

This study was performed on the reference FrodoKEM-640 C implementation¹ enriched with the modifications described in section II. The study was conducted on a Linux system with kernel version 4.18.0, an Intel Core i3-2120 CPU @ 3.30GHz, 16GB of RAM, and GCC version 8.5.0. The reported metrics are averages from multiple test vector runs, measured using the CPU Time Stamp Counter register.

Table I presents execution times (in clock cycles) for key generation, encryption, decryption, and a complete FrodoKEM run. The approximate implementation total cycles assume that an AxADD is implemented on board, replacing its execution time with that of a single operation. Parentheses indicate the percentage gain of AxADD over the reference FrodoKEM-640. The reduction in clock cycles exceeds that in generated bytes due to error generation overhead. Unlike simple random sampling, pseudo-random byte extraction from SHAKE128 requires additional function calls for setup and data retrieval,

TABLE I: AxFrodoKEM-640. Execution time in kilo clock cycles (kcc) and required number of bytes for error generation (B) for both reference implementation and with AxADD.

	KeyGen	Encrypt	Decrypt	FrodoKEM
Execution Time Reference (kcc)	55,756	72,418	73,215	201,388
Execution Time AxADD (kcc)	54,580 (-2.11%)	71,078 (-1.85%)	71,078 (-2.19%)	197,272 (-2.04%)
Error Generation Reference (B)	839,680	839,808	850,048	2,529,536
Error Generation AxADD (B)	829,440 (-1.22%)	829,440 (-1.23%)	839,680 (-1.22%)	2,498,560 (-1.22%)

followed by Gaussian sampling. As a result, FrodoKEM-640 achieves a 1.22% reduction in generated bytes, translating into a 2.04% decrease in execution time.

IV. CONCLUSION AND FURTHER PERSPECTIVES

This paper demonstrates the feasibility of using Approximate Computing (AxC) to optimize Post-Quantum Cryptography (PQC), with FrodoKEM as a proof of concept. By employing approximate adders (AxADDs) to introduce errors, we achieved a 2.19% execution time improvement and a 1.23% reduction in generated bytes. While gains in FrodoKEM are limited by the large public matrix A , this study establishes AxC as a viable method for integrating error generation into LWE-based computations, suggesting broader applicability to other schemes with different structural properties.

Beyond performance, AxC raises considerations regarding entropy reduction due to the absence of explicit random error sampling. This analysis is beyond the scope of this paper. Future work could address this by randomizing the AxADD point in the approximate matrix inner product.

In summary, AxC offers a promising approach for improving LWE-based cryptographic schemes, opening avenues for further research in this promising area.

REFERENCES

- [1] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *J. ACM*, vol. 56, Sept. 2009.
- [2] "SHA-3 standard : Permutation-based hash and extendable-output functions," tech. rep., National Institute of Standards and Technology (US), 2015.
- [3] J. Howe, A. Khalid, C. Rafferty, F. Regazzoni, and M. O'Neill, "On practical discrete gaussian samplers for lattice-based cryptography," *IEEE Transactions on Computers*, vol. 67, no. 3, pp. 322–334, 2018.
- [4] D. S. Crescenzo, R. C. Rodriguez, R. Alidori, F. Bruguier, E. Valea, P. Benoit, and A. Bosio, "Hardware Accelerator for FIPS 202 Hash Functions in Post-Quantum Ready SoCs," in *2024 IEEE 30th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, 2024.
- [5] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *2013 18th IEEE European Test Symposium (ETS)*.
- [6] E. Alkim, J. W. Bos, L. Ducas, P. Longa, I. Mironov, M. Naehrig, V. Nikolaenko, C. Peikert, A. Raghunathan, and D. Stebila, "FrodoKEM: Learning with errors key encapsulation." Submission to the NIST PQC Standardization Project, 2021. Available at: <https://frodokem.org/files/FrodoKEMspecification-20210604.pdf>.
- [7] V. Mrazek, R. Hrbacek, Z. Vasicek, and L. Sekanina, "Evoapprox8b: Library of approximate adders and multipliers for circuit design and benchmarking of approximation methods," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2017, pp. 258–261, 2017.

Low-Latency (i)FFT RTL Implementation for the FALCON Post-Quantum Signature Algorithm

Alexandre Ortega, Lilian Bossuet and Brice Colombier

Université Jean Monnet Saint-Etienne, CNRS, Institut d'Optique Graduate School, Laboratoire Hubert Curien UMR 5516, F-42023, SAINT-ETIENNE, France

{alexandre.ortega; lilian.bossuet; b.colombier}@univ-st-etienne.fr

I. INTRODUCTION

FALCON [1] is one of the three post-quantum digital signature schemes that have been recently standardized by NIST due to the future threat that quantum computers pose to classical cryptographic schemes [2]. Despite this, there is currently no full hardware register-transfer level (RTL) implementation of FALCON. One possible explanation is the rather unusual requirement for a double-precision floating-point Fast Fourier Transform (FFT) [3], which is used in FALCON to speed up polynomial multiplication. In this work, we propose a full RTL constant-time implementation of the FFT and its inverse (iFFT), on FPGA, tailored for the specific context of FALCON. Section II presents the FFT in the context of FALCON before Section III describes the proposed architecture. Afterwards, the performances of the proposed implementation are detailed and compared with previous works in Section IV. Section V concludes.

II. THE FAST FOURIER TRANSFORM IN FALCON

In FALCON, the FFT over the ring $\mathbb{Q}[x]/(\phi)$ is used with $\phi = x^N + 1$ and $N = 2^k$ a power of two. N is a security parameter of FALCON that can be equal to either 512 or 1024. Due to FALCON security requirements, IEEE-754 compliant double-precision floating-point arithmetic is being used [1]. Using the fact that FALCON polynomials are in $\mathbb{Z}[x]/(\phi)$, as well as the roots of unity symmetry in $\mathbb{Z}[x]/(\phi)$, the storage requirements can be halved and more than half of the computations can be omitted [1]. Before the optimizations, the amount of computations to perform is:

$$\#Ops = \log_2(N) \times \frac{N}{2} \quad (1)$$

After applying the optimizations, (1) becomes:

$$\#Ops = (\log_2(N) - 1) \times \frac{N}{4} \quad (2)$$

III. DESCRIPTION OF THE PROPOSED HARDWARE ARCHITECTURE

Fig. 1 shows the proposed hardware architecture. On top of a reset and clock signals, the input signals are:

- `start` is used to make the component start the computation of either the FFT or the iFFT.
- `inv` is used to choose between performing the FFT or the iFFT.

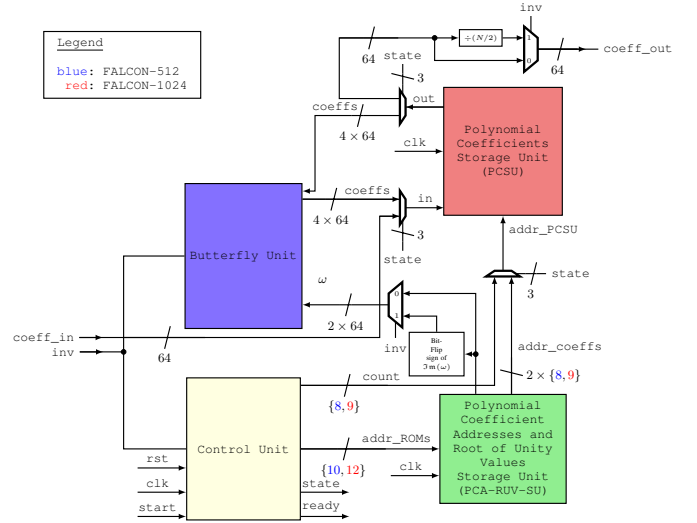


Fig. 1. Block diagram of the proposed hardware architecture of the (i)FFT for FALCON

- `coeff_in` is a 64-bit bus used to stream in the coefficients of the input polynomial to transform.

The outputs are:

- `ready` indicates that the component has finished performing the computations and is streaming out the result.
- `coeff_out` is a 64-bit bus used to stream out the coefficients of the result.

The proposed design is divided in four main blocks.

1) *The butterfly unit*: Made with three complex double-precision floating-point operators, an adder as well as a subtractor and a multiplier, it can be reconfigured dynamically to perform either the radix-2 decimation-in-time FFT or the radix-2 decimation-in-frequency iFFT.

2) *The Polynomial coefficients storage unit*: Two true dual port RAMs are used to store the polynomial coefficients. One RAM stores the real parts of the polynomial coefficients and the other RAM stores their imaginary parts. For FALCON-512, polynomials have 512 coefficients. Only the real and imaginary parts for the first half of these coefficients are stored as explained in Section II, so the two RAMs will each store 256 double-precision floating-point values with 8-bit addresses. A double-precision floating-point value is stored on 64 bits (i.e. 8 bytes). Hence, the RAMs will each store

$256 \times 8 = 2.048$ kB. An identical reasoning with FALCON-1024 gives a 9-bit address bus and two RAMs each storing $512 \times 8 = 4.096$ kB.

3) *The Polynomial Coefficient Addresses and Root of Unity Values Storage Unit*: Four single-port ROMs and one dual-port ROM are used to store the pre-computed coefficient addresses in RAM and root of unity values. For each butterfly operation, two complex coefficients and one complex root of unity are used. Two single-port ROMs are used to store the pre-computed coefficient addresses in RAM. Using (2), it is determined that 1024 butterfly operations are required for the 512-coefficient (i)FFT, and 2304 for the 1024-coefficient (i)FFT. As the RAMs each store 256 values for the 512-coefficient (i)FFT and 512 values for the 1024-coefficient (i)FFT, an 8-bit wide address bus, for the two ROMs, is required for the 512-coefficient (i)FFT and an 9-bit wide address bus is required for the 1024-coefficient (i)FFT. This gives a storage requirement of $1024 \times (8/8) = 1024 \times 1 = 1.024$ kB and $2304 \times (9/8) = 2.592$ kB respectively for the 512-coefficient and the 1024-coefficient (i)FFT.

The choice was made to only store once the 64-bit values that can be used for either the real part or the imaginary part in one dual-port ROM and to store the sequence in which those values are used in two single-port ROMs. The reason for that choice was to reduce the amount of memory required to store the values needed for the root of unity. If the root of unity values to be used are stored consecutively in a straightforward manner, which means that repetitions are possible in the dual-port ROM, $1024 \times 8 = 8.192$ kB are needed for the 512-coefficient (i)FFT and $2304 \times 8 = 18.432$ kB for the 1024-coefficient (i)FFT. If the root of unity values are stored only once along with the order in which they are accessed, 382 values need to be stored in the dual-port ROM which corresponds to $382 \times (64/8) = 3.056$ kB, and 1024 addresses coded on 9 bits in both single-port ROMs which corresponds to $2 \times 1024 \times (9/8) = 2 \times 1152 = 2.304$ kB. This means that this solution requires $3056 + 2304 = 5.36$ kB of storage capacity for the 512-coefficient (i)FFT, which is a reduction of 34.6% compared with the straightforward solution. An identical reasoning, gives a storage requirement of 11.888 kB for the 1024-coefficient (i)FFT, which represents a reduction of 35.5% compared with the straightforward solution. Hence, instead of only one dual-port ROM, two single-port and one dual-port ROMs are used to store the root of unity values.

Concerning the root of unity values for the iFFT, their real parts are the same as the root of unity for the FFT and their imaginary parts are of opposite sign. Hence, only the values for the FFT are stored. When performing the iFFT the same values are read from the ROM, but not in the same order. Indeed, to perform the FFT operations, the addresses are read consecutively starting from the highest value. To perform the iFFT operations it is the opposite, the addresses are read consecutively starting from zero. Additionally, the sign bit of the imaginary part of the root of unity value is flipped.

4) *The Control Unit*: The control unit purpose is to manage the dataflow of the (i)FFT.

TABLE I
(i)FFT-512/(i)FFT-1024 IMPLEMENTATION RESULTS

	This work		[4]	Vivado 2023.2	
Floating-point precision	Double		Double	Single	
FFT length	512	1024	512	512	1024
LUT	9658	9677	8396	1741	1793
FF	369	374	2526	3468	3508
DSP	36	36	9	10	10
BRAM	8	11	9.5	4	5
Latency (cycles)	3074	6658	19800	4589	9474

IV. RESULTS AND COMPARISONS WITH PREVIOUS WORKS

The proposed hardware implementation is described in VHDL and synthetised using AMD-Xilinx Vivado 2023.2. Table I reports the implementation results of the proposed design and compares it with the Vivado 2023.2 (i)FFT IP, and a co-design implementation of FALCON (i)FFT for the security parameter $N = 512$ [4]. The FPGA targetted, in all the reported results in Table I, is the AMD-Xilinx ZCU104+ (xczu7ev-ffvc1156-2-e) FPGA. The fairest comparison is with Mandal et al [4]. design as both design use double-precision. Both have similar metrics for the LUTs and the BRAMs. The proposed design uses $4\times$ more DSP blocks but around $6.5\times$ less FFs and clock cycles. As expected when comparing the proposed design to Vivado's IP, which uses single-precision, it uses around $2\times$ more BRAMs, more LUTs and DSPs. However, the proposed design uses around $10\times$ less FFs and achieves a lower latency.

V. CONCLUSION

A low-latency full hardware constant-time RTL implementation of the (i)FFT, tailored for FALCON parameters, was presented. It achieves the best latency of the literature among FPGA-based implementations. This work addresses one of the major difficulties reported concerning the full hardware implementation of FALCON. This work can be used as an essential building block for future hardware implementation works on FALCON.

ACKNOWLEDGMENT

This work received funding from the France 2030 program, managed by the French National Research Agency under grant agreement No. ANR-22-PETQ-0008 PQ-TLS.

REFERENCES

- [1] P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, Z. Zhang, *et al.*, "Falcon: Fast-Fourier lattice-based compact signatures over NTRU," *Submission to the NIST's post-quantum cryptography standardization process*, vol. 36, no. 5, pp. 1–75, 2018.
- [2] NIST, "Nist Announces First Four Quantum-Resistant Cryptographic Algorithms," <https://nist.gov/news-events/news/2022/07/nist-announces-first-four-quantum-resistant-cryptographic-algorithms>, 2022.
- [3] L. Beckwith, D. T. Nguyen, and K. Gaj, "High-Performance Hardware Implementation of Lattice-Based Digital Signatures," *Cryptology ePrint Archive*, Paper 2022/217, 2022.
- [4] S. Mandal and D. Roy, "Design of a Lightweight Fast Fourier Transformation for FALCON using Hardware-Software Co-Design," in *GLSVLSI'24 Proceedings*, pp. 228–232, 06 2024.

Approximate Computing for Cryptography: Leveraging FD-SOI Back-Gate Scaling in LPPN

Andrea Marenco¹, Emanuele Valea¹, Elena Ioana Vatajelu²

¹Univ. Grenoble Alpes, CEA, List, F-38000 Grenoble, France

²Univ. Grenoble Alpes, CNRS, Grenoble INP, TIMA, 38000 Grenoble, France

Abstract—The increasing use of digital communications has intensified the need for secure and cost-effective cryptographic implementations, especially in the Internet-of-Things (IoT). Post-Quantum Cryptography (PQC) schemes like Learning with Errors (LWE) and Learning Parity with Noise (LPN) rely on error distributions that are complex to generate in hardware. The Learning Parity with Physical Noise (LPPN) approach replaces explicit error sampling with controllable computational inaccuracies. This work implements LPPN on 22nm Fully Depleted Silicon-On-Insulator (FD-SOI) technology, leveraging back-gate voltage scaling to generate errors. Experimental results show that this technique significantly improves error controllability, demonstrating FD-SOI’s potential for secure PQC applications.

I. INTRODUCTION

The increasing reliance on digital communications has heightened concerns about security and privacy, especially in cost-sensitive IoT applications. Ensuring efficient and secure implementations of encryption primitives remains an engineering challenge. Hard learning problems, such as Learning with Errors (LWE) and Learning Parity with Noise (LPN), are the foundation of Post-Quantum Cryptography (PQC) [1]. These schemes rely on matrix-vector operations perturbed by noise from specific distributions, making error generation a crucial aspect. Traditional approaches require complex two-phase sampling chains, which are hard to implement securely in hardware [2].

To address this, the Learning Parity with Physical Noise (LPPN) scheme replaces external error sampling with internal computational inaccuracies [3]. First demonstrated via Over-Scaling on 65nm technology, its feasibility on advanced nodes is unclear. In this work, we implement LPPN on 22nm FD-SOI technology, exploring back-gate voltage scaling as a more precise method for error control. Experimental results from on-chip measurements demonstrate its effectiveness, highlighting FD-SOI’s potential for secure PQC implementations.

II. EXISTING IMPLEMENTATIONS OF LPPN

The core of the LPN problem is a dot product $\langle x, k \rangle$ of two N -bit binary vectors, which can be computed using bitwise AND followed by $N - 1$ XOR gates. Figure 1 shows an example of a serial hardware architecture for $N = 4$. Kamel et al. [3] introduced LPPN applying Over-Scaling to the dot product computation, inducing a controllable error by lowering the supply voltage. This occurs due to timing violations when the propagation delay causes the signal to be sampled by the output flip-flop before it has stabilized,

causing incorrect outputs with probability ϵ . This probability is shown to be proportional to the supply voltage. However, the results from [3] are based on SPICE simulations on a 65nm node. In this paper, we demonstrate the same principle on a 22nm FD-SOI process, showing that the controllability of ϵ is more complex. To address this, we propose back-gate voltage scaling as an alternative to Over-Scaling. By applying Reverse Body Biasing (RBB) to the critical path cells, we increase the threshold voltage V_T and slow down propagation, intentionally generating timing violations. This method provides a more precise control over ϵ , offering a novel way to tune error probabilities in approximate circuits.

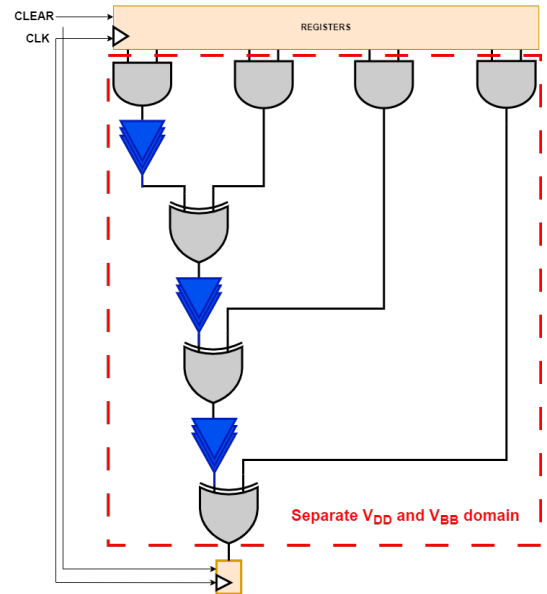


Fig. 1. Architecture for a 4-bit binary inner product, including delaying buffers at each stage of the XOR reduction chain.

III. FD-SOI-BASED LPPN IMPLEMENTATION

The LPPN module was designed at gate-level using System Verilog and synthesized with Synopsys Design Compiler. Place & Route was performed with Cadence Innovus using GlobalFoundries 22nm FD-SOI (GF22FDX) technology. A 128-bit serial architecture was chosen for cryptographic applications. As shown in Figure 1, four buffers were added between stages to meet critical path constraints, closely tied to the circuit’s clock frequency. The inner product module has a

dedicated power domain, enabling independent supply voltage Over-Scaling and back-gate voltage scaling with respect to the rest of the circuit.

The LPPN module can operate in two modes:

- **Fast Mode:** performing multiple consecutive operations without resetting the input and output registers. In this mode, errors generated by the operation i depend on the results of the operation $i - 1$.
- **Slow Mode:** it introduces a reset mechanism between consecutive operations, ensuring that each computation starts from a reset state. This reset allows to generate errors independently of the previous output value.

IV. ON-CHIP MEASUREMENTS

In order to compare the controllability of the error provided by Over-Scaling and back-gate voltage scaling, measurements were conducted on-silicon at room temperature (25°C) for the Fast Mode and Slow Mode scenarios.

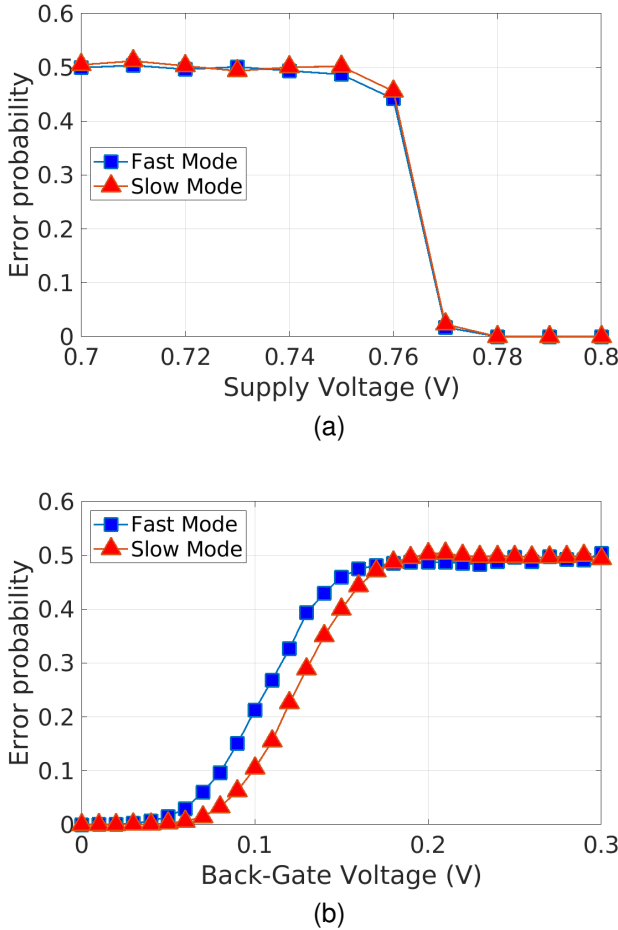


Fig. 2. On-chip measurement results. (a) Supply voltage sweep. (b) Back-gate voltage sweep.

The measurement protocol consisted in loading and clearing the needed operands depending on the scenario, executing the computation and comparing the result to the expected error-free output. Ten thousands computations were carried out for

each error point calculation. For each scenario two separate parameter sweeps were performed:

- On supply voltage V_{DD} (Figure 2a) from 0.7 V to 0.8 V with 10 mV step.
- On back-gate voltage V_{BB} (Figure 2b) from 0 V to 0.3 V with 10 mV step.

Non-sweeping parameters were kept at their nominal values of 0.8 V for V_{DD} and 0 V for V_{BB} . The nominal clock frequency, which is the maximum frequency that generates no error, was determined to be 90 MHz. The results of these scenarios show an increase in ϵ as V_{DD} decreases (or V_{BB} increases). To quantify this effect, we introduce the sensitivity of ϵ with respect to V_{DD} (or V_{BB}) denoted as S_ϵ , and defined as:

$$S_\epsilon = \max \left(\frac{\Delta \epsilon}{|\Delta V|} \right)$$

A higher S_ϵ value indicates greater difficulty in controlling the error probability with the chosen voltage scaling technique. Table I shows the sensitivity values from the sweeps. For both Fast and Slow scenarios, the sensitivity of the V_{BB} sweep is about four times smaller than that of the V_{DD} sweep, resulting in a much larger error tuning range.

TABLE I
ERROR PROBABILITY SENSITIVITY (V^{-1})

Control method		SOTA Over-Scaling (V_{DD})	This work Back-gate scaling (V_{BB})
Sensitivity*	Fast mode	23.5	5.9
	Slow mode	23.9	6.5

* Lower is better

V. CONCLUSIONS

This paper explores the implementation of the LPPN scheme on a 22nm FD-SOI process. Through on-chip measurements, we show that traditional Over-Scaling offers limited control over error probability at this technology node, raising concerns about its scalability. To address this, we propose and validate back-gate voltage scaling as a novel technique for better error control. Our results demonstrate that this approach improves error rate controllability, reducing sensitivity by up to four times compared to supply voltage scaling.

ACKNOWLEDGMENT

This work was supported by the French National Research Agency (ANR) via the CARNOT LIST AXPQC funding.

REFERENCES

- [1] C. Peikert, "A decade of lattice cryptography." Cryptology ePrint Archive, Paper 2015/939, 2015.
- [2] D. Bellizia, N. E. Mrabet, A. P. Fournaris, S. Pontié, F. Regazzoni, F.-X. Standaert, É. Tasso, and E. Valea, "Post-quantum cryptography: Challenges and opportunities for robust and secure hw design," in *2021 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, pp. 1–6, 2021.
- [3] D. Kamel, F.-X. Standaert, A. Duc, D. Flandre, and F. Berti, "Learning with physical noise or errors," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 5, pp. 957–971, 2020.

Characterization and modelling of the thermal behavior of 55nm SiGe HBT

A. Sarafinof¹, C. Mukherjee¹, M. De Matos¹, F. Cacho², C. Maneux¹

¹IMS Lab, University of Bordeaux, Talence, France

²STMicroelectronics, Crolles, France

Abstract—In this paper, the thermal model of 55 nm SiGe HBT is investigated. Subsequently, we use the S-Parameters technique to extract the thermal impedance. DC and pulsed measurements are performed to obtain the parameters representative to the thermal behaviour. Afterwards, we present the comparison of our approach on measured data from two SiGe HBT technologies from STMicroelectronics, B55x and B55.

Index Terms—SiGe HBTs, thermal impedance, thermal network, self-heating, S-Parameters.

I. INTRODUCTION

The miniaturization of the electronic devices and the increase of the power performance demand at high frequency due to the 5G and 6G requirements [1] lead to push HBT operation closer to their Safe Operating Area (SOA) limit [2]. This limit is defined by multiple limiting mechanisms such as the self-heating and the impact ionization defining the limit beyond which the device starts to degrade and become unstable. STMicroelectronics has enhanced its SiGe HBT technology [3], to face the 6G requirements, this new generation is called B55X [4]. The main upgrade of the device is related to his architecture allowing the heat to be dissipated more efficiently through the backend of line.

In this paper, the first part is dedicated to the thermal characterization to model the impact of this new device architecture on the thermal equivalent circuit. In a second phase, the device has been submitted to some pulsed measurements and compared to simulations .

II. THERMAL CHARACTERIZATION

A. Thermal Resistance

To extract the thermal resistance, the intersection method is used. By measuring Gummel plots for two different V_{CE} values and two different ambient temperatures T_{AMB} , one collector current $I_{C,int}$ is extracted [5]. Then, using the equation (1), the thermal resistance is extracted. Among all the dimensions studied, ranging from emitter area $A_E = 0.2 \times 0.6 \mu m^2$ to $A_E = 0.4 \times 5 \mu m^2$, the results shown in this paper are related to $A_E = 0.2 \times 5 \mu m^2$, which is considered as the main dimension.

$$R_{TH} = \frac{T_{amb,2} - T_{amb,1}}{I_{C,int}(V_{CE,1} - V_{CE,2})} \quad (1)$$

By observing the figure 1 comparing the thermal resistance of the two devices generations, the B55X shows a reduction of close to 40% compared to the B55. This RTH decrease is due to the B55X architecture optimized to reduce the self-heating thanks to the reduced thinness of the SSTI, lower than 100nm, allowing to expand the heat dissipation cone [6].

B. Thermal impedance

To extract the thermal impedance (Z_{TH}), the S-Parameters measurements are performed [7]. These measurements were achieved between 30kHz and 3GHz since the thermal behaviour appears dominant at low frequencies compared to the electrical one. To extract Z_{TH} , the equation (2) is used with Y_{ij}^{DC} , the value for the frequency close to 0 and Y_{ij}^{AC} , the Y-parameters corresponding to isothermal conditions.

The figure 2 shows the plot of the magnitude of Z_{TH} . Two differences can be noted, the first one is related to the low frequency (≈ 105 Hz) value of each plot which represents the R_{TH} value. The second difference is the decrease behaviours as a function of the frequency. These different frequency behaviours are representative of the differences of heat dissipation through the device and therefore the architectural difference.

C. Thermal network

The thermal network used in the current HiCuM simulation includes only one R_{TH} - C_{TH} cell representing a single pole. A more accurate thermal pole representation is a network with three cells called Cauer Network as described on the figure 3. The Cauer network is an electrical representation of the different parts of the device architecture in terms of thermal representation.

The figure 4 permit to observe the extraction of this network. The objective is to match the plot from the equation of the network with the measurement plot. The magnitude of the simulated 3-cell thermal network is close to the measurement demonstrating a better representation than the previous 1-cell network proving the accuracy of the Cauer network. Another evidence of the accuracy of the 3-cell model compared to the single cell model is the pulsed measurement described in the next section.

III. PULSED MEASUREMENT AND SIMULATIONS

The difference between the two different network in simulation can be observe by performing transient simulations. To observe these differences, a pulsed voltage is sent to Base Emitter junction from 0.7V to 0.9V while keeping constant the Collector Emitter voltage at 1.5V [9]. Similarly, HiCuM simulations are performed with the 1-cell thermal network and with the modified thermal network with three cells.

The results of the different simulations compared to measurements are presented in the figure 5. The normalize current reach 1 at I_C equal to his maximum value. The main difference with the modified network is the I_C values during the rise of the pulse. The network described by the three-pole thermal model shows better accuracy than 1-cell network confirming the S-Parameters analysis. The junction temperature (T_j) is also simulated to compare the two networks. The figure 6 represent the junction temperature maximum for different pulse widths. The temperature is really impacted by the pulse

width because at low pulse width, the I_C does not reach his steady state and T_J does not reach the DC value.

IV. CONCLUSION

For the first time, SiGe HBT B55X from STMicroelectronics has been characterized thermally. The thermal characterization shows a better heat dissipation compared to the precedent device generation resulting in a lower junction temperature thanks to a 40 % lower thermal resistance. The thermal behavior has been simulated with the extraction of the thermal network and the integration in the HiCuM model. Two network has been compared and the Cauer network is more accurate to the measurement than the 1-cell network. The next step is performing dynamic stress measurements to study their degradation mechanisms depending on the stress types.

REFERENCES

- [1] E. C. Strinati et al., in 2022 EuCNC/6G Summit, june 2022, p. 423-428. doi: 10.1109/EuCNC/6GSummit549.41.2022.9815700.
- [2] A. J. Scholten, in 2020 IEEE BCICTS, nov. 2020, p. 1-8. doi: 10.1109/BCICTS48439.2020.9392953.
- [3] P. Chevalier et al, in 2014 IEEE International Electron Devices Meeting, dec. 2014. doi: 10.1109/IEDM.2014.7046978.
- [4] P. Chevalier et al., in 2024 IEEE BCICTS, oct. 2024, p.13-17. doi: 10.1109/BCICTS59662.2024.10745707.
- [5] M. Couret, "Failure mechanisms implementation into SiGe HBT compact model operating close to safe operating area edges", dec. 2020.
- [6] C. Mukherjee et al., IEEE Trans. Electron Devices, vol.66, n°5, may 2019. doi: 10.1109/TED.2019.2906979.
- [7] A. K. Sahoo et al., IEEE Electron Device Lett., vol.32, n°2, feb. 2011. doi: 10.1109/LED.2010.2091252.
- [8] S. Fukunaga et T. Funaki, NOLTA, vol.11, n°2, 2020. doi: 10.1587/nolta.11.157.
- [9] M. Couret et al., 2019 IEEE 32nd ICMTS, mar. 2019. doi: 10.1109/ICMTS.2019.8730964.

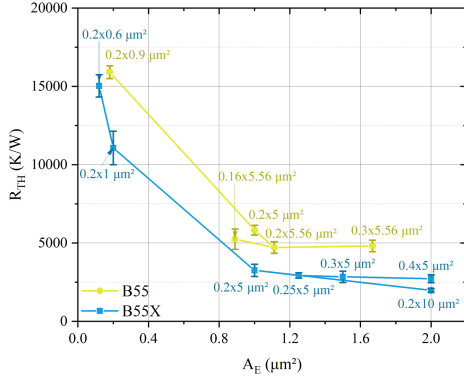


Fig. 1: Thermal resistance with their error bars versus emitter area according to different emitter drawn dimensions of B55 and B55x

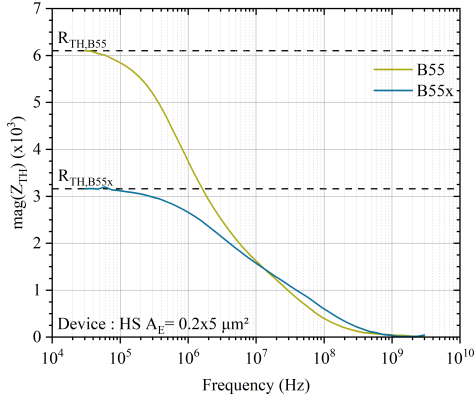


Fig. 2: Thermal impedance versus frequency for $0.2 \times 5 \mu\text{m}^2$ emitter area of B55 and B55x

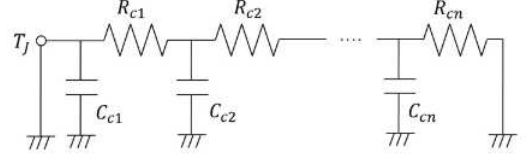


Fig. 3: Electrical equivalent circuit of Cauer thermal network [8]

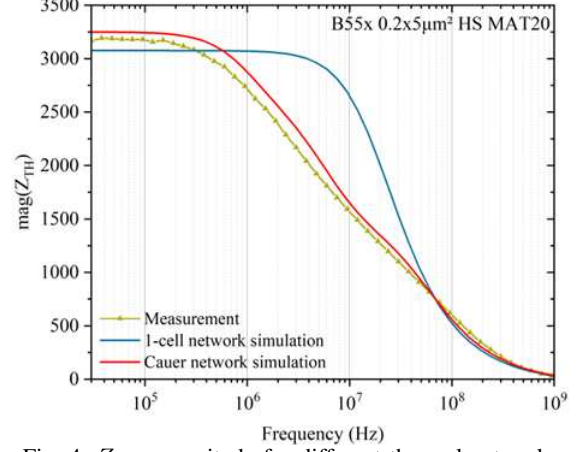


Fig. 4: Z_{TH} magnitude for different thermal networks

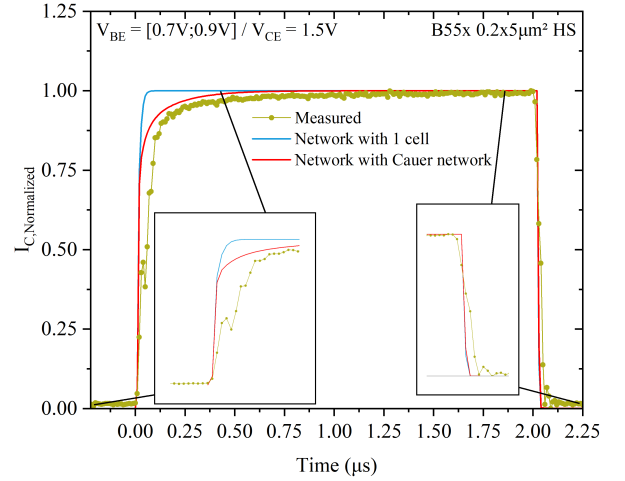


Fig. 5: Normalized collector current response as a function of time comparing measurement and simulations

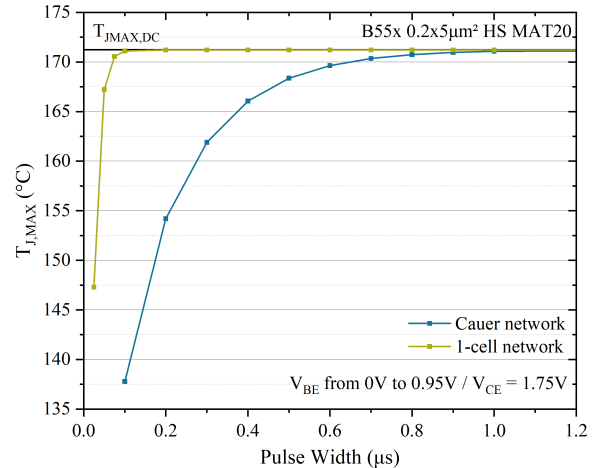


Fig. 6: Simulation of $T_{J,MAX}$ versus the pulse width for specific bias condition

Vers une électronique embarquée pour l'instrumentation nucléaire et l'identification de particules

Julien Portanguen^{1,2}, Gwenolé Corre¹, Yoann Moline¹, Jean-Michel Bourbotte¹,
Victor Buridon¹, Wilfrid Husson¹, Martial Owczaruk¹, Vincent Métivier²

¹Université Paris-Saclay, CEA, List, F-91120 Palaiseau, France

²SUBATECH, IMT Atlantique, Nantes Université, CNRS/IN2P3, F-44000 Nantes, France

Contact : julien.portanguen@cea.fr

Résumé—Dans l'instrumentation nucléaire, le photodétecteur le plus répandu pour la scintillation est le tube photomultiplicateur (PMT). Une alternative est apparue il y a quelques années avec les photomultiplicateurs au silicium, ou SiPM. La discrimination des particules pour la non-prolifération nucléaire, la sécurité ou la sûreté reste un domaine dans lequel les SiPM sont moins efficaces que les PMT, malgré leur plus grande compacité et durabilité. Pour garantir les performances de discrimination de particules, il est essentiel de préserver les caractéristiques intrinsèques du signal. Les travaux menés ont cherché à obtenir un modèle complet du SiPM pour simuler leurs réponses à des interactions gammas et neutrons du scintillateur jusqu'à l'étape de numérisation. Un script Python a permis de générer des modèles SPICE de SiPM pour un nombre de détecteurs et des caractéristiques variables. Des expériences de scintillation ont fourni des données d'entrée réalistes pour les temps d'arrivée des photons sur le photodétecteur. Enfin, des comparaisons d'architectures de préamplification ont montré un impact sur les capacités de discrimination du SiPM.

Index Terms—Modèle électrique équivalent, discrimination neutron-gamma, scintillateur, identification de particules, *silicon photomultiplier* (SiPM), *simulation program with integrated circuit emphasis* (SPICE)

I. INTRODUCTION

L'instrumentation nucléaire répond à diverses applications telles que l'imagerie médicale et astronomique, la sûreté ou la sécurité. Les rayonnements ionisants déposent aléatoirement leur énergie dans les détecteurs au cours du temps sous forme d'impulsions. L'objectif est de mesurer les caractéristiques de ces impulsions qui varient en fonction de l'application. Notre cadre applicatif s'intéresse à la discrimination de particules à l'aide de systèmes portables pour la caractérisation des sources ou la radioprotection [1]. Nous réalisons des mesures avec un scintillateur plastique sur un photomultiplicateur silicium (SiPM) dans l'objectif de concevoir un système embarqué. Ces détecteurs génèrent une lumière dans le domaine visible à la suite d'une interaction avec un rayonnement ionisant. Les tubes photomultiplicateurs (PMT) et les SiPM sont les deux technologies utilisées pour la collecte des photons. Les SiPM autorisent des systèmes plus compacts et mécaniquement plus durables que ceux utilisant des PMT. Malgré un gain et un bruit sensibles à la température, les SiPM nécessitent une

tension de fonctionnement plus faible que les PMT, typiquement 30 V contre 1000 V, et ne sont pas affectés par les champs magnétiques ni détruits par la lumière directe [2]. Un signal généré par l'ensemble scintillateur et SiPM aura généralement une décroissance de quelques centaines de nanosecondes et une amplitude de quelques millivolts. Dans notre cas, nous essayons de séparer les interactions γ des interactions neutroniques. La distinction entre ces impulsions réside dans la forme de l'impulsion et est l'une des plus difficiles parmi les différentes particules ionisantes car la différence est ténue. Des impulsions normalisées de scintillateur plastique sont présentées en figure 1. La discrimination des particules dans les systèmes embarqués est généralement réalisée par la méthode *Pulse Shape Discrimination* (PSD) [3]. Cette méthode repose sur le calcul de deux intégrales appelées par la suite Q_{tail} et Q_{total} . Ces intégrales sont calculées pour chaque impulsion en fixant l'instant de début des portes d'intégration et leur durée. Le ratio *Tail-To-Total* (TTT) prend en compte ces deux intégrations de la manière suivante :

$$\text{TTT-ratio} = \frac{Q_{\text{tail}}}{Q_{\text{total}}}. \quad (1)$$

Cette technique est facile à mettre en œuvre sur des architectures intégrées comme des FPGA. Ce papier propose de définir des modèles de simulation de SiPM qui permettent une comparaison avec les PMT en s'appuyant sur des données d'entrée réalistes afin d'en conserver le sens physique. La section II décrit le protocole expérimental et les résultats ayant permis l'obtention de ces données. La section III décrit la génération du modèle SPICE de SiPM. Enfin la section IV décrit succinctement les travaux en cours sur l'électronique pour les SiPM dont certains résultats ont été soumis récemment.

II. ACQUISITION DE DONNÉES D'ENTRÉE

Pour notre application, nous avons besoin de données de scintillation réalistes. Nous avons mené des expériences avec du Cs-137 (émission γ) et du Cf-252 (émissions neutroniques et γ). Pour obtenir ces impulsions, nous avons mis en place un banc d'essai composé d'un PMT H11284-MOD Hamamatsu et d'un scintillateur plastique EJ-276 d'Eljen Technology. Des

études internes ont montré que ce PMT était le meilleur candidat pour la discrimination n/γ . L'EJ-276 est l'un des scintillateurs plastiques classiquement utilisés dans la littérature et fournit un point de départ cohérent pour la discrimination. Les impulsions collectées sont traitées en Python a posteriori. Nous avons ajusté la triple décroissance exponentielle du scintillateur convolué par la fonction de transfert gaussienne du PMT. La chaîne de mesure combine ces deux contributions, qui forment une gaussienne modifiée exponentielle. Ces résultats sont comparés aux données du fabricant. Par concision, tous ces résultats seront présentés lors du colloque du GDR SOC².

III. MODÈLES POUR LA SIMULATION ÉLECTRIQUE

Les codes Monte Carlo tels que Geant4, MCNP ou PHITS sont adaptés à la simulation de la physique des scintillateurs. Ils fournissent une représentation de la probabilité d'interaction et de l'énergie déposée associée. Cependant, les modèles SiPM ont encore leurs limites pour la photodétection. Les premiers étaient axés sur le modèle équivalent électrique de la diode à avalanche pour photon unique (SPAD) [4]. La figure 2 montre un modèle SPICE couramment utilisé dans la littérature [5] et sur lequel le développement s'est basé. Ces modèles ont été appliqués exclusivement à des sorties de photons uniques (ou multiples simultanés) à des fins de validation du modèle. Des modèles plus complexes ont ensuite pris en compte un plus grand nombre de photons, survenant à différents moments [6], [7]. Ces modèles ont été conçus pour les besoins de la spectrométrie γ , et nous voulons maintenant aller plus loin avec des objectifs de discrimination de la forme de l'impulsion. Un script Python a été développé pour écrire la netlist automatiquement. Cette approche permet de modifier tous les paramètres pour configurer n'importe quel SiPM. Le nombre de cellules actives et totales, l'impédance de quenching, l'impédance équivalente ou l'impédance de lecture peuvent être modifiées. Le modèle SPICE créé peut activer

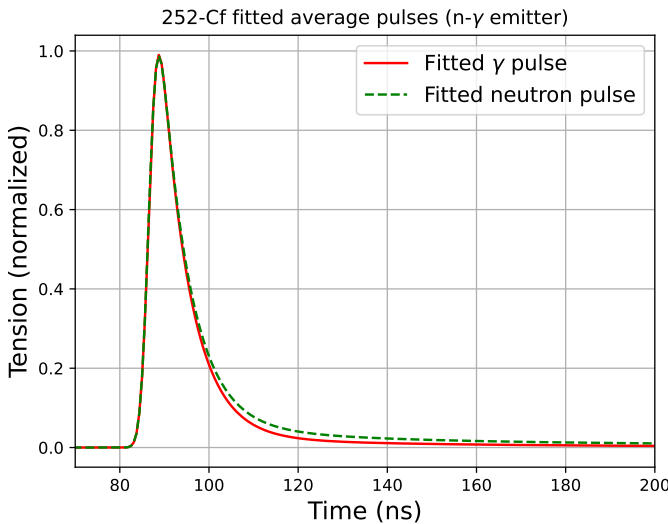


FIGURE 1. Impulsions moyennes γ et neutron en tension en sortie d'un ensemble scintillateur et PMT (ajustées)

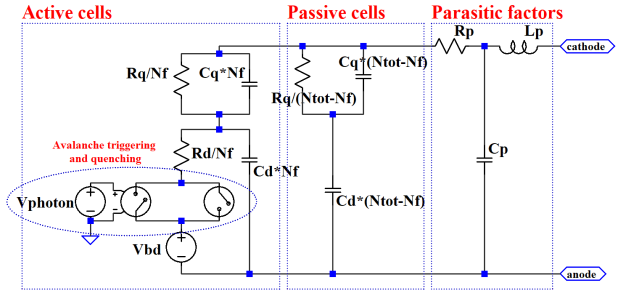


FIGURE 2. Modèle électrique équivalent de SiPM [5] pour N_f cellules actives et $N_{tot} - N_f$ cellules passives. R_Q et C_Q représentent l'impédance de quenching, R_D et C_D le modèle équivalent de diode, V_{photon} déclenche l'avalanche, V_{bd} est la tension d'avalanche, R_P , C_P et L_P représentent les éléments parasites.

les SPAD à différents moments grâce à un fichier d'entrée. Comme le modèle ne peut pas gérer les phénomènes physiques stochastiques, ceux-ci sont inclus dans ce fichier d'entrée. Nous simulons également les effets secondaires induits par le SiPM, tels que les courants d'obscurité, la diaphonie ou les impulsions secondaires de cette façon. Cette approche permet de simuler n'importe quelle stimulation photonique en utilisant la spécification technique du SiPM. Par concision, ces résultats seront présentés lors du colloque.

IV. CONCLUSION ET PERSPECTIVES

Les travaux présentés permettent de définir la capacité théorique atteignable de discrimination d'un SiPM. Ce travail préliminaire a été réalisé en vue de la conception d'une électronique spécifique (ASIC) pour la discrimination n/γ . Nous travaillons actuellement sur une étude comparative de préamplificateurs pour optimiser les performances de discrimination n/γ en utilisant des SiPM.

RÉFÉRENCES

- [1] J. Adams and G. White, "A versatile pulse shape discriminator for charged particle separation and its application to fast neutron time-of-flight spectroscopy," *Nuclear Instruments and Methods*, vol. 156, no. 3, pp. 459–476, 1978.
- [2] E. Roncali and S. R. Cherry, "Application of silicon photomultipliers to positron emission tomography," *Annals of biomedical engineering*, vol. 39, pp. 1358–1377, 2011.
- [3] M. Grodzicka-Kobylka, T. Szczesniak, M. Moszyński, K. Brylew, L. Swiderski, J. Valiente-Dobón, P. Schotanus, K. Grodzicki, and H. Trzaskowska, "Fast neutron and gamma ray pulse shape discrimination in ej-276 and ej-276g plastic scintillators," *Journal of Instrumentation*, vol. 15, no. 03, p. P03030, 2020.
- [4] F. Corsi, A. Dragone, C. Marzocca, A. Del Guerra, P. Delizia, N. Dinu, C. Piemonte, M. Boscardin, and G. F. Dalla Betta, "Modelling a silicon photomultiplier (sipm) as a signal source for optimum front-end design," *Nuclear Instruments and Methods in Physics Research Section A : Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 572, no. 1, pp. 416–418, 2007.
- [5] F. Acerbi and S. Gundacker, "Understanding and simulating sipms," *Nuclear Instruments and Methods in Physics Research Section A : Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 926, pp. 16–35, 2019.
- [6] K. D. McGee, "Silicon photomultiplier modeling of csi (tl) with front end electronics using a monte carlo model," 2019.
- [7] A. K. Jha, H. T. Van Dam, M. A. Kupinski, and E. Clarkson, "Simulating silicon photomultiplier response to scintillation light," *IEEE transactions on nuclear science*, vol. 60, no. 1, pp. 336–351, 2013.

Enhancing Keystone Security Against Cache Timing Attacks: A Modular Approach

Oussama Elmnaouri*, Pascal Cotret*, Vianney Lapôte†, Loïc Lagadec*

* Lab-STICC, UMR CNRS 6285, ENSTA (29806 Brest Cedex 9, France)

firstname.lastname@ensta.fr

† Lab-STICC, UMR CNRS 6285, Université de Bretagne-Sud (56100 Lorient, France)

vianney.lapotre@univ-ubs.fr

Abstract—Confidential computing includes various methods to enhance data security, notably by processing sensitive information within Trusted Execution Environments (TEEs). However, TEEs remain vulnerable to Side-Channel Attacks (SCAs), such as cache timing attacks, which exploit timing variations to extract confidential data. Existing TEE designs do not provide sufficient protection against these threats, highlighting the need for stronger security measures. This study focuses on integrating countermeasures specifically targeting timing and cache vulnerabilities within a TEE. The implementation will leverage the RISC-V architecture to explore its potential in mitigating SCA within TEE.

Index Terms—Computer Architecture, Confidential Computing, Hardware Security, TEEs, SCAs.

I. INTRODUCTION

Trusted Execution Environments (TEEs) are essential for protecting sensitive data by providing isolated environments that guarantee both data confidentiality and integrity. Notable TEEs include proprietary solutions like ARM TrustZone [1], Intel SGX [2] and AMD SEV [3], as well as open source solutions such as Keystone [4] and Penglai [5] for RISC-V. While TEEs offer strong protection against numerous software attacks, they remain vulnerable to cache-based and timing side-channel attacks. These attacks exploit variations in execution time or cache access patterns to extract sensitive data, posing a major challenge to TEE security.

Microarchitectural cache timing attacks, such as Prime+Probe, Flush+Reload, and Evict+Time [6], pose significant challenges to the security of TEEs [7] as they can leak sensitive data by analyzing cache accesses. Various countermeasures have been proposed to mitigate these risks [6], but each comes with trade-offs, some cause high performance overhead, while others do not fully protect against all types of side-channel attacks.

II. THREAT MODEL

The threat model considers that applications executing in both the Rich Execution Environment (REE) and the TEE are vulnerable to cache-based and transient execution side-channel attacks [8]. The attacker is any application running on the system that shares the cache with the victim. This

The work presented in this paper was realized in the frame of the SCAMA project number ANR-23-CE39-0011, supported by a grant of the French National Research Agency (ANR).

includes both untrusted applications in the REE and potentially malicious applications in the TEE. We do not consider physical attacks (e.g., fault injection, power analysis).

III. KEYSTONE STUDY

Keystone [4] is an open-source TEE designed for RISC-V processors, combining both security concepts from ARM TrustZone and Intel SGX to establish a clear separation between two execution domains. The non-secure world operates under normal processor conditions, running the untrusted operating system and normal applications while the secure world ensures the execution of sensitive applications, isolated from both the operating system and other applications (sensitive and normal). Keystone is based on a Security Monitor (SM) that manages the entire lifecycle of enclaves and ensures a secure communication between the two worlds by using the RISC-V Physical Memory Protection (PMP) [9] to enforce memory isolation, ensuring that confidential data and enclaves are safeguarded against unauthorized access, providing a flexible and efficient security framework. Figure 1 presents the architecture of a secure system based on enclaves, highlighting the different privilege levels and the separation between trusted and untrusted worlds.

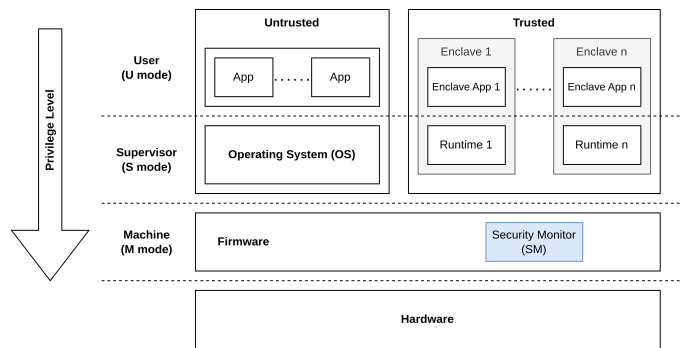


Fig. 1. Keystone Architecture: Secure and Non-Secure World Isolation (adapted from [4]).

IV. COUNTERMEASURES ON KEYSTONE

Various countermeasures have been developed to mitigate cache timing side-channel attacks [6], each addressing different vulnerabilities:

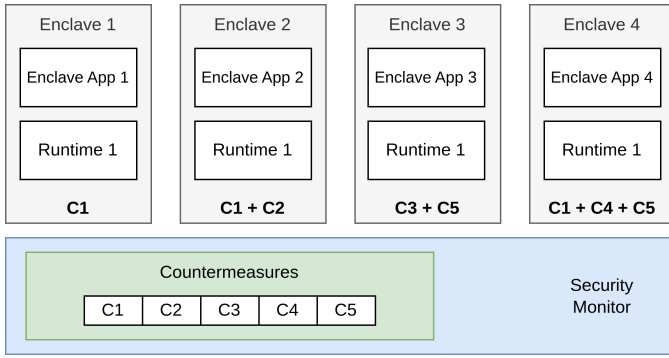


Fig. 2. Flexible Enclaves with Customizable Countermeasures library.

- **Constant-time execution:** Ensures that both cache access patterns and control flow remain independent of secret data, preventing any information leakage through cache timing attacks. This technique is widely used in cryptographic algorithms like AES to prevent key leakage due to execution time variations [10].
- **Noise injection:** Introduces randomness in timing measurements or accesses to shared resources to obscure variations exploitable by an attacker, thereby preventing the leakage of sensitive information. This technique is widely used in cryptographic implementations, real-time systems, and secure embedded devices [11], [12].
- **Enforcing Determinism:** Eliminates execution time variations that could be exploited by an attacker to extract sensitive information through timing channels. This technique is widely integrated in debugging frameworks, cloud computing, and virtual machines security [13], [14].
- **Time Partitioning:** Controls access to shared resources over time to prevent cache timing-based attacks. This is achieved through techniques that influence concurrent resource access and program transitions. This approach is widely used in countermeasures such as Cache Flushing [15], Lattice Scheduling [16], and Execution leases [17].
- **Hardware Partitioning:** Isolates hardware resources to prevent cache side-channel attacks by ensuring that each process has its dedicated space. This technique is widely used in countermeasures such as cache locking [18], Cache Coloring [19], and Quasi-Partitioning [20].

This work aims to develop a flexible and modular framework within Keystone. Although existing research primarily addresses individual mitigation techniques, our approach integrates these countermeasures into a unified framework, thereby providing users with enhanced flexibility to selectively enable the protections that best align with their specific security and performance requirements. One possible direction is to integrate these protections within the SM to enhance enclave lifecycle management and security [21]. However, this remains an open question, and further analysis is required to assess the feasibility, trade-offs, and effectiveness of such an approach.

Figure 2 illustrates how we envision the Keystone structure with the integration of these security techniques and protections.

REFERENCES

- [1] B. Ngabonziza, D. Martin, A. Bailey, H. Cho, and S. Martin, "Trustzone explained: Architectural features and use cases," in *CIC*, pp. 445–451, 2016.
- [2] V. Costan and S. Devadas, "Intel SGX explained." Cryptology ePrint Archive, Paper 2016/086, 2016.
- [3] I. Advanced Micro Devices, "Amd sev-snp: Strengthening vm isolation with integrity protection and more," tech. rep., Advanced Micro Devices, Inc., January 2020. White Paper.
- [4] D. Lee, D. Kohlbrenner, S. Shinde, K. Asanović, and D. Song, "Keystone: an open framework for architecting trusted execution environments," in *EuroSys*, EuroSys '20, (New York, NY, USA), Association for Computing Machinery, 2020.
- [5] E. Feng, X. Lu, D. Du, B. Yang, X. Jiang, Y. Xia, B. Zang, and H. Chen, "Scalable memory protection in the PEngLAI enclave," in *USENIX OSDI*, pp. 275–294, USENIX Association, July 2021.
- [6] Q. Ge, Y. Yarom, D. Cock, and G. Heiser, "A survey of microarchitectural timing attacks and countermeasures on contemporary hardware," *Journal of Cryptographic Engineering*, vol. 8, pp. 1–27, 04 2018.
- [7] M. Ghaniyoun, K. Barber, Y. Xiao, Y. Zhang, and R. Teodorescu, "Teesecc: Pre-silicon vulnerability discovery for trusted execution environments," in *ISCA*, ISCA '23, (New York, NY, USA), Association for Computing Machinery, 2023.
- [8] W. Wang, "Side channel risks in hardware trusted execution environments (tees)," in *Side Channel Risks in Hardware Trusted Execution Environments (TEEs)*, May 2019. Presented at a research event.
- [9] R.-V. International, *The RISC-V Instruction Set Manual: Volume II: Privileged Architecture, Version 20240411, Apr. 2024, Section 3.7.*, April 2024.
- [10] D. A. Osvik, A. Shamir, and E. Tromer, "Cache attacks and countermeasures: the case of aes," in *CT-RSA*, CT-RSA'06, (Berlin, Heidelberg), p. 1–20, Springer-Verlag, 2006.
- [11] W.-M. Hu, "Reducing timing channels with fuzzy time," in *RISP*, pp. 8–20, 1991.
- [12] E. Brickell, "Technologies to improve platform security," in *CHES - Invited Talk*, (Nara, Japan), Intel Corporation, September 2011.
- [13] W. Wu, E. Zhai, D. Jackowitz, D. Wolinsky, L. Gu, and B. Ford, "Warding off timing attacks in deterland," *ArXiv*, vol. abs/1504.07070, 2015.
- [14] A. Aviram, S.-C. Weng, S. Hu, and B. Ford, "Efficient system-enforced deterministic parallelism," *Commun. ACM*, vol. 55, p. 111–119, May 2012.
- [15] Y. Zhang and M. K. Reiter, "Düppel: retrofitting commodity operating systems to mitigate cache side channels in the cloud," in *CCS*, CCS '13, (New York, NY, USA), p. 827–838, Association for Computing Machinery, 2013.
- [16] D. E. Denning, "A lattice model of secure information flow," *Commun. ACM*, vol. 19, p. 236–243, May 1976.
- [17] M. Tiwari, X. Li, H. M. G. Wassel, F. T. Chong, and T. Sherwood, "Execution leases: A hardware-supported mechanism for enforcing strong non-interference," in *2009 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 493–504, 2009.
- [18] Z. Wang and R. B. Lee, "New cache designs for thwarting software cache-based side channel attacks," *SIGARCH Comput. Archit. News*, vol. 35, p. 494–505, June 2007.
- [19] T. Kim, M. Peinado, and G. Mainar-Ruiz, "STEALTHMEM: System-Level protection against Cache-Based side channel attacks in the cloud," in *USENIX Security*, (Bellevue, WA), pp. 189–204, USENIX Association, Aug. 2012.
- [20] Z. Zhou, M. K. Reiter, and Y. Zhang, "A software approach to defeating side channels in last-level caches," in *CCS*, CCS '16, (New York, NY, USA), p. 871–882, Association for Computing Machinery, 2016.
- [21] S. Nashimoto, R. Ueno, and N. Homma, "Comparative analysis and implementation of jump address masking for preventing tee bypassing fault attacks," in *ARES*, ARES '24, (New York, NY, USA), Association for Computing Machinery, 2024.

Verilog-A Compact Model of Ferroelectric Memory Devices For Compute-In-Place Applications

Poovendran Muthusamy, Chhandak Mukherjee, Marina Deng, Cristell Maneux, François Marc
IMS Laboratory, University of Bordeaux, France.

Abstract—In the rapidly evolving field of nanoelectronics, ferroelectric memory devices have emerged as promising candidates for advancing compute-in-place (CIP) applications. By leveraging their unique characteristics, such as nonlinearity and hysteresis, these devices offer the potential to enhance memory architectures and improve computational efficiency. This research introduces a compact, current-based model for the ferroelectric capacitor (FeCAP), specifically developed to capture its complex behaviour and enable optimized performance in CIP applications. Implemented in Verilog-A, the model provides researchers and engineers with a powerful tool for simulating and designing energy-efficient memory architectures.

Index Terms—Ferroelectric Memory, Verilog-A Modelling, Compute-In-Place.

I. INTRODUCTION

The rising demands in modern nanoelectronics call for memory architectures that can overcome Von Neumann bottleneck and energy inefficiencies. Compute-In-Place architectures, which integrate processing modules directly with the memory [1], offer a promising approach by reducing data transfer delay and computational load for the Micro Controller Unit (MCU), enhancing speed and energy efficiency. Ferroelectric memory devices have attracted considerable attention for these applications because of their unique properties: non-linearity, hysteresis, and non-volatility, which enable rapid switching and improved retention [2]. However, modelling these complex behaviours remains challenging. To address this, we introduce a compact, current-based model for ferroelectric memory devices, implemented in Verilog-A. This current based model can accurately capture dynamic ferroelectric behaviour and has been previously validated through extensive simulations reproducing the behaviour of experimental data [3], paving the way for its integration into energy-efficient, high-performance computing systems. This work shows a new implementation of this model in Keysight ICCAP environment suitable for device compact modelling using Verilog-A description language.

II. FERROELECTRIC MEMORY DEVICE

A. Ferroelectric Memory Field Effect Transistor

As illustrated in Fig. 1, the ferroelectric capacitor is integrated directly on top of the CMOS gate to form Ferroelectric Memory Field Effect Transistor (FeMFET). This structure

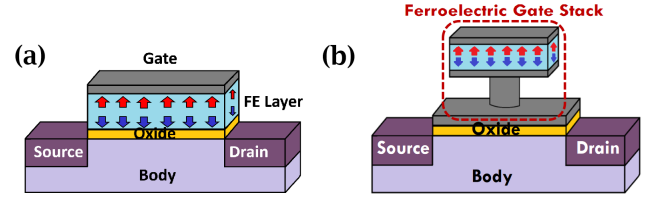


Fig. 1. (a) Ferroelectric Field Effect Transistors (FeFET) where FE layer is stacked in CMOS gate, (b) Ferroelectric Memory Field Effect Transistor (FeMFET), formed by the integration of FeCAP on top of CMOS gate.

eliminates the interface layer that lies between the gate oxide and the ferroelectric material in the Ferroelectric Field Effect Transistors (FeFET)s — a layer often linked to increased imprint effects that can degrade device performance over time [4]. Eliminating this layer can significantly enhance endurance of the device. This improved endurance supports compute-in-place architectures by ensuring reliable operation during frequent switching cycles.

III. COMPACT MODELLING

A. Current based Preisach model

The current-based model captures the complex dynamics of the FeCAP by considering three main components of the current: ferroelectric current (I_{FE}), leakage current ($I_{leakage}$) and dielectric current (I_{DE}) [5]. Among these, the ferroelectric current is modelled using the Lorentzian function, which provides an accurate representation of its behaviour under varying electrical conditions. Furthermore, the Polarization-Electric field (P-E) hysteresis loop is derived from the integration of the current, enabling detailed insights into the non-linear and hysteric properties of the FeCAP.

B. Expressions and parameters used

In the following, the key equations and parameters used in the model are presented. The total current (I_{Total}) is modelled as the sum of the three components as listed above is expressed in equation (1).

$$I_{Total} = I_{FE} + I_{DE} + I_{leakage} \quad (1)$$

Since leakage is negligible in ferroelectric devices, $I_{leakage}$ is approximated to zero. In equation (2), surface area of

the capacitor is represented by S whereas t_{fe} denotes the ferroelectric layer thickness, while C_{DE} is the base capacitance and V_{off} is the offset voltage of FeCAP.

$$I_{DE} = C_{DE} \cdot \frac{dV}{dt} \quad \text{with} \quad C_{DE} = \frac{\epsilon_0 \cdot \epsilon_R \cdot S}{t_{fe}} \quad (2)$$

The rate of change of polarization with respect to the electric field is given by dP/dE . The ferroelectric current is modelled using a Lorentzian function characterized by its amplitude (A) and width (w), with π used as a normalization factor as shown in equations (3) & (4).

$$\frac{dP^\pm}{dE} = \frac{2 \cdot A^\pm \cdot w^\pm}{4\pi((E - E_c^\pm)^2 + (w^\pm)^2)} \quad (3)$$

$$I_{FE} = S \cdot \frac{dP}{dE} \cdot \frac{dE}{dt} \quad \text{with} \quad E = \frac{V_{applied} - V_{off}}{t_{fe}} \quad (4)$$

These expressions and parameters together provide a robust framework for simulating the dynamic behaviour of ferroelectric memory devices. The permittivity of free space is given by ϵ_0 and ϵ_R is the relative permittivity of the ferroelectric material used. The values of other parameters used in the model, obtained from experimental analysis [3], are presented in Table I.

TABLE I
FITTING PARAMETERS VALUES AND FECAP DIMENSIONS

A (C/m ²)	w (V/10nm)	S (μm ²)	V_{off} (V)
0.36	0.54	306	0.32
A_{Leak}	t_{fe} (nm)	E_c (V/nm)	ϵ_R
0	10	0.179	29.7

IV. VERILOG-A CODE AND RESULTS

A. Verilog-A code used on ICCAP

The following Verilog-A code implements the ferroelectric capacitor model within the ICCAP simulation environment.

```
// Rate of change of Polarization (dP/dE)//
if (dV_dt > 0) begin
dP_dE = (2 * A_plus / `PI) * (w_plus)
/ (4 * (V(p, n) - V_off - E_c_plus)**2 + w_plus**2);
end else if (dV_dt < 0) begin
dP_dE = (2 * A_minus / `PI) * (w_minus)
/ (4 * (V(p, n) - V_off - E_c_minus)**2 + w_minus**2);
end
// Ferro current equation (I_fe)//
Ife = area * dE_dt * dP_dE;
I(p,n) <+ Ife;
// Polarization//
V(Pol) <+ idt(I(p,n)) / area;
```

B. Simulated Results

The simulation results of the FeCAP model in the Keysight ICCAP environment are presented below. Fig. 2 shows transient analysis under a ramp voltage, highlighting dynamic response of the ferroelectric capacitor. Fig. 3 presents the I-V characteristics, illustrating the relationship between applied voltage and current.

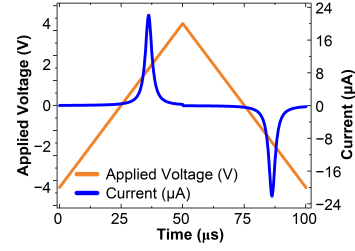


Fig. 2. Transient simulation of FeCAP using Verilog-A model in ICCAP

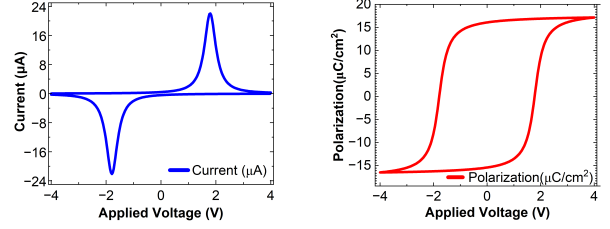


Fig. 3. I-V Characteristics of FeCAP Fig. 4. P-V Characteristics of FeCAP

Additionally, polarization is simulated by integrating the current within the model, and the results obtained are shown in Fig. 4.

CONCLUSION

This work presents a compact Verilog-A model for FeCAPs, readily usable with SPICE tools, and capable of accurately simulating ferroelectric polarization. The model supports compute-in-place architectures by ensuring reliable and efficient operation during frequent switching cycles. The validation is anticipated in the near future with experimental data.

ACKNOWLEDGMENT

This work has received state aid managed by the Agence Nationale de la Recherche (ANR) under France 2030, referring to ANR-23-PEEL-0003.

REFERENCES

- [1] J. P. Noel et al., "Compute-In-Place Serial FeRAM: Enhancing Performance, Efficiency, and Adaptability in Critical Embedded Systems," IEEE VLSI-SoC, 2023.
- [2] Chenming Hu et al., "Ferroelectric HfO₂ Memory Transistors With High-k Interfacial Layer and Write Endurance Exceeding 10¹⁰ Cycles," IEEE Electron Device Letters, 2021.
- [3] M. Bocquet et al., "Memory Window in Si:HfO₂ FeRAM Arrays: Performance Improvement and Extrapolation at Advanced Nodes," in Proc. IEEE Int. Memory Workshop (IMW), 2023.
- [4] Y. Zhou et al., "Mechanisms of imprint effect on ferroelectric thin films" J. Appl. Phys., 2005.
- [5] I. Fina et al., "Non-ferroelectric Contributions to the Hysteresis Cycles in Manganite Thin Films: A Comparative Study of Measurement Techniques," J. Appl. Phys., 2011.

Modélisation de la consommation énergétique d'une station dans un réseau Wi-Fi HaLow

Sébastien Maudet, Guillaume Andrieux, Jean-François Diouris
Université de Nantes, CNRS, IETR UMR 6164, F-85000 La Roche-sur-Yon, France
 sebastien.maudet@univ-nantes.fr

Abstract—Le déploiement d'un réseau IoT est soumis à des contraintes de consommation énergétique. Pour minimiser les coûts de service, il est essentiel d'optimiser la durée de vie des objets, qui sont souvent alimentés par des sources d'énergie peu fiables. Cette optimisation doit s'appuyer sur des modèles finement ajustés qui prennent en compte toutes les spécificités de la transmission d'un objet connecté. Dans cette étude, un modèle de consommation d'énergie est proposé pour le protocole 802.11ah. Ce dernier est basé sur des mesures réalisées in-situ et les résultats montrent l'influence du nombre de stations sur la consommation d'énergie d'une station.

Index Terms—Énergie; IoT; Wi-Fi HaLow; IEEE 802.11ah.

I. INTRODUCTION

L'efficacité énergétique est l'un des principaux défis de l'IoT. Dans ce type de réseau, le nombre d'appareils est important, ils sont géographiquement dispersés et l'accès à l'énergie n'est pas fiable. La norme Wi-Fi HaLow a été spécifiquement conçue pour répondre à ces besoins et elle hérite des principales caractéristiques du protocole Wi-Fi.

Au niveau de la couche physique, les transmissions sont réalisées dans des bandes de fréquence inférieures à 1 GHz, avec une modulation OFDM (*Orthogonal Frequency Division Modulation*) dont les caractéristiques sont dans un rapport de 10 avec la norme 802.11ac. La portée est estimée à 1 km et le débit entre 150 kbit/s et 78 Mbit/s. Une bande passante à 1 MHz a été ajoutée afin d'augmenter la portée de transmission et améliorer la consommation d'énergie [1], [2].

Au niveau de la couche MAC (*Media Access Control*), la norme 802.11ah réutilise les mêmes mécanismes d'accès au média que le Wi-Fi classique. Ces méthodes sont basées sur le CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*). La norme introduit également deux nouveaux mécanismes d'accès (RAW-*Restricted Access Window* et TWT-*Target Wake Time*) qui permettent à l'AP (*Access Point*) de réserver des canaux de transmissions temporels pour les stations en fonction de leurs besoins [1], [2]. En outre, la couche MAC intègre une nouvelle structure d'identification des stations (AID-*Association ID*) pour gérer jusqu'à 8 000 appareils, et des en-têtes raccourcis pour minimiser la taille des trames et donc l'empreinte énergétique [3].

Ces caractéristiques renforcent la nécessité d'une modélisation précise de la consommation d'énergie [4].

Ces travaux ont été soutenus par La Roche-sur-Yon Agglomération, la Région Pays-de-la-Loire et l'Union européenne à travers le Fonds européen de développement régional (FEDER) dans le cadre des plateformes WISE'Labs. Ces travaux ont été financés par l'Agence nationale de la recherche (ANR-22-PEFT-0007) dans le cadre de France 2030 et du projet NF-FITNESS.

II. MESURE DE LA CONSOMMATION D'ÉNERGIE

Le modèle de consommation énergétique est défini à partir de l'observation du fonctionnement d'une station 802.11ah, en utilisant un banc de test représenté sur la Fig. 1.

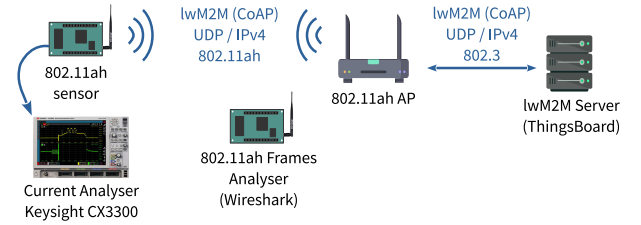


Fig. 1. Implémentation du banc de test du protocole 802.11ah.

La station est construite autour d'un SoC NRC-7292 qui contient deux processeurs Cortex M0 et M3, de la mémoire et toutes les interfaces nécessaires au développement d'une solution IoT [5]. Le point d'accès est un Raspberry Pi 3B+ équipé d'un *shield* APhi-7292 [6]. Le serveur est directement connecté au point d'accès et exécute une solution lwM2M.

Une autre carte Raspberry Pi 3B+ capture les échanges entre la station et l'AP à l'aide de *wireshark*. Le courant consommé par la station est mesuré à l'aide d'un analyseur de courant *Keysight CX-3300* [3]. Le modèle analytique est défini à partir de ces mesures de consommation d'énergie.

III. MODÈLE ANALYTIQUE DE CONSOMMATION D'ÉNERGIE

Le fonctionnement d'une station lors d'une phase de réveil est représenté sur la figure 2, en utilisant un outil générique basé sur des chaînes de Markov absorbantes [7].

La station se réveille, démarre son RTOS (*Real Time Operating System*), réalise des mesures physiques et transmet les données au serveur. Les opérations de transmission sont marquées d'un losange (◆). Si toutes les trames sont transmises avec succès (p_s), la chaîne de Markov se termine à l'état ● S. Si la transmission d'une seule trame échoue (p_{fa}), la chaîne de Markov se termine dans l'état ▲ F. Chaque opération de transmission est modélisée indépendamment par une chaîne de Markov absorbante. Cette chaîne décrit les différents échanges effectués par une station utilisant les couches de communication IP/UDP. Elle prend également en compte le nombre de retransmissions et le nombre d'appareils dans le réseau, en intégrant les stations exposées et cachées grâce à une distribution géographique.

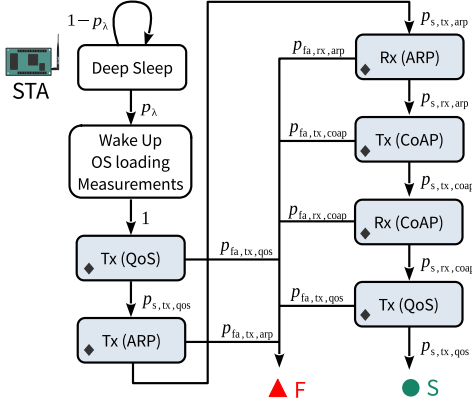


Fig. 2. Modélisation du fonctionnement d'une station 802.11ah lors d'une phase de réveil.

La station commence par envoyer une trame QoS *null data* pour annoncer son réveil à l'AP. Ils s'échangent ensuite des trames ARP (*Address Resolution Protocol*) pour mettre à jour et vérifier le mappage IP/MAC du réseau. Les données, au format CoAP/JSON sont ensuite envoyées par la station au serveur. Cet échange se termine par une trame QoS *null data* pour annoncer la mise en veille profonde de la station. Toutes les trames sont acquittées par une trame NDP-ACK.

L'énergie moyenne consommée par une station 802.11ah et sa probabilité de transmission avec succès sont ainsi déterminées à l'aide de ce modèle :

$$\begin{aligned} \bar{E}_{\text{tot}} = E_{\text{wu}} + N_{\text{tx,qos}} \cdot \bar{E}_{\text{tx,qos}} + N_{\text{tx,arp}} \cdot \bar{E}_{\text{tx,arp}} \\ + N_{\text{rx,arp}} \cdot \bar{E}_{\text{rx,arp}} + N_{\text{tx,coap}} \cdot \bar{E}_{\text{tx,coap}} \\ + N_{\text{rx,coap}} \cdot \bar{E}_{\text{rx,coap}}, \end{aligned} \quad (1)$$

$$p_s = \prod_{tr} p_{s,tx}^{N_{tx,tr}} \cdot \prod_{tr} p_{s,rx}^{N_{rx,tr}}, \quad (2)$$

où \bar{E}_{tot} est l'énergie moyenne consommée par une station pendant une phase de réveil et p_s est la probabilité de transmission avec succès. Le modèle est détaillé dans l'article [4].

IV. ANALYSE DES RÉSULTATS

La figure 3 montre l'évolution de la probabilité de transmission et l'énergie moyenne consommée par bit utile par une station 802.11ah en fonction du nombre de stations dans le réseau et pour différentes distances qui la sépare de l'AP. La probabilité de transmission est représentée sur l'axe de droite et l'énergie consommée sur l'axe de gauche.

Les limites sont principalement liées au nombre de stations présents dans le réseau. Lorsqu'il n'y a qu'une seule station, la probabilité de réussite est proche de 100%. La station effectue alors une seule tentative de transmission pour chaque trame. Lorsqu'il y a beaucoup de stations, la probabilité de succès est proche de 0%, et la station utilise toutes ces tentatives pour transmettre des données sans succès. De même, à mesure que la distance entre la station et l'AP augmente, la probabilité de transmission diminue, ce qui augmente la consommation d'énergie. Pour surmonter ce problème, la norme propose différentes solutions : le mécanisme de protection contre les

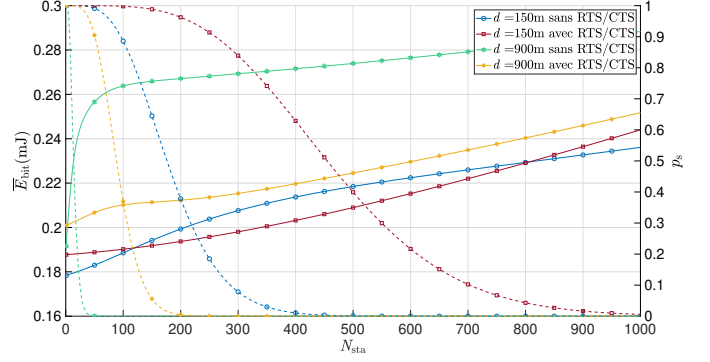


Fig. 3. Probabilité de transmission (ligne pointillée) et énergie moyenne consommée par bit utile (ligne continue) par une station 802.11ah ($N_{\text{data}} = 165$ bytes ; $P_{\text{ta}} = 23$ dBm).

collisions RTS/CTS et les mécanismes d'économie d'énergie RAW et TWT. Ces derniers permettent de regrouper les stations en fonction de leurs besoins et d'affiner les paramètres de transmission. Pour être optimale, la mise en grappe doit favoriser la répartition géographique et minimiser les stations cachés.

V. CONCLUSION

Dans cette étude, un modèle de consommation énergétique d'une station 802.11ah est présenté. Le processus de transmission des trames est modélisé individuellement par une chaîne de Markov absorbante. Les résultats montrent l'influence du nombre de nœuds dans un réseau 802.11ah sur la consommation d'énergie. Cette consommation dépend également de paramètres classiques tels que la puissance d'émission, la charge, le nombre de retransmissions, la distance et le cycle de service [4]. Dans tous les cas, l'énergie consommée par bit utile atteint une valeur limite qui dépend du nombre total de nœuds dans le réseau. Cette étude [4] propose des stratégies d'optimisation, telles que l'ajustement de la puissance d'émission et l'utilisation de RTS/CTS pour minimiser les collisions. Ces stratégies peuvent être appliquées immédiatement pour améliorer les performances du réseau et optimiser l'efficacité énergétique dans les déploiements IoT.

REFERENCES

- [1] L. Tian, S. Santi, A. Seferagić, J. Lan, and J. Famaey, "Wi-fi halow for the internet of things: An up-to-date survey on ieee 802.11ah research," *Journal of Network and Computer Applications*, vol. 182, p. 103036, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S108480452100062X>
- [2] N. Ahmed, D. De, F. A. Barbhuiya, and M. I. Hussain, "Mac protocols for ieee 802.11ah-based internet of things: A survey," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 916–938, 2022.
- [3] S. Maudet, G. Andrieux, R. Chevillon, and J.-F. Diouris, "Evaluation and analysis of the wi-fi halow energy consumption," *IEEE Internet of Things Journal*, pp. 1–1, 2024.
- [4] —, "Refined energy consumption model of an sta in a wi-fi halow network," *IEEE Transactions on Communications*, pp. 1–1, 2025.
- [5] Newracom. (2023) Nrc7292. [Online]. Available: <https://newracom.com/products/nrc7292>
- [6] Alfa_Network. (2023) Ahpi7292s. [Online]. Available: <https://www.alfa.com.tw/products/ahpi7292s>
- [7] F. Ait Aoudia, M. Gautier, M. Magno, O. Berder, and L. Benini, "A generic framework for modeling mac protocols in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 25, no. 3, pp. 1489–1500, 2017.

CMOS Design of a Bidirectional AC-Switch Control Circuit with Capacitive Isolation

Jasper Arbois, Francesco Guinta, Ming Zhang
C2N-CNRS
University of Paris-Saclay
Palaiseau, France
ming.zhang@universite-paris-saclay.fr

Nicolas Llaser
Computer Science
College LPO Dorian
Paris, France
nicolas.llaser@yahoo.com

Abstract—In this paper, CMOS design of an AC-switch control circuit is proposed with the following restrictions: immune to a floating ground having the same frequency as the AC-switch control signal (1MHz) but a relatively high magnitude and short signal delay ($<10\text{ns}$). The proposed capacitive-isolation circuit is designed in a CMOS HV $0.35\mu\text{m}$ technology. The simulation results have shown that the floating voltage immunity of the proposed circuit is up to $80V_{pp}$ (24 times V_{dd}) with a signal delay of 10ns.

Keywords—capacitive isolation, CMOS, signal transmission, Bidirectional AC switch, floating ground

I. INTRODUCTION

An electronic switch is one of the common electronic components used in electronic circuits. To implement a switch, CMOS technology is the best candidate compared to a Bipolar technology for its drain-source voltage V_{ds} can be down to 0V. For a DC switch, the simplest structure is using one MOS transistor (PMOS or NMOS), while for an AC switch, it becomes more sophisticated because the current flowing is bidirectional during each signal cycle. As a result, not only more CMOS transistors must be used but also a floating voltage node must be included within the switch. If the circuit must be CMOS-compatible, more restrictions should be respected by the switch control circuit design.

One of the applications of an AC switch is tuning the tank circuit of a receiver [1]. As the signal is sinusoidal, an AC switch must be used. As a tank circuit can also be used within a HIFU (High Intensity Focused Ultrasound) medical system to optimize the power transmission to ultrasound (US) transducers [2-4], an AC switch can also be used for the latter.

Auto-tuning tank circuit is one of the possible approaches to achieve more efficient power transmission to transducers, which motivates this work. To allow a bidirectional current flowing through an AC switch, a floating node within the switch is needed (Fig. 1), which complicates the switch control signal transmission. To overcome the impact of floating node on the control signal transmission (V_o must be referenced to the floating ground), an electrical isolation

system must be used.

Three kinds of electrical isolation can be distinguished from the literature: inductive isolation [5], capacitive isolation [6] and optical isolation [7]. Among them, capacitive isolation is the least used among the three isolation systems. But it offers the highest dV/dt speed leading to a rather short signal delay. Moreover, the capacitive isolation is completely compatible with CMOS technology. Even though a capacitive isolator has a limited barrier voltage, it can still reach up to 125V in the CMOS HV $0.35\mu\text{m}$ technology. Therefore, a capacitive isolation is the most suitable for our application and motivates this study.

II. PROPOSED CAPACITIVE ISOLATION SYSTEM

The proposed AC-switch driver circuit is shown in Fig. 1. It consists in a modulator and a demodulator separated by a capacitive isolator. Several key points are worth to mention, which clearly distinguish this work from existing work [6]; 1) to be compatible with CMOS technology, a CMOS-based circuit is proposed; 2) as this design aims at integrated circuit (IC) design, significant improvements have been made such as using grounded capacitor, no resistive charging, with chosen charging range; 3) to further improve the circuit performances in terms of circuit operation frequency, output signal dynamic range as well as isolator impact on carrier frequency and immunity against the floating voltage from the AC switch, important modifications on circuit structure are also made. Both modulator and demodulator are supplied by two independent power supplies V_{dd} and V_{cc} .

With a capacitive isolator, to increase the signal transmission efficiency, a modulation / demodulation system is used to carry the 1MHz control signal on a high-frequency carrier signal to decrease the impact of the isolator C_c .

For a digital signal transmission, an amplitude modulation (AM) is performed by the modulator (Fig. 2). By way of the capacitive isolator C_c , the AM signal is then sent to and demodulated by the demodulator back to 1MHz signal before being used for AC-switch control (Fig. 3).

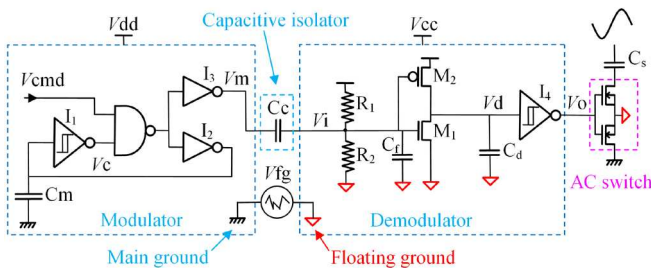


Fig. 1 Proposed CMOS capacitive isolation AC switch driving circuit

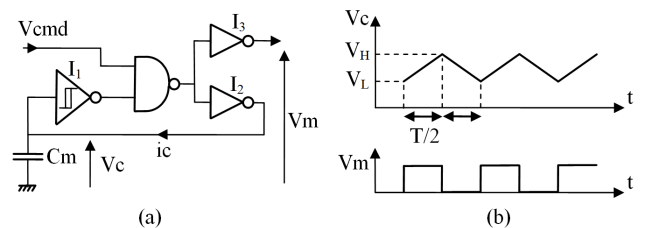
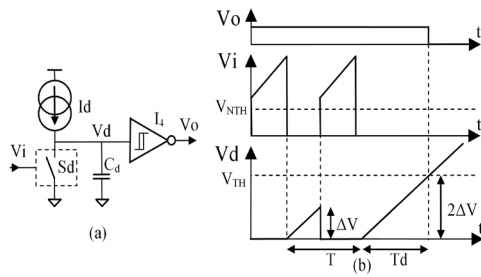


Fig. 2 Proposed modulator (a) and waveforms corresponding to charging / discharging the capacitor C_m ; V_c and the output voltage V_m for $V_{cmd}=0$ (b).

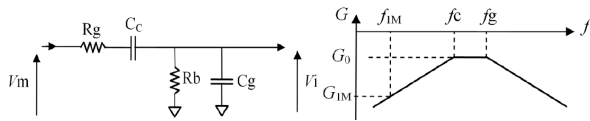


III. DESIGN CONSIDERATIONS

The main idea behind the circuit design is to separate the floating ground signal from the AC-switch control signal by choosing a high frequency carrier (100 MHz) with respect to low frequency floating voltage (1MHz). The capacitive isolator combined with the resistive biasing circuit forms a high pass filter (Fig. 4). The latter can bring -40dB attenuation to the floating ground signal if the cutoff frequency of the filter is chosen at 100MHz.

IV. SIMULATION

The proposed circuit was simulated in Cadence with CMOS HV 0.35 μ m technology with two independent voltage supplies of 3.3V.



A sinusoidal voltage source was used to bias the floating ground. Despite the presence of a floating voltage, the control signal could still be restored at the output of the demodulator confirming the proposed operation principle, as shown in Fig. 5. According to the simulations, the maximum floating voltage can be up to $80V_{pp}$ while the maximum breakdown voltage is of $\pm 125V$ for capacitors in this CMOS process, resulting in a floating ground immunity of $251V/\mu s$ and a signal delay of 10ns. A total power of 2.65mW was observed, among which 2/3 were consumed by the resistive biasing circuit. Further power reduction is still necessary and under investigation.

V. CONCLUSION

In this paper, a CMOS capacitive isolation AC-switch control signal transmission system has been proposed. The proposed circuit was designed in a CMOS HV 0.35 μ m technology with a voltage supply of 3.3V. The proposed circuit is compatible not only with CMOS technology but also with MRI environment. The observed signal delay was 10ns, i.e., only 1% of the signal cycle. Moreover, the simulation results showed floating ground immunity up to 251V/ μ s, i.e., 80V/pp, which should satisfy our application. Among the total power consumption of 2.65mW, two-third

was consumed by the resistive biasing circuit. Further efforts will be on the power reduction.

TABLE I. SUMMARY OF IC DESIGN OF THE PROPOSED CAPACITIVE ISOLATION SYSTEM

<i>Parameter</i>	<i>IC sim.</i>
Power supply	3.3V
Power consumption	2.65mW
Clock frequency	100MHz
Command frequency	1MHz
Signal delay	10ns
Maximum floating ground	80Vpp
Immunity	251V/ μ s

REFERENCES

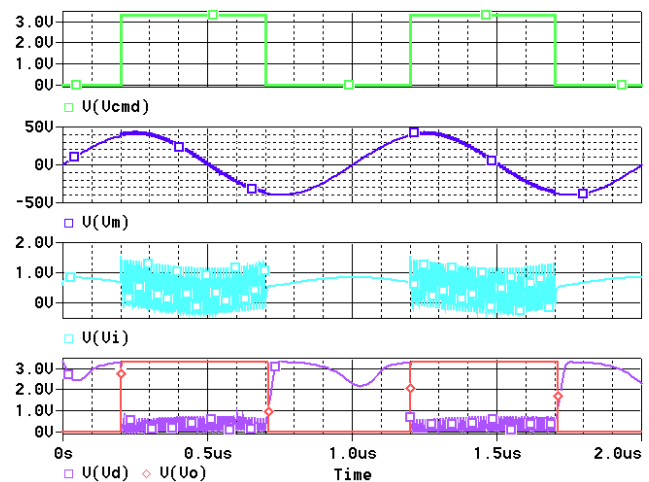


Fig. 5 Simulation results of the proposed capacitive isolation system with a 80V_{pp} floating ground.

- [1] P. Si, A. P. Hu, S. Malpas, D. Budgett, "A frequency control method for regulating wireless power to implantable devices," IEEE transactions on biomedical circuits and systems, Vol. 2, N° 1, March, pp. 22-29, 2008.
- [2] X. Wang, M. Zhang, N. Lasler, "CMOS 0.35 μm implementation of an auto-tuning system for a resonant converter," Analog Integrated Circuits and Signal Processing, oct., 2016.
- [3] Thomas M. Carpenter, "High-Power Gallium Nitride HIFU Transmitter With Integrated Real-Time Current and Voltage Measurement," IEEE Trans. Biomedical Cir & Sys, Vol. 15, No. 2, April, pp. 270-280, 2021.
- [4] Carlos Christoffersen, et al, "Quasi Class-DE Driving of HIFU Transducer Arrays," IEEE Trans. Biomedical Cir & Sys, Vol. 13, No. 1, Feb. , pp. 214-224, 2019.
- [5] A. Seidel, M. Costa, J. Joos, and B. Wicht, "Isolated 100% PWM gate driver with auxiliary energy and bidirectional FM/AM signal transmission via a single transformer," 2015 IEEE (APEC),USA, 2015, pp. 2581-2584.
- [6] Andrew L . Stone, et al, "Intrinsic safety isolation with capacitive coupling," US 9 , 990 , 837 B1, Jun. 5 , 2018.
- [7] Yan Zhang, et al, "Monolithic integration of broadband optical isolators for polarization-diverse silicon photonics", Optica Vol. 6, Issue 4, pp. 473-478, 2019.

Sécuriser la blockchain : détection de dérives temporelles en embarqué

Quentin JAYET

Univ. Grenoble Alpes
CEA, LETI, DSYS
F-38000, Grenoble, France
quentin.jayet@cea.fr

Christine HENNEBERT

Univ. Grenoble Alpes
CEA, LETI, DSYS
F-38000, Grenoble, France
christine.hennebert@cea.fr

Yann KIEFFER

Univ. Grenoble Alpes
Grenoble INP, LCIS
26000 Valence, France

yann.kieffer@lcis.grenoble-inp.fr

Vincent BEROULLE

Univ. Grenoble Alpes
Grenoble INP, LCIS
26000 Valence, France

vincent.beroulle@lcis.grenoble-inp.fr

Abstract—La blockchain crée un historique partagé et répliqué dans un réseau distribué. Son protocole de consensus repose sur une preuve garantissant le temps écoulé entre deux blocs. Dans Bitcoin, cette preuve est la *Proof of Work* qui est inadaptée aux systèmes embarqués en raison de sa forte consommation énergétique. *Proof of Hardware Time* propose une alternative à faible consommation reposant sur la mesure du temps écoulé au sein d'un *System on Module* sécurisé comprenant un microprocesseur ARM Cortex-A7 intégrant une *TrustZone* et un *Trusted Platform Module*. Mais cette mesure peut être altérée par des attaques. Ce papier introduit une méthode permettant de détecter des dérives temporelles induites par des attaques en température sur des composants de sécurité matérielle.

Index Terms—Détection, Dérive Temporelle, TEE, TPM, Attaque en température, Blockchain

I. INTRODUCTION

Les blockchains s'appuient sur des protocoles de consensus pour ordonner des événements au sein de réseaux distribués. Le protocole utilisé par Bitcoin décourage les comportements malveillants grâce au mécanisme de la *Proof of Work* qui garantit le temps écoulé entre deux enregistrements successifs. La *Proof of Work* garantit ce temps écoulé en saturant de calculs le matériel utilisé. Ainsi, elle engendre de la confiance entre les pairs du réseau, au détriment de la consommation d'énergie. C'est pourquoi la recherche s'intéresse à d'autres mécanismes de preuves visant à garantir le temps écoulé entre deux enregistrements. Nos travaux de recherche ont conduit à introduire la *Proof of Hardware Time* (PoHT) qui exploite l'horloge de composants de sécurité matérielle, et pourrait fournir de la confiance à très faible consommation [1]. Cependant, pour que cette approche soit sécurisée, la mesure du temps écoulé ne doit pas dériver par rapport au temps qui s'écoule. Pour tenir cette promesse, le matériel qui génère la preuve doit être d'un niveau de sécurité élevé. S'appuyant sur l'étude conduite dans [2], qui montre que les horloges des différents composants de sécurité matérielle dérivent différemment lorsqu'elles sont soumises à la même contrainte de température, ce papier introduit un procédé de détection des dérives d'horloge des composants de sécurité matérielle utilisés pour construire la PoHT au sein d'un dispositif em-

barqué intégrant un processeur ARM Cortex-A7 et un Trusted Platform Module (TPM).

II. ÉTAT DE L'ART

Les mesures du temps écoulé peuvent être altérées par des attaques [3]. Par conséquent, dans les preuves basées sur le temps écoulé [4], l'attestation de mesure peut être intégrée même si la mesure numérique du temps ne reflète pas le temps réel. Cette vulnérabilité rend possible les *long range attacks* [5]. Ces attaques consistent à réécrire l'historique d'une blockchain en générant une chaîne alternative plus longue à partir d'un bloc plus ancien. Cette attaque devient réalisable si un adversaire parvient à réduire la mesure numérique du temps entre deux blocs successifs.

III. PROOF OF HARDWARE TIME

La *Proof of Hardware Time* (PoHT) atteste du temps écoulé [1] sur un *System on Module* (SoM) intégrant une *TrustZone* et un *Trusted Platform Module* (TPM). Dans cette étude, le SoM utilisé est une carte d'évaluation STM32MP157F-DK2, équipée d'un processeur ARM Cortex-A7 intégrant une ARM *TrustZone* et d'un TPM 2.0 (STPM4RasPI). Le SoM possède différentes horloges pouvant être utilisées pour mesurer le temps : (1) les timers, (2) la *Real-Time Clock* (RTC) et (3) l'horloge du TPM. La Figure 1 illustre l'implémentation de la PoHT au sein du SoM. Après la réception d'un bloc, un nouveau bloc est construit dans le *Normal World*, puis transféré vers la *TrustZone* pour calculer la preuve. Un délai aléatoire, généré à l'aide d'un générateur de nombres aléatoires, détermine la durée d'attente. Deux attestations de temps sont émises par le TPM : une au début et une à la fin de cette période d'attente, afin de mesurer le temps écoulé. Cette mesure est utilisée comme preuve et insérée dans le bloc. Ensuite, le bloc est signé par la *TrustZone*, puis transmis aux pairs du réseau via le *Normal World*.

IV. SYSTÈME DE DÉTECTION BASÉ SUR LA MESURE DES DÉRIVES D'HORLOGE

Le procédé de détection proposé dans cette section exploite l'observation du comportements des dérives des horloges dans [2]. L'horloge RTC est choisie comme référence au sein du système embarqué. En cas de dérive, une interruption est

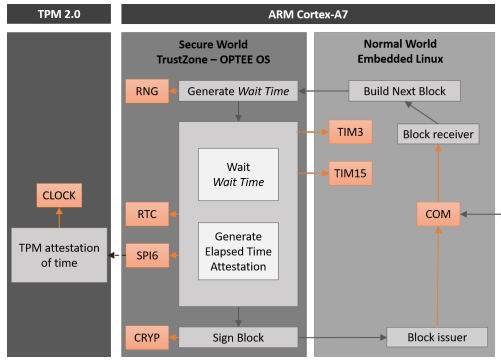


Fig. 1: Principales étapes de la PoHT réalisées au sein du SoM

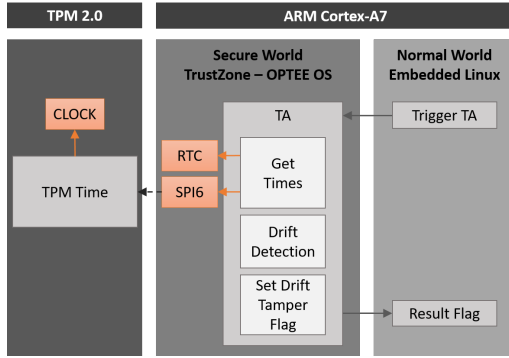


Fig. 2: Architecture embarquée du procédé de détection

levée. La décision étant binaire, il s'agit de décider du seuil à partir duquel l'interruption est générée.

La Figure 2 présente l'architecture embarquée de notre système de détection. Ce système est mis en œuvre dans une application sécurisée au sein de la TrustZone. L'exécution de cette application sécurisée est lancée à intervalles de temps réguliers par un mécanisme équivalent à un *watchdog* sécurisé. A chaque appel, les mesures de temps des différentes horloges sont collectées, en particulier, la mesure de la RTC et de l'horloge du TPM. Le seuil de déclenchement de l'interruption est défini comme variation de 0,5% de la fréquence d'horloge pour ne pas détecter les fluctuations liées aux temps d'accès aux mesures. Lorsqu'une interruption est levée, le SoM effectue une série d'actions pour limiter l'impact de la dérive détectée. Ces actions peuvent être, par exemple, l'arrêt du dispositif, l'arrêt de la génération d'attestations ou un processus de synchronisation des horloges avec le réseau.

La Figure 3 illustre l'évolution de la fréquence d'horloge du TPM lors des attaques réalisées dans [2]. Il est observé qu'une élévation de la température du SoM entraîne une diminution de la fréquence d'horloge du TPM.

Nous avons simulé une attaque thermique en ajustant manuellement la fréquence d'horloge du TPM à l'aide des commandes TPM [6]. Dans cette simulation, une réduction brutale de la fréquence d'horloge est appliquée afin de provoquer l'apparition du *flag* d'interruption. La Figure 4 présente les résultats de cette simulation, mettant en évidence

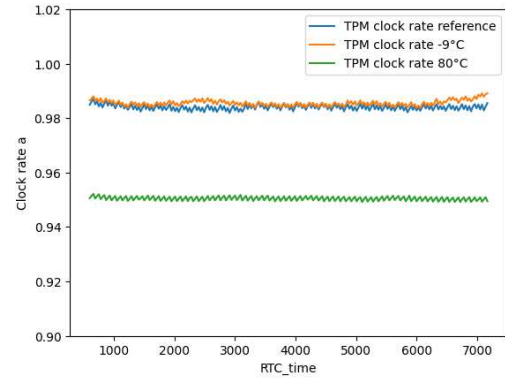


Fig. 3: Fréquence d'horloge du TPM pendant l'attaque

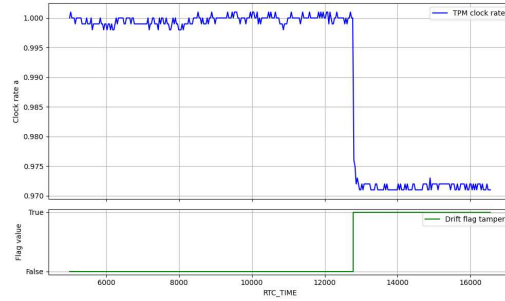


Fig. 4: Fréquence d'horloge du TPM pendant l'attaque et valeur du *flag* d'interruption

l'apparition du *flag* d'interruption lorsque la fréquence du TPM varie de 0,5%.

V. CONCLUSION

Ce papier introduit un procédé de détection des dérives du temps écoulé, mesuré par des composants de sécurité matérielle en embarqué. Une simulation d'attaque en température est réalisée pour illustrer le bon fonctionnement du procédé mis en œuvre. Dans le contexte des blockchains et de la PoHT, ce procédé de détection peut permettre de fournir une contre-mesure aux *long range attacks*.

REFERENCES

- [1] Q. Jayet, C. Hennebert, Y. Kieffer, and V. Beroulle, "Embedded elapsed time techniques in trusted execution environment for lightweight blockchain," in *2024 IEEE International Conference on Blockchain (Blockchain)*, 2024, pp. 81–88.
- [2] —, "Securing elapsed time for blockchain: Proof of hardware time and some of its physical threats," in *2024 27th Euromicro Conference on Digital System Design (DSD)*, 2024, pp. 137–144.
- [3] F. M. Anwar and M. Srivastava, "Applications and challenges in securing time," in *12th USENIX Workshop on Cyber Security Experimentation and Test (CSET 19)*, 2019.
- [4] M. Bowman, D. Das, A. Mandal, and H. Montgomery, "On elapsed time consensus protocols," in *Progress in Cryptology – INDOCRYPT 2021*, A. Adhikari, R. Küsters, and B. Preneel, Eds. Cham: Springer International Publishing, 2021, pp. 559–583.
- [5] E. Deirmentzoglou, G. Papakyriakopoulos, and C. Patsakis, "A survey on long-range attacks for proof of stake protocols," *IEEE Access*, vol. 7, pp. 28 712–28 725, 2019.
- [6] Trusted Computing Group, "Trusted platform module library," [Accessed 03-12-2024]. [Online]. Available: <https://trustedcomputinggroup.org/wp-content/uploads/TPM-Rev-2.0-Part-2-Structures-00.99.pdf>

Embedded Electronic System for Real-Time Wheel flat detection and monitoring

LAVIALE Mathias
LGIPM
University of Lorraine
F-57000 Metz, France

TANOUGAST Camel
LGIPM
Université de Lorraine
F-57000 Metz, France

BEN-AKKA Marouane
LGIPM
Université de Lorraine
F-57000 Metz, France

RAMENAH Harry
LGIPM
Université de Lorraine
F-57000 Metz, France

GORSE Jean
PLCD
F-57070 Saint-Julien-Lès Metz, France

ROSSIGNOL Stéphane
LORIA
CentraleSupélec
Metz, France

Abstract — This paper proposes a system for detecting mechanical stress in rails caused by moving trains, using signal processing for the interpretation of train wheel defaults. Based on previous research on train wheel defaults, this system identifies wave propagations associated with mechanical stress produced by the impact of the train's movement on the rails. Signal processing methods based as the Short Time Fourier Transform and Wavelet Transform are subsequently used on the onsite measured data to observe features of train wheel behavior and to detect defective wheels and/or rails.

Keywords— Wavelet, sensors, signal processing, train wheel default detection.

I. INTRODUCTION

Defective wheels and/or rails can accelerate railway track degradation, leading to excessive maintenance costs. Wheel flats appear as flat spots on the wheel's tread, mostly caused by emergency braking and slippery conditions on the rail, which reduce braking efficiency and deteriorates the tread [1]. Based on previous research, it has been proven that vibrations caused by a passing train (dynamic stress) are affected by the condition of the wheels and railway tracks [1-5]. The objective of this research is to develop an embedded system for the real-time measurement of vibrations associated with mechanical stress produced by the impact of train movements on rails, to detect wheel flats for predictive maintenance purposes using time-frequency processing [6].

II. EMBEDDED SYSTEM AND SENSORS

The technological choice for the proposed system is based on the need to capture the vibrations emitted in the rail by moving trains. The proposed solution is to apply sensors (accelerometers) on the rail's web (see Fig. 1), providing a reliable and safe location for the sensors. The proposed detection system is shown in Fig. 2. It is based on a Microchip PIC32 microcontroller development board and two accelerometers, ensuring real-time sampling acquisition. The sensors are commonly available piezoelectric accelerometers from Mikroelectronika, chosen for their wide configurable measuring range from $\pm 8g$ to $\pm 64g$ and their ability to be sensitive to both low and high frequencies components of the signal, ensuring accurate tracking of mechanical stress signatures related to potential wheel defects. The system is programmed to sample data at a speed of 4.4 kHz, following the literature [1] [4], where most of the energy in the spectrum is contained below 2 kHz. The embedded system is linked via a UART connection to a laptop running data acquisition software.

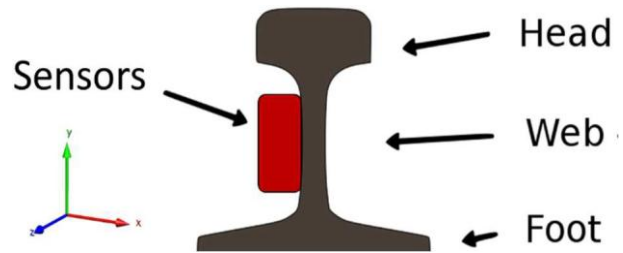


Fig. 1. Sensors on the rail cross-section.

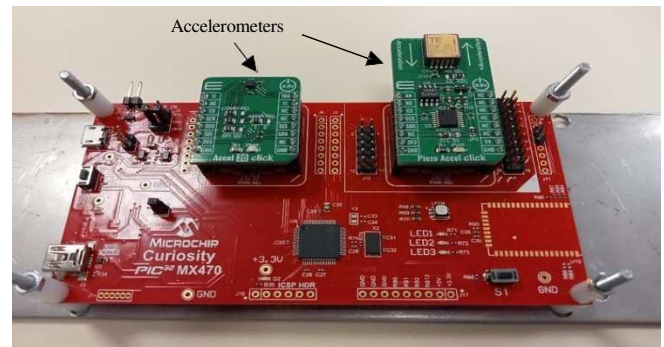


Fig. 2. System prototype with two accelerometers sensors.

III. DATA MEASUREMENTS AND PROCESSING

With the help of industrial partners, a measurement campaign was organized on four freight wagons with bogie-type wheels before their mandatory maintenance, and on two freight wagons freshly out of maintenance with brand-new bogie-type wheels noted as “healthy”. These onsite measurements serve as signal references for data processing and comparison to detect wheel defects. During the measurement campaign, the specific types of defects present and their severity prior to maintenance were unknown to us. The train speed during the tests ranged from 10 to 15 km/h. Fig. 3 shows the acceleration measurement values for a convoy of three wagons and one locomotive before maintenance. Subsequent expert inspection confirmed the presence of a wheel-flat among the tested wagons, thereby validating the dataset as representative of a known defect case. To characterize this fault, processing methods such as the Short Time Fourier Transform (STFT) and the Wavelet Transform (WT) were applied. The STFT is great for identifying initial spectral energy concentrations and frequency evolution over time. To refine the analysis, the Wavelet Transform (WT) was then applied, allowing for localized examination of non-stationary features in the signal. This multi-resolution capability is particularly suitable for detecting short-duration transients generated by wheel-rail impacts and assessing their relevance in the context of wheel defect detection [1][2].

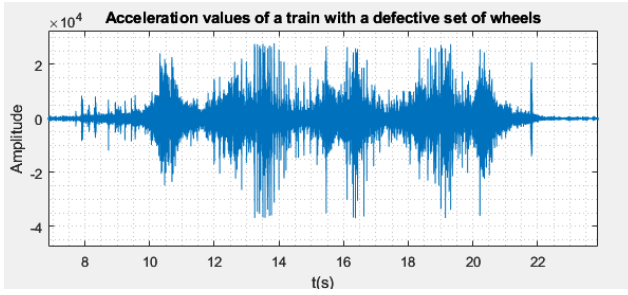


Fig. 3. Acceleration data from a three wagons convoy with a flat wheel.

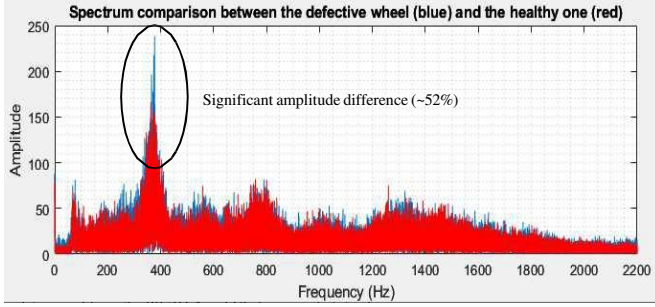


Fig. 4. FFT spectrum comparison between the defective wagon (green plot) and a healthy one (blue plot).

Initial spectral analysis using the Fast Fourier Transform (FFT), despite the non-stationary nature of the signals, helped identify frequency bands associated with wheel defects. Fig. 4 shows that in case of a common wheel defect (wheel-flat) the 300-400 Hz band is the most affected part of the spectrum where the neighboring band (500-1500 Hz) stays mostly unaffected.

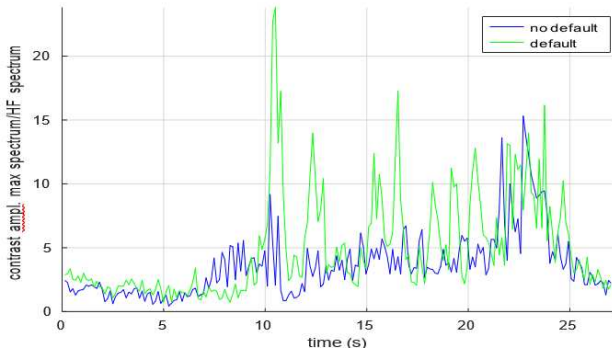


Fig. 5. STFT spectrum comparison between the defective train (green plot) and a healthy one (blue plot).

To refine this analysis and address signal quasi-stationarity, the Short-Time Fourier Transform (STFT) was applied using 1155-sample frames (~ 0.23 s) with a Hanning window. The two bands are refined to 305-418Hz (peak amplitude) and 616-1156Hz (mean energy). Fig. 5 compares both bands, showing consistent amplitude elevation in faulty cases, though without sufficient spatial resolution to localize the defect precisely. To improve localization, the Wavelet Transform (WT) was applied using the Morlet wavelet, selected for its excellent time resolution. This approach enables the detection of transient impact signatures and allows accurate identification of the affected bogie. Processing and comparing the “defective” and “healthy” peak amplitude spectrum (300-400 Hz) data highlights critical differences on one wagon. Fig. 6 shows two bogies (one centered at 2 seconds and the other at 4 seconds), the left one has a strong signature with the amplitude reaching green while

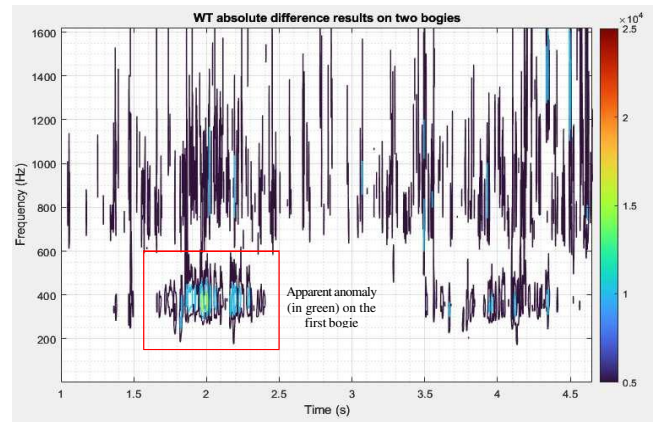


Fig. 6. WT absolute difference results between the defective bogie and the healthy bogies.

the right one remains largely deep blue, with this information it is possible to confirm the presence of a default and locate the bogie. Upon closer inspection maintenance experts confirmed that the highlighted bogie corresponded to the damaged one, validating the effectiveness of the proposed approach for defect detection using the developed embedded electronic and sensor system.

IV. CONCLUSION

The paper presents an embedded electronic system designed for real-time detection and monitoring of wheel flats on railway vehicles. The system captures structural responses from the rail induced by moving trains and applies signal processing techniques, including the Short-Time Fourier Transform (STFT) and the Wavelet Transform (WT), to identify wheel defects. The proposed approach demonstrates strong potential for deployment in predictive maintenance strategies, enabling early detection of wheel flats directly on site. Future work will explore additional directions such as barycenter analysis and new measurement campaigns to enrich the dataset and develop a structured database distinguishing defective and healthy wheels across various chassis configurations (bogies, standard suspensions, coil springs, etc.). To ensure real-time automated processing, the system will be ported to an FPGA platform and enhanced with AI-based methods.

REFERENCES

- [1] V. Belotti, F. Crenna, R. C. Michelini, G. B. Rossi, “Wheel-flat diagnostic tool via wavelet transform” *Mechanical Systems and Signal Processing*, vol. 20, 2006, pp.1953-1966.
- [2] B. Liang, S. Iwnicki, Y. Simon, D. Crosbee, “Railway wheel-flat and rail surface defect modelling and analysis by time-frequency techniques” *Vehicle System Dynamics*, vol. 51 (9), 2013, pp.1403-1421.
- [3] A. Mosleh, A. Meixedo, D. Ribeiro, P. Montenegro, R. Calçada, “Early wheel flat detection: an automatic data-driven wavelet-based approach for railways” *Vehicle System Dynamics*, vol. 61 (6), 2023, pp. 1644-1673.
- [4] A. Mosleh, P. A. Montenegro, P. A. Costa, R. Calçada, “Railway Vehicle Wheel Flat Detection with Multiple Records Using Spectral Kurtosis Analysis” *Applied Sciences*, vol. 11 (9), 2021, pp. 4002 (25)
- [5] A. Bracciali, G. Lionetti, M. Pieralli, “Effective wheel flats detection through a simple device” *World Congress on Railway WCR*, 1997.
- [6] J. W. Leis, *Digital signal processing using Matlab for students and researchers*, Hoboken, New Jersey : University of Southern Queensland, 2011.

Evaluation of Fixed-Point Quantized CNNs with Statistical Fault Injection

Wilfred Guillemé*, Angeliki Kritikakou[§], Youri Helen^{||}, Cédric Killian[‡] and Daniel Chillet*

*Univ. Rennes, Inria/IRISA, [§]IUF, ^{||}DGA MI, [‡]Univ. St-Etienne, Lab. Hubert Curien

Abstract—Convolutional Neural Networks (CNN), particularly those used in critical applications, such as autonomous driving, medical systems, and aerospace, require high reliability. While these algorithms exhibit inherent resilience, they remain susceptible to Single-Event Effects (SEE) occurring at the hardware and impacting the model execution. These effects, usually induced by interactions with radiation particles, can lead to errors in electronic components, potentially causing incorrect inferences and increasing the risk of mispredictions. To evaluate the fault sensitivity of fixed-point quantized CNN architectures, we propose SFI4NN, a Statistical Fault Injection (SFI) framework.

Index Terms—CNN, Fault Tolerance, Mitigation, SEU, TMR.

I. INTRODUCTION

Convolutional Neural Networks (CNN) are now deployed in many fields, including autonomous vehicle driving, medical systems, and aerospace. The reliability of these algorithms is crucial in these critical domains, even though they are inherently resilient. Single-Event Effects (SEE) can occur due to interactions with radiation particles, leading to errors in electronic components during inference. When a particle strikes a memory element, such as a D flip-flop, its value may change leading to Single-Event Upset (SEU). In the presence of such faults, AI algorithms can produce erroneous inferences, increasing the risk of incorrect predictions. To assess the resilience of CNNs against SEUs, Statistical Fault Injection (SFI) is typically used, as exhaustive fault injection is infeasible. However, no existing framework allows to perform resilient assessment for fixed-point quantized CNN. To address this limitation, this work presents a SFI framework (SFI4NN) that allows resilience analysis of quantized CNN models, for faults occurring not only on the network parameters, but also in the intermediate data. Our framework enables the simulation of bit-flips, taking into account their direction from 0 to 1 or from 1 to 0, their location within the CNN, including the layer and specific bit, to better understand their impact on the model's prediction.

II. RELATED WORK

Existing studies mainly evaluate the resilience with respect to faults occurring in model parameters [1]. However, intermediate data, temporarily stored during inference, is also susceptible to faults. In [2], a SFI method is proposed to limit the number of faults tested per layer and bit, leading to efficient evaluation of CNN robustness while significantly reducing simulation time. However, this study focuses only on floating-point representations and is not usable for embedded systems that generally implement fixed-point computation processing units. Some works explore the vulnerabilities of different data

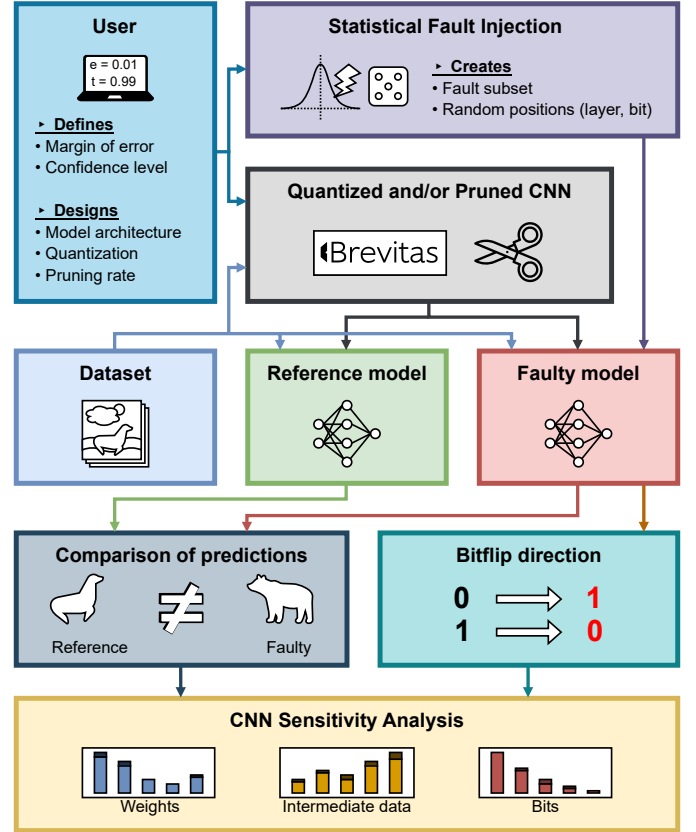


Fig. 1: SFI4NN's framework overview.

representations. In [3], fixed-point and floating-point formats are compared. In fixed-point representation, the most critical bits are the ones with higher significance, while in floating-point representation, the exponent bits are the most sensitive. Furthermore, several studies have shown asymmetries in the resilience of models related to the direction of bit-flips [4]. However, further investigation is required to characterize their impact on CNN classification.

III. METHODOLOGY

Exhaustive fault injection becomes computationally infeasible for complex models. To circumvent this problem, SFI [5] employs sampling techniques to estimate the impact of faults, while significantly reducing the number of fault injections. This approach, described by the following equation (1), allows for evaluating the robustness of the CNN against SEUs by defining an error margin and confidence level before the fault injection campaign.

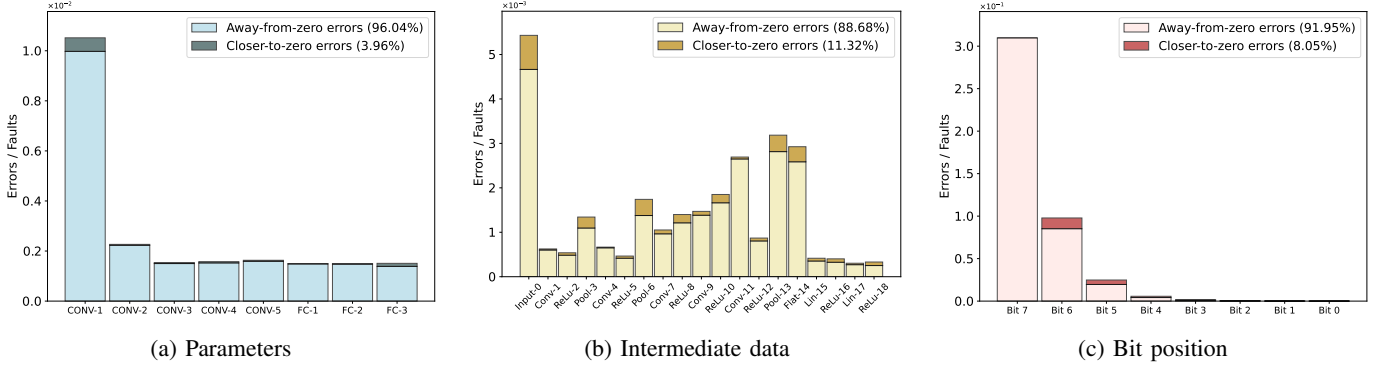


Fig. 2: Quantized AlexNet sensitivity results based on SFI4NN ($e = 10\%$, $t = 90\%$).

$$n(i, l) = \frac{N(i, l)}{1 + e^2 \cdot \frac{N(i, l) - 1}{t^2 \cdot p(i) \cdot (1 - p(i))}} \quad (1)$$

$N(i, l)$ depends on the CNN architecture and represents the total number of faults that can be injected into the network, based on the bit position i and layer l . The parameters e and t correspond to the error margin and confidence level defined by the user, respectively. Finally, p denotes the probability of an error occurring at each bit position. It is important to note that the number of injected faults increases for higher-order bits, as they have a greater impact on the network's output and are therefore more likely to cause prediction errors. These probabilities, which determine the fault injection rates based on the layers and bit positions, are computed using equation (2).

$$\forall i \in I \quad p(i) = p_{\min} + \frac{(D(i) - D_{\min})(p_{\max} - p_{\min})}{(D_{\max} - D_{\min})} \quad (2)$$

$D(i)$ represents the distance of a bit-flip on a value based on the selected bit. For a bit-flip on the least significant bit of a signed integer, D_{\min} is always equal to 1. In contrast, D_{\max} depends on the data length I . p_{\max} is automatically set to 0.5, while p_{\min} also depends on the data length, defined by the relation $p_{\min} = \frac{1}{2^I}$.

IV. EXPERIMENTS

The CNN architecture used in this study is based on AlexNet [6], which consists of five convolutional layers followed by three linear layers. The final classification is determined by the neuron with the highest output value. The AlexNet architecture has about 28.5 million parameters and processes 289,994 intermediate data points per CNN inference. The dataset used in this study is CIFAR-10 [7], consisting of 60,000 color images divided into 10 distinct classes, with each having a resolution of 32×32 pixels. The training set contains 50,000 samples, while the test set includes 10,000. Since the CNN is designed for execution on an embedded system, the model is quantized using an 8-bit fixed-point format with the Brevitas [8] framework. This method employs a Scale Factor (SF) to map weights to their corresponding signed integer representation, ranging from -128 to $+127$.

The fault injection campaign follows the SFI4NN approach, with a 10% error margin and a 90% confidence level. The results are presented in Figure 2. In each subfigure, the X-axis represents either the layer index or the bit position, while the Y-axis indicates the number of errors that resulted in a prediction mismatch between the faulty and the reference model, normalized by the total number of injected faults. Figure 2a illustrates the sensitivity of the layers containing the CNN parameters in the AlexNet architecture. As shown, the first convolutional layer is the most sensitive, with approximately a 1% chance of causing a misprediction. SFI results for intermediate data are presented in Figure 2b. A large variation in sensitivity can be observed between layers. By combining the results from the parameter and intermediate data analyses, we can assess the bit-level sensitivity across the entire CNN, as shown in Figure 2c. As expected, most significant bits exhibit the highest sensitivity on model behavior.

V. CONCLUSION

This paper presents a fault injection method named SFI4NN based on a statistical approach, designed to assess the resilience of fixed-point quantized CNNs. Validated on an AlexNet trained on CIFAR-10, this method helps identify critical elements to protect and enables the evaluation of lighter, cost-effective hardware protection schemes beyond traditional triplication.

REFERENCES

- [1] A. Bosio, P. Bernardi, A. Ruospo, and E. Sanchez, "A reliability analysis of a deep neural network," 2019.
- [2] A. Ruospo, G. Gavarini, C. De Sio, J. Guerrero, L. Sterpone, S. Reorda, E. Sanchez, R. Mariani, J. Aribido, and J. Athavale, "Assessing convolutional neural networks reliability through statistical fault injections," 2023.
- [3] G. Li, S. Hari, M. Sullivan, T. Tsai, K. Pattabiraman, J. Emer, and S. Keckler, "Understanding error propagation in deep learning neural network (dnn) accelerators and applications," 2017.
- [4] Y. Zhang, H. Itsuji, T. Uezono, T. Toba, and M. Hashimoto, "Estimating vulnerability of all model parameters in dnn with a small number of fault injections," 2022.
- [5] R. Leveugle, A. Calvez, P. Maistri, and P. Vanhauwaert, "Statistical fault injection: Quantified error and confidence," 2009.
- [6] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," 2012.
- [7] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.
- [8] A. P., "Xilinx/brevitas," 2023. [Online]. Available: <https://doi.org/10.5281/zenodo.3333552>

The technological choice in the eco-design of AI accelerators: landscape and research directions

Louis Maillard^{1,2}, Chiara Sandionigi¹, Hana Krichene³, David Bol⁴, Maxime Pelcat²

¹Université Grenoble Alpes, CEA, List, F-38000 Grenoble, France

²Université Rennes, INSA Rennes, CNRS, IETR - UMR 6164, F-35000 Rennes, France

³Université Paris-Saclay, CEA, List, F-91120 Palaiseau, France

⁴ICTEAM Institute, UCLouvain, Louvain-La-Neuve, Belgium

Abstract—As artificial intelligence (AI) technologies are employed in more and more sectors and activities, their environmental footprint has emerged as a significant concern. Among these technologies, AI accelerators implemented in integrated circuits (IC) are expected to grow in production volume. The selection of the IC implementation type (e.g., GPU, ASIC, FPGA, MCU) for the accelerators has direct effects on the environmental costs for manufacturing and use, the lifetime of the circuit and the possibility of more life cycles in a perspective of circular economy. This work explores the environmental implications in the broad domains of AI and ICs, and sets the stage for future research on the IC implementation technology selection for a sustainable development of AI accelerators.

INTRODUCTION

Among AI technologies, hardware acceleration of AI algorithm is expected to see its market increase in the following years. As an example, Canalys evaluates that smartphones equipped with dedicated AI hardware will account for 16% shipment share in 2024 and 54% by 2028 [1]. AI accelerators are dedicated hardware, such as ASICs and FPGAs, capable of efficiently executing AI models and supporting real-time on-device inference with optimized latency. However, the increasing deployment of these technologies raises critical environmental concerns due to resource consumption, greenhouse gas emissions, and electronic waste. The choice of the IC type for implementing such accelerators can affect their environmental impacts all along the life cycle, from the manufacturing phase to the end-of-life, going through the possibility of a second life, making its selection a crucial step for the sustainable development of AI accelerators. The objective of our research is to develop a methodology for systematically integrating technology selection into the eco-design workflow, aiming to identify the parameters that can guide this choice from the design phase, also anticipating the perspectives for a circular economy of the circuit. In order to explore how the technology choice affects the environmental impact of these systems, we surveyed the following research domains :

- Embedded AI, to outline the challenges and opportunities within this field;
- Operational footprint of AI;
- Embodied footprint of ICs.

This paper gives an overview on the state of the art of the three domains, and outlines the future research directions on the technology selection for AI accelerators.

I. EMBEDDED AI

Embedded AI is emerging as a transformative paradigm, spanning multiple industries. In the automotive sector, manufacturers have introduced hardware systems tailored to AI-driven functionalities [14]. The Internet of Things benefits from embedded AI in sensors deployed across industrial, agricultural, and home automation applications [13]. In healthcare, AI-embedded devices contribute to diagnostics, monitoring, and personalized treatments [9].

Embedded AI is offering significant advantages while posing notable challenges. Among the key opportunities, leveraging hardware acceleration for embedded AI tasks enables low-power computing compared to executing those same tasks purely in software, which can contribute to improved battery life and reduced energy consumption [11]. Additionally, the capacity for local computing enhances the autonomy of systems, reducing dependency on centralized servers. Another benefit that embedded intelligence fosters is low latency, a crucial factor in real-time applications [12]. The confidentiality of data is reinforced, as local processing minimizes exposure to external threats [9]. Embedded AI systems can be distributed across various nodes, reducing network congestion [13]. By executing smaller AI models, embedded AI promotes computational sobriety, an essential principle for sustainable development of AI [5]. Lastly, some of the learning layers of AI model could be implemented in embedded devices, therefore reducing the load on warehouse scale computing systems [10].

Despite these advantages, embedded AI faces significant limitations. The primary challenge lies in the trade-off between the constrained hardware resources, particularly memory and power consumption, and the rapid evolution and obsolescence of AI models. These increasingly complex models demand the processing of ever-larger volumes of data and require substantial computational power, making their direct implementation on resource-limited embedded systems difficult. While model training is often performed on high-performance servers, the deployment and efficient execution of these advanced models on embedded devices with limited capabilities remains a key obstacle. Furthermore, the diversity of architectural standards complicates the optimization and compatibility of AI applications across various embedded platforms [9].

II. OPERATIONAL FOOTPRINT OF AI

AI is experiencing unprecedented growth and shows no signs of slowing [2]. The direction of this growth is that AI technologies become more integrated into various sectors. Moreover, AI models become increasingly complex, and therefore, their energy consumption escalates, contributing to a significant carbon footprint. However, accurately assessing this impact remains a challenge due to the intricate nature of AI services and the lack of transparency from major AI firms [3]. The absence of systematic reporting mechanisms makes it difficult to obtain precise data on energy consumption and emissions, yet existing studies indicate a clear trajectory: AI's environmental impact is on the rise [4], [5]. Moreover, current evaluations likely underestimate this impact, as studies often focus only on energy consumption [5].

A key contributor to AI carbon footprint is the process of training AI models [6]. Training requires extensive computational power, often relying on large-scale data centers that consume significant amounts of electricity. However, more recent research provides a nuanced perspective, suggesting that inference emissions are also a major contributor in the carbon footprint of AI. Carbon emissions from inference could represent 1.85 to 3.33 times the carbon emissions generated by training the model [4], [7]. While training remains a significant factor, this suggests that the long-term energy consumption associated with inference should not be overlooked.

Beyond direct energy consumption, the systemic and behavioral effects of AI, that could increase AI's carbon footprint, remain insufficiently studied [8]. Similarly, AI-driven improvements in efficiency across industries could lead to rebound effects.

In most studies, only AI services based on models running on data centers and powerful GPU are studied, but other types of AI systems, like embedded AI, should be considered.

III. EMBODIED FOOTPRINT OF ICs

ICs significantly contribute to the carbon footprint of computing systems [15], particularly in AI systems [5]. The manufacturing process is energy-intensive and generates considerable greenhouse gas emissions [16]. However, their environmental impact is multifaceted, extending beyond carbon emissions to include abiotic resource depletion and substantial water usage during manufacturing [15], [16]. Adding environmental scores from the established Power, Performance, Area, and Cost metrics can contribute to the sustainable development of ICs [17]. Finally, the disposal of the equipment containing these components leads to a large volume of electronic waste, presenting additional sustainability challenges. In 2019, a staggering 53.6 million metric tons of e-waste were generated globally, with only 17.4% being officially collected and recycled [18]. The majority of e-waste, including integrated circuits, is improperly disposed of, often ending up in landfills or through informal recycling methods, particularly in low- and middle-income countries [18]. These practices can lead to the release of toxic substances such as lead, mercury, and cadmium into the soil, water, and air, causing

severe health consequences for exposed populations, especially children [19].

RESEARCH DIRECTIONS

This research will bridge the domains of sustainable AI, embedded AI, and the sustainability of integrated circuits, aiming to develop a comprehensive methodology that integrates environmental considerations into the early stages of AI accelerator design. We want to explore how the choice of technology for implementing an AI accelerator affects its environmental impact throughout its entire life cycle. Specifically, this research will delve into the comparative analysis of various technological options, such as ASICs and FPGAs, assessing their respective environmental impacts throughout the manufacturing, operational, and end-of-life phases. Our aim is to identify the parameters that can guide this choice from the design phase, while also anticipating the perspectives for a circular economy of the circuit.

REFERENCES

- [1] Canalys, "Now and next for AI-capable smartphones", May 2024
- [2] Our World in Data, 'Annual Private Investment in Artificial Intelligence', April 2025
- [3] J. Dodge et al, 'Measuring the Carbon Intensity of AI in Cloud Instances', arXiv, June 2022
- [4] CJ. Wu et al, 'Sustainable AI: Environmental Implications, Challenges and Opportunities', arXiv, January 2022
- [5] AL. Ligozat et al, 'Unraveling the Hidden Environmental Impacts of AI Solutions for Environment Life Cycle Assessment of AI Solutions', Sustainability 14, no. 9, January 2022
- [6] E. Strubell et al, 'Energy and Policy Considerations for Deep Learning in NLP', arXiv, June 2019
- [7] A. Berthelot et al, 'Towards a Multi-Criteria Evaluation of the Environmental Footprint of Generative AI Services', 2024
- [8] L. Kaack et al, 'Aligning Artificial Intelligence with Climate Change Mitigation', Nature Climate Change 12, no. 6, June 2022
- [9] Z. Zhang et al, 'A Review of Artificial Intelligence in Embedded Systems', Micromachines 14, no. 5, May 2023
- [10] H. Li et al, 'Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing', IEEE Network 32, no. 1, January 2018
- [11] D. Tchuinkou Kwadjo et al, 'Towards a Component-Based Acceleration of Convolutional Neural Networks on FPGAs', Journal of Parallel and Distributed Computing 167, September 2022
- [12] V. Mazzia et al, 'Real-Time Apple Detection System Using Embedded Systems With Hardware Accelerators: An Edge AI Application', IEEE Access 8, 2020
- [13] M. Merenda et al, 'Edge Machine Learning for AI-Enabled IoT Devices: A Review', Sensors 20, no. 9, January 2020
- [14] R. Islayem et al, 'Hardware Accelerators for Autonomous Cars: A Review', arXiv, April 2024
- [15] U. Gupta et al, 'Chasing Carbon: The Elusive Environmental Footprint of Computing', IEEE International Symposium on High-Performance Computer Architecture (HPCA), 2021
- [16] Q. Wang et al, 'Environmental Data and Facts in the Semiconductor Manufacturing Industry: An Unexpected High Water and Energy Consumption Situation', Water Cycle 4, 2023
- [17] M. Garcia Bardon et al, 'DTCO Including Sustainability: Power-Performance-Area-Cost-Environmental Score (PPACE) Analysis for Logic Technologies', IEEE International Electron Devices Meeting (IEDM), 2020
- [18] V. Forti et al, 'The Global E-waste Monitor 2020: Quantities, flows and the circular economy potential', 2020
- [19] SM. Parvez et al, 'Health consequences of exposure to e-waste: an updated systematic review', The Lancet Planetary Health, Volume 5, 2021

High Temperature Stress and NBTI Reliability Investigation of JL-VNWFETs

Y. Wang¹, C. Mukherjee¹, F. Marc¹, M. Deng¹, J. Müller², S. Pelloquin², G. Larrieu², and C. Maneux¹

¹ IMS Laboratory, University of Bordeaux, UMR CNRS 5218, Cours de la libération, 33405 Talence, France

²LAAS CNRS, University of Toulouse, UPR CNRS 8001, Av. du Colonel Roche, 31400 Toulouse, France

Abstract— This work presents results from accelerated aging tests to investigate the reliability of Junctionless Vertical Si Nanowire FET (JL-VNWFET) and to analyze the underlying degradation mechanisms. From long-term Temperature Storage (TS) and Negative Bias Temperature Instability (NBTI) tests, a strong degradation of device threshold voltage is observed that leads to a leakage current increase with stress time. A preliminary model of the degradation mechanism is derived based on the V_{TH} -shift extracted during stress and relaxation phases.

Keywords—JL-VNWFET, reliability, temperature storage test, NBTI, V_{TH} shift, degradation mechanism

I. INTRODUCTION

Emerging 3D transistor technologies, such as the junctionless vertical nanowire transistors (VNWFETs), are promising candidates for non-conventional computing architectures for their high compactness, low latency, and low power consumption, especially for designing neural networks (NNs). Its 3D vertical structure allows to increase the number of transistors per unit area through vertical stacking. The transistor fabrication process is also simplified, compared to conventional technologies, due to uniform high doping in the junctionless channel, while the gate-all-around (GAA) structure provides better immunity to short channel effects (SCE) and improved gate control [1]. However, the highly scaled architecture as well as the ultra-thin oxide layer lead to prominent electrothermal and trapping effects that in turn impact the electrical performance of the device and hence the performance of the logic circuits constructed based on this technology [2] [3]. For long-term operation, reliability is therefore a major challenge for this emerging technology. In order to investigate the underlying failure mechanisms, we therefore performed thermal and electrical stress tests, on the JL-VNWFET in order to understand and model the degradation physics.

II. TEMPERATURE STORAGE TEST

A. Thermal stress setup

The VNWFET devices were subjected to long-term thermal stress through temperature storage (TS) tests using the standard Measure-Stress-Measure (MSM) technique [4]. For this, we essentially subjected the JL-VNWFETs to uniform high-temperatures (T_{stress} of 100°C, 150°C or 200°C) without any applied voltage bias for a period of stress time (t_h). Following thermal stress, the samples were measured at room temperature under DC operating conditions to study the drift of device parameters. The stress-measurement cycles were repeated with increasing intervals of t_h , that followed a logarithmic distribution, until the degradation was stabilized. The devices under test included transistors with nanowire diameters ranging between 17 nm and 34 nm, with 64 or 81 nanowires in parallel. The I_D - V_G curves showed a similar degradation trend for all devices. As a representative example, the I_D - V_G characteristics of a VNWFET with a 17

nm diameter and 64 nanowires in parallel (Fig. 1) were chosen for the analysis. Two key observations include: (1) a shift in the threshold voltage (V_{TH}) and (2) an initial increase in the on current, followed by a decrease at higher stress temperatures at high gate and drain bias conditions.

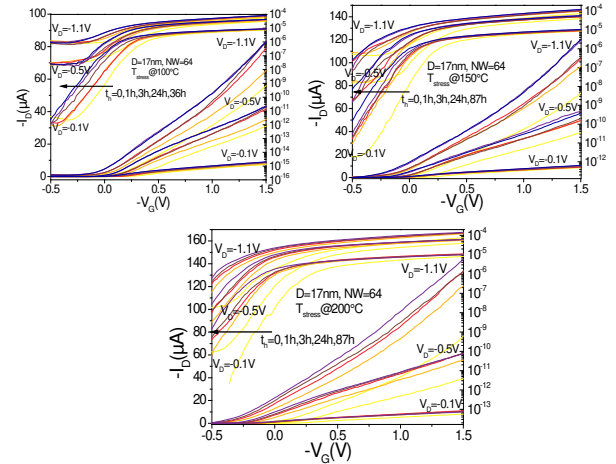


Figure 1: Evolution of I_D - V_G characteristics of a JL-VNWFET (17nm diameter and 64 nanowires in parallel) with stress time at (a) 100°C, (b) 150°C, and (c) 200°C.

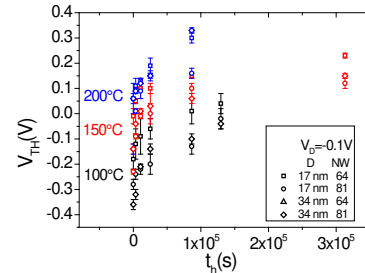


Figure 2: Extracted V_{TH} of the JL-VNWFETs under test as a function of the stress time for a $V_D = -0.1V$ at (a) 100°C, (b) 150°C, and (c) 200°C.

B. Analysis of temperature-induced degradation

In [2] and [3], we reported that charge trapping at Si-SiO₂ interface and oxide layer leads to the shift of V_{TH} and a linear increase of drain current can be observed with temperature. This behavior was attributed to a strong temperature dependence of the threshold voltage and an almost negligible variation in mobility with temperature [5]. The thermal stress tests indicate a similar phenomenon which likely provokes trap-induced V_{TH} -shift. However, the nature of the degradation (permanent or recoverable) depends on whether there is capture/release of carriers into/from pre-existing traps/defects in the Si/SiO₂ layers (recoverable, as is the case for DC temperature measurements) or defect generation (quasi-permanent). Several studies illustrate that temperature increases the activation energy [6], which in turn alters the interface state causing a shift in the threshold voltage.

III. NEGATIVE BIAS TEMPERATURE INSTABILITY

A. NBTI stress setup

Negative Bias Temperature Instability (NBTI) is a critical concern in highly scaled emerging FET technologies. In JL-VNWFET devices, oxide thickness scaling leads to an increased sensitivity to NBTI. We initially analysed the thermal degradation of the devices under temperature stress and since the general degradation due to NBTI showed a similar behaviour, we performed NBTI stress measurements at room temperature to understand the degradation due to bias stress. The HP4155 semiconductor parameter analyser is used for its ability to maintain a stable, constant voltage over long stress periods. Using GPIB programs written in Parameter Extraction Language (PEL), developed in-house, we automated the stress-measure-stress sequence to minimize possible human errors. To avoid the formation of hot carrier by the applied high electric field and Therefore a weak drain voltage ($V_D = -40\text{mV}$) was maintained during the NBTI stress, the same drain bias used for DC I-V measurements for V_{TH} extraction. Three different gate stress biases (-0.8V , -1V , -1.2V) were applied on the available VNWFET geometries (Fig. 3). The goal of this systematic study was to derive geometry and stress bias dependences of degradation model parameters.

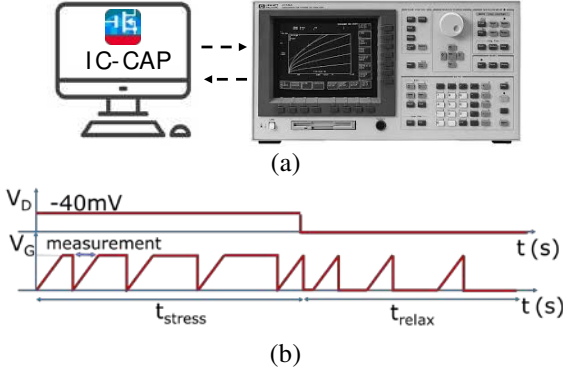


Figure 3: (a) Setup used for NBTI stress, (b) V_D and V_G stress bias waveforms for NBTI stress/relax phases.

B. Analysis of NBTI stress tests

The NBTI stress tests results showed similar results as the thermal stress tests. Fig. 4 presents the evolutions of I_D - V_G characteristics of a representative device, with 17nm diameter and 81 nanowires in parallel, stressed under a gate voltage of -1.1V . As can be observed from Fig. 4 (a), the V_{TH} shift of the VNWFET reaches a maximum value around 36 h and the characteristics do not evolve further even if the stress continued. During the relaxation phase, the device characteristics showed an almost complete recovery within the first half an hour and recovered to its initial state within 3 hours of relaxation (Fig. 4 (b)). The characteristics recorded 36 hours after the recovery started showed no further evolution. The threshold voltage of the JL-VNWFET was extracted from the I_D - V_G plots at different stress times (Fig. 5) which follows a classical stretched exponential model describing hole trapping in pre-existing oxide/interface traps during the NBTI stress. The threshold voltage shift can be empirically modelled by the following equations during the stress and the recovery phase, respectively [7],

$$\Delta V_{TH, stress} = \Delta V_{THmax} \left(1 - \exp \left(- \frac{t_{stress}}{\tau_{stress}} \right)^{\beta_1} \right)$$

$$\Delta V_{TH, relax} = \Delta V_{THmax} \exp \left(- \frac{t_{relax}}{\tau_{relax}} \right)^{\beta_2}$$

Parameters $\beta_1, \beta_2, \tau_{stress}$ and τ_{relax} are fitting parameters and have the values 0.42, 90s and 0.4 and 312 s, respectively. ΔV_{THmax} is the maximum V_{TH} shift observed during NBTI stress. All devices showed a similar trend and a full recovery after 36 hours. The rapid and complete recovery of the device characteristics not only infers a hole trapping dominated degradation, but also a weak interface trap/defect generation mechanism which is normally a relatively slow but quasi-permanent process and would have otherwise prevented a full recovery during the relaxation.

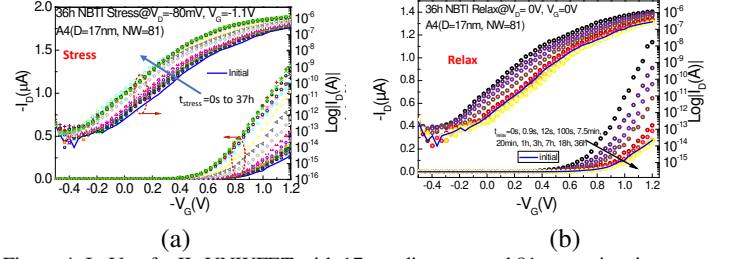


Figure 4: I_D - V_G of a JL-VNWFET with 17 nm diameter and 81 nanowires in parallel showing the evolution of the characteristics for (a) 36h of stress and (b) 36h of relaxation.

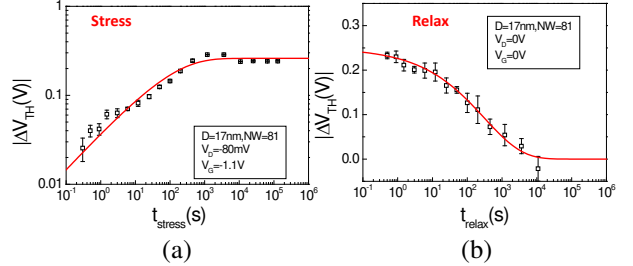


Figure 5: Threshold voltage shift of a JL-VNWFET with 17 nm diameter and 81 nanowires in parallel for (a) stress and (b) relax comparing the hole trapping model (1) with experimental data.

IV. CONCLUSION

We investigated for the first time the reliability of JL-VNWFETs under thermal and bias stresses. Progressive degradation was observed with stress time. From our analysis it was inferred that interface traps dominate the degradation induced by long-term temperature stress whereas the bias stress degradation is mainly caused by hole trapping. This work will be further exploited for improving the compact model of JL-VNWFETs and reliability prediction of 3D logic circuits based on this technology.

ACKNOWLEDGMENT

This work was supported by the project FVLLMONTI funded by European Union's Horizon 2020 research and innovation program under grant agreement N°101016776 and by the LAAS-CNRS micro and nanotechnologies platform, a member of the Renatech French national network.

REFERENCES

- [1] Y. Guerfi, G. Larrieu, *Nanoscale Res Lett* 11, 2016.
- [2] Y. Wang, et al., *EuroSOI-ULIS*, 2023.
- [3] Y. Wang et al., *ESSDERC* 2023
- [4] B. Kaczer, et al., *IRPS*, 2008.
- [5] C. Mukherjee, et al., *IEEE Trans. Electron Dev*, 2023
- [6] Mahapatra et al., *IEEE Transactions on Electron Devices*, 2013
- [7] N. Parihar et al., *IEEE TED*, March 2016.

Optimisation pour l'implémentation FPGA des algorithmes d'apprentissage par renforcement fondés sur la fonction valeur pour la navigation autonome de robots mobiles logistiques en environnement déterministe à forte densité d'obstacles

Marouane BEN-AKKA
LGIPM

Université de Lorraine
F-57000 Metz, France

marouane.ben-akka@univ-lorraine.fr

Camel TANOUGAST
LGIPM

Université de Lorraine
F-57000 Metz, France

camel.tanougast@univ-lorraine.fr

Abstract— La planification de trajectoire évitant les obstacles est une tâche fondamentale en robotique mobile, en particulier pour les applications logistiques. L'intégration de méthodes de planification efficaces pour les robots mobiles logistiques constitue un enjeu crucial pour garantir des transports autonomes fiables et performants en entrepôt, où la navigation dans des environnements denses, la réactivité en temps réel et la faible consommation énergétique sont essentielles. Cet article propose une stratégie optimisée par epsilon-greedy à décroissance progressive, adaptée aux algorithmes d'apprentissage par renforcement fondée sur la fonction de valeur d'action. Cette stratégie est envisagée pour une intégration embarquée sur FPGA d'algorithmes de planification locale de trajectoire pour robots de transport. Des simulations et une implémentation FPGA sont présentées afin d'évaluer la performance de la politique proposée pour la sélection des actions dans un environnement déterministe caractérisé par une forte densité d'obstacles ($> 25\%$). Comparativement aux travaux similaires, la stratégie proposée et son intégration FPGA appliquée à l'algorithme classique Q-learning montrent un bon compromis entre la performance (en termes de temps de convergence), la consommation des ressources logiques et la puissance, permettant ainsi l'implémentation embarquée d'un planificateur de trajectoire temps réel pour robots logistiques.

Keywords—Planification trajectoire, robot logistique, FPGA, apprentissage par renforcement, fonction valeur d'action.

I. INTRODUCTION

Dans le domaine de la logistique des entrepôts automatisés, des centres de tri et des hubs de distribution, la navigation autonome des robots mobiles est essentielle pour assurer un transport fluide et optimisé des marchandises. Ces environnements, souvent caractérisés par une forte densité d'obstacles — tels que des étagères, d'autres robots ou du personnel —, qu'ils soient statiques ou dynamiques, exigent une prise de décision rapide en temps réel afin d'éviter les collisions et de garantir une efficacité opérationnelle [1]. L'intégration de solutions de planification de trajectoire à la fois efficaces et économes en énergie est donc cruciale pour améliorer la performance et l'autonomie des systèmes robotiques logistiques. Les algorithmes d'apprentissage par renforcement fondés sur la fonction de valeur incluent des méthodes telles que le *SARSA* (*State-Action-Reward-State-Action*), l'*Expected SARSA*, le *Q-learning*, le *Double Q-learning*, le *TD(λ)* (*Temporal Difference avec λ*) et le *Monte Carlo Control*, qui visent tous à estimer les valeurs d'action ou d'état afin de permettre à un agent d'apprendre une politique optimale en fonction des récompenses reçues. Ces algorithmes d'apprentissage par renforcement (*Reinforcement Learning*, RL) permettent à un agent de prendre des décisions optimales en interagissant avec son environnement par essais et erreurs [2]. Parmi les nombreuses applications de ces techniques, la planification de trajectoire en robotique mobile occupe une place centrale. Elle vise à déterminer une

trajectoire optimale permettant à un agent de se déplacer d'un point de départ à une cible tout en évitant les obstacles [3, 4]. La littérature distingue généralement trois niveaux de densité d'obstacles : inférieure à 10 % (faible), comprise entre 10 % et 25 % (modérée), et supérieure à 25–30 % (forte densité), en particulier lorsque la structure de l'environnement rend la navigation plus complexe. Bien que les algorithmes d'apprentissage par renforcement basés sur Q présentent de bonnes performances dans des environnements à faible densité d'obstacles, leur efficacité diminue significativement lorsque cette densité augmente, entraînant une convergence plus lente et une dégradation globale des performances. Dans le cadre applicatif de la planification de trajectoire pour les robots logistiques, une densité d'obstacles de l'ordre de 30 % est généralement considérée comme significative, notamment lorsque les obstacles sont distribués de manière aléatoire ou selon des configurations complexes. Par ailleurs, l'implémentation matérielle permettant de garantir des applications en temps réel sur des systèmes contraints en ressources et soumis à des exigences de latence reste un défi majeur. Dans ce contexte, où une faible latence et un traitement rapide des données sont requis, l'implémentation sur FPGA apparaît comme une solution pertinente [5]. Dans cet article, nous proposons une implémentation sur FPGA d'une nouvelle stratégie epsilon-greedy à décroissance progressive, visant à optimiser les approches par renforcement reposant sur l'estimation de la fonction de valeur d'action, dans le cadre de la planification de trajectoire pour robots mobiles. L'objectif est d'améliorer la vitesse de convergence, la stabilité des récompenses et l'efficacité de l'exploration. Pour valider cette approche, l'optimisation proposée est appliquée au Q-learning, implémenté sur une plateforme Xilinx Zynq, et évaluée dans un environnement maillé avec une densité d'obstacles de 30 %.

II. MODELISATION ET SIMULATION

La modélisation de l'environnement pour la planification de trajectoire d'un robot est obtenue par une discrétisation en grille, représentant les espaces structurés sous forme de cellules (libres ou obstacles). Chaque cellule correspond à une position possible, simplifiant ainsi la navigation. Des actions élémentaires de mouvement (haut, bas, gauche, droite), compatibles avec le rayon de braquage du robot mobile, permettent des déplacements unitaires entre cellules adjacentes. Cette configuration permet une exploration systématique des états possibles à travers l'estimation des valeurs Q état-action. L'environnement de simulation a été configuré pour évaluer les performances de l'agent dans une grille de taille 50×50 , avec une densité d'obstacles fixée à 30 % (Figure 1). L'algorithme implémenté simule le comportement de l'agent au sein de cet environnement, avec les paramètres suivants : taux d'apprentissage de 0,625, facteur de réduction de 0,875, nombre maximal d'étapes par

épisode de 500, et 1000 épisodes au total. Le système de récompenses est défini comme suit :

- Récompense négative de -50 en cas de collision,
- Récompense maximale de $+200$ si objectif atteint.
- Pénalité de -5 pour les états intermédiaires non terminaux.
- Pénalité de -10 pour visites répétées d'un même état au cours d'un même épisode.

Ce schéma de récompense discret permet à l'agent d'estimer efficacement les valeurs $Q(s, a)$ et de favoriser l'apprentissage de trajectoires optimales.

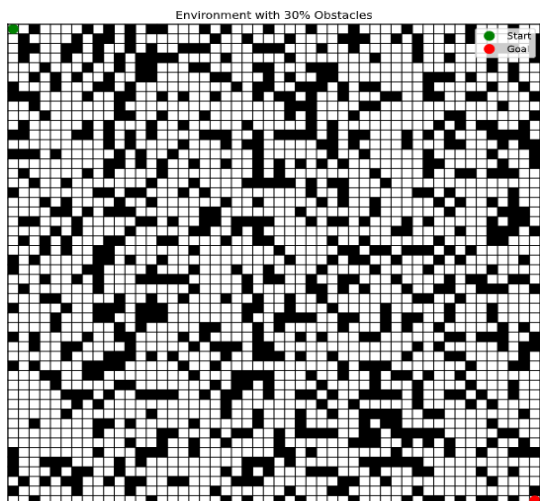


Figure 1. Environnement maillé 50×50 avec 30% d'obstacles.

L'évolution des récompenses cumulées au fil des épisodes pour un environnement comportant 30 % d'obstacles est présentée à la Figure 2. L'axe horizontal représente le nombre d'épisodes, tandis que l'axe vertical indique la récompense totale obtenue par l'agent à chaque épisode. Les résultats montrent que la combinaison de la fonction de récompense discrète et du mécanisme d'exploration adaptatif est déterminante pour les performances améliorées dans le cas du Q-Learning.

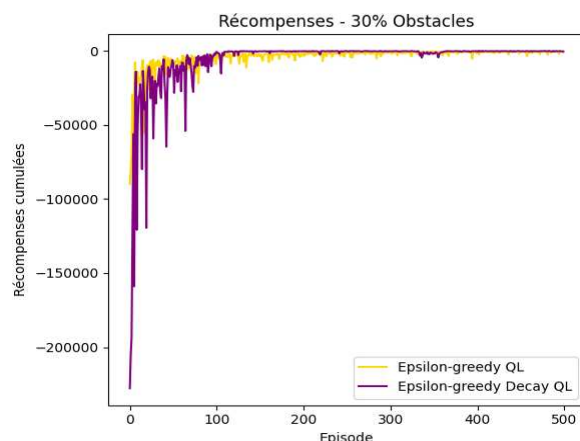


Figure 2. Variation des récompenses au fil des épisodes - Environnements maillés 50×50 /densité obstacles de 30 %.

La Figure 3 présente les meilleures performances obtenues selon la taille des mots binaires. Une politique optimale est atteinte en $77,4 \mu s$ après 98 épisodes, ou en $711 \mu s$ après 91 épisodes selon les configurations testées. L'implémentation FPGA de la stratégie proposée d'optimisation par *epsilon-greedy* à décroissance progressive a été réalisée pour la technologie Xilinx UltraScale+ ZCU104, sous

l'environnement Vivado 2022.1. Une comparaison avec des travaux antérieurs (dans les mêmes conditions : $Z = 4$ actions et technologie FPGA Xilinx ZCU) montre que la stratégie de planification par sélection d'action fondée sur la valeur, basée sur un générateur de politiques LFSR 16 bits, consomme 15 mW , 101 LUTs et 64 FFs. L'originalité principale réside dans l'intégration d'une stratégie *epsilon-greedy* décroissante, ajustant progressivement l'exploration pour renforcer l'exploitation, améliorer les retours cumulés et équilibrer temps d'apprentissage et consommation de ressources logiques.

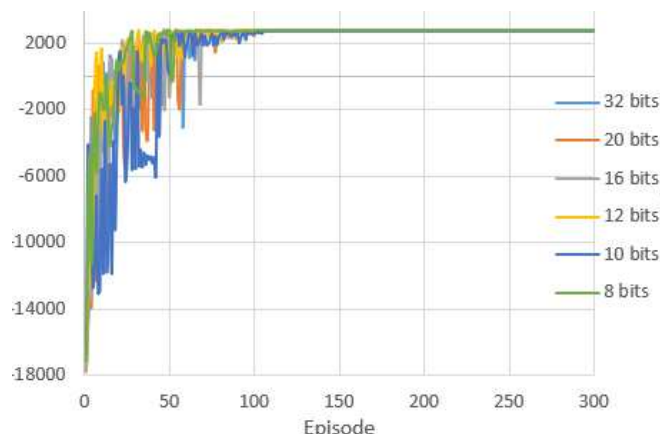


Figure 3. Récompenses totales - Environnement maillé 12×12 .

III. CONCLUSION

Les résultats expérimentaux obtenus à travers les différentes simulations montrent qu'un générateur de politiques intégrant une stratégie *epsilon-greedy* décroissante, fondée sur l'évolution des récompenses cumulées au fil des épisodes, offre un bon compromis entre temps d'apprentissage et consommation de ressources logiques. Les résultats d'implémentation sur FPGA confirment qu'un générateur de politiques basé sur un LFSR 16 bits assure une meilleure convergence dans les environnements à forte densité d'obstacles, tout en nécessitant de faibles ressources logiques et une faible consommation dynamique. En particulier, le générateur de politiques exécutant la stratégie *epsilon-greedy* décroissante fonctionne à 552 MHz , utilise 101 LUTs, 64 FFs, et consomme seulement 15 mW . Dans le cadre de travaux futurs, cette approche sera évaluée pour les algorithmes d'apprentissage par renforcement fondés sur la fonction de valeur.

REFERENCES

- [1] Farrell, S., Zarroug, M., Byram, B., Chowdhary, G., & Balaprakash, P. (2025). Safe Human Robot Navigation in Warehouse Scenario. arXiv preprint arXiv:2503.21141.
- [2] Qin, Z., Li, N., Liu, X., Liu, X., Tong, Q., Liu, X., 2021. Overview of research on model-free reinforcement learning. Computer science 48, 180–187.
- [3] Mohamed Reda, Ahmed Onsy, Amira Y Haikal, and Ali Ghanbari. Path planning algorithms in the autonomous driving system: A comprehensive review. Robotics and Autonomous Systems, 174:104630, 2024.
- [4] Younes Regragui and Najem Moussa. A real-time path planning for reducing vehicles traveling time in cooperative-intelligent transportation systems. Simulation Modelling Practice and Theory, 123:102710, 2023.
- [5] Da Silva, L.M., Torquato, M.F., Fernandes, M.A., 2018. Parallel implementation of reinforcement learning q-learning technique for fpga. IEEE Access 7, 2782–279

Système Opto-électronique Préliminaire pour l'Évaluation de l'Instabilité de la Cheville

Wenzheng Wang¹, Hang Li¹, Ibrahim Saliba², Alexandre Hardy³, Julien Denoulet¹, Sylvain Feruglio¹

¹ LIP6, CNRS UMR 7606, Sorbonne Université, 75005 Paris, France

² Département de chirurgie orthopédique, Hôpital Cochin, 75014 Paris, France

³ Département de chirurgie orthopédique, Clinique du Sport Paris, 75005 Paris, France

Résumé—Les entorses de la cheville sont des blessures sportives courantes qui peuvent évoluer vers l'instabilité latérale chronique de la cheville. Cette étude propose une méthode non invasive basée sur la spectroscopie proche infrarouge (NIRS) pour évaluer objectivement l'état des ligaments de la cheville. Un modèle de tissu multicouche centré sur le Ligament Talo-Fibulaire Antérieur (LTFA) a été construit, et des simulations optiques ont validé la pénétration suffisante des photons en réflexion. Des mesures préliminaires de réflectance sur quelques sujets ont montré que les ligaments sains avaient une réflectance plus élevée que les ligaments blessés dans des conditions exemptes d'œdème. Ces résultats confirment le potentiel du NIRS pour le développement d'outils de diagnostic portables et de faible puissance pour l'évaluation de l'intégrité des ligaments.

Mots clés—Instabilité de la cheville, système de diagnostic embarqué, système de surveillance des ligaments, NIRS

I. INTRODUCTION

Les entorses de la cheville représentent environ 15 % à 25 % des blessures liées au sport [1, 2]. Une part importante de ces cas peut évoluer vers l'instabilité latérale chronique de la cheville (CLAI, *Chronic Lateral Ankle Instability*) [3, 4], caractérisée par une instabilité récurrente et une gêne fonctionnelle. Bien que la reconstruction chirurgicale soit souvent efficace, les évaluations cliniques actuelles reposent principalement sur des questionnaires [5] et sur l'imagerie (radiographies, IRM, etc.), qui présentent des limites notables en termes d'évaluation dynamique [6], sensibilité et dépendance à l'opérateur [7–9].

La spectroscopie proche infrarouge (NIRS, *Near-Infrared Spectroscopy*), une méthode optique non invasive utilisée dans les diagnostics biomédicaux, est prometteuse pour l'évaluation de l'intégrité des ligaments [10–13]. Cet article explore une approche NIRS pour l'évaluation non invasive et objective des ligaments de la cheville, afin de soutenir l'évaluation postopératoire et inspirer le développement de systèmes de diagnostic embarqués.

II. MODÉLISATION

La structure anatomique du ligament latéral de la cheville, comme le montre la Fig. 1a, comprend différents ligaments [14], dont le Ligament Talo-Fibulaire Antérieur (LTFA). Ce ligament est le plus souvent blessé en raison de sa position et de sa fonction lors de la flexion plantaire [15–17].

Ce projet a été financé par le CNRS dans le cadre des programmes interdisciplinaires MITI, la recherche exploratoire et avec le soutien de l'IUIS.

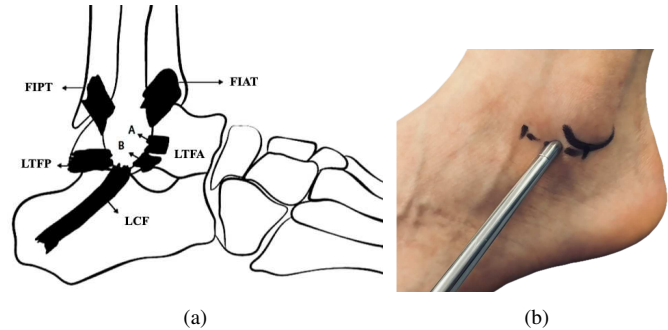


FIGURE 1 – (a) Anatomie des ligaments latéraux de la cheville [2]; (b) Disposition expérimentale pour la mesure du LTFA.

Pour étudier la faisabilité du NIRS dans l'évaluation de la fonction ligamentaire, un modèle détaillé de tissu multicouche a été construit sur la base des propriétés anatomiques et optiques [18–20]. Des simulations de Monte-Carlo ont été utilisées pour analyser la propagation des photons dans cette structure en couches en réflexion. Les résultats ont montré que les photons peuvent atteindre la couche du LTFA et en revenir [19], confirmant que les signaux réfléchis peuvent véhiculer des informations pertinentes sur l'intégrité du ligament. La distribution des trajectoires des photons a également permis de déterminer les distances entre la source et le détecteur, contribuant ainsi à optimiser la conception expérimentale.

III. MÉTHODES EXPÉRIMENTALES

AvaLight-HAL (Avantes, Pays-Bas), une source halogène compacte et stable, a été utilisée comme source lumineuse. La réflectance spectrale a été acquise par un spectromètre AvaSpec-2048XL avec une plage de détection de 450 à 1160 nm, qui a été utilisé pour mesurer la réflectance de la cheville autour du LTFA. Les données ont été traitées par le logiciel AvaSoft (v8.16).

Les mesures ont été effectuées sur cinq sujets à l'aide d'une installation standardisée, afin de minimiser les interférences de la lumière ambiante. Une sonde à fibre optique a été placée en percutané à l'emplacement anatomique du LTFA, comme illustré à la Fig. 1b. La réflectance bilatérale du ligament a été mesurée.

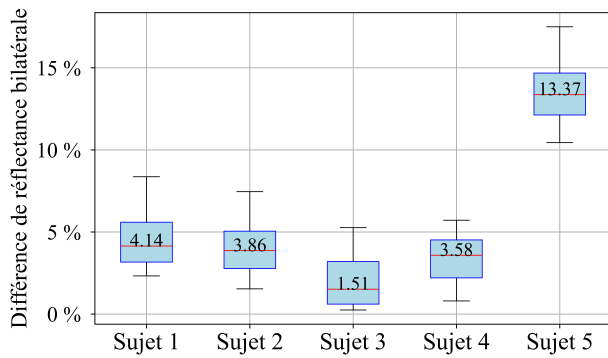


FIGURE 2 – Analyse bilatérale de la réflectance au niveau du LTFA.

IV. RÉSULTATS ET ANALYSE

Fig. 2 montre une distribution symétrique de la réflectivité bilatérale des chevilles chez quatre des cinq sujets, ce qui indique que leurs structures ligamentaires étaient dans un état considéré comme normal. Le cinquième sujet présentait une asymétrie notable due à une blessure antérieure. La réflectance était généralement plus élevée dans les ligaments sains, ce qui s'explique par leur structure tissulaire plus dense et plus uniforme. En revanche, les ligaments blessés présentaient une réflectance plus faible, liée à l'altération des fibres de collagène et à une diffusion accrue.

En outre, l'absence d'œdème a été contrôlée afin d'isoler les effets spécifiques aux ligaments sur les signaux NIRS, conformément aux directives médicales suggérant d'éviter l'évaluation de la phase aiguë.

Ces résultats suggèrent que le NIRS peut servir d'outil fiable pour déduire l'intégrité structurelle des tissus mous tels que les ligaments. Malgré la petite taille de l'échantillon, cette expérience fournit des preuves encourageantes de la sensibilité du NIRS aux changements biomécaniques des ligaments, notamment en conditions post-traumatiques ou post-opératoires.

V. APPROCHE DU DÉVELOPPEMENT DE SYSTÈMES

A. Sélection de Composants Opto-électroniques

Bien que le système Avantes ait validé le principe d'utilisation du NIRS pour l'évaluation des ligaments, sa taille, sa complexité et sa consommation d'énergie en limitent l'usage dans des scénarios portables. Par conséquent, nous avons sélectionné des composants plus appropriés pour construire un système embarqué dédié.

Pour la photodétection, la photodiode à double bande Thorlabs DSD2 a été choisie pour sa large sensibilité spectrale (400-1800 nm), rendue possible par sa structure double (silicium et d'arséniure d'indium-gallium). Au départ, un amplificateur SR570 (Stanford Research Systems, États-Unis) a été utilisé pour convertir le photocourant en tension. Cependant, en raison de son grand facteur de forme et de son manque de capacité d'intégration, nous avons développé un amplificateur de transimpédance (TIA, *Trans-Impedance Amplifier*) sur mesure. Ce TIA présente des paramètres programmables de gain et temps de réponse, permettant une adaptation dynamique à

des conditions lumineuses et des longueurs d'onde variables. Il permet de conserver un rapport signal-bruit suffisamment élevé et de garantir une détection précise de l'état des ligaments dans l'ensemble des plages spectrales concernées.

Concernant les sources lumineuses, des LED NIR et SWIR (Short-Wavelength IR) ont été sélectionnées sur la base d'études antérieures et de performances figurant sur les fiches techniques. Leur intégration modulaire permet l'extension future du système à différents types de tissus ou de conditions.

B. Proposition de Système Embarqué

L'architecture du système intégré comprend un microcontrôleur central qui coordonne l'intensité lumineuse à l'aide de potentiomètres numériques, en pilotant des LED à longueurs d'onde multiples en mode pulsé. La photodiode DSD2 reçoit les signaux réfléchis, qui sont ensuite amplifiés, filtrés et numérisés au moyen d'un Convertisseur Analogique-Numérique (CAN) multicanal à haute résolution. Des taux d'échantillonnage supérieurs à 100 Hz et une résolution d'au moins 12 bits garantissent la fidélité du signal. Les données traitées seront télétransmises à une station de base distante pour un contrôle et une analyse en quasi temps réel.

VI. CONCLUSION

Cette étude a validé la faisabilité de l'utilisation du NIRS pour évaluer la fonction des ligaments chez les patients CLAI. Les données spectrales ont montré que les ligaments sains présentaient une réflectance plus élevée que les ligaments blessés, ce qui reflète les différences dans la structure des tissus. Un système embarqué de faible puissance a été proposé, intégrant un microcontrôleur, une photodiode à double bande et un double TIA reconfigurable, permettant une surveillance en temps réel et portable des tissus biologiques.

Malgré la taille limitée de l'échantillon, les résultats encourageants posent les bases du développement d'outils de diagnostic précis et portables. Ce cadre offre une solution prometteuse pour l'évaluation postopératoire objective et le traitement personnalisé des lésions ligamentaires.

RÉFÉRENCES

- [1] M. Machado et al. 2021. DOI : 10.1177/22104917211035552
- [2] I. Saliba et al. 2024. DOI : 10.3390/jcm13020442
- [3] M. Drakos et al. 2022. DOI : 10.1016/j.fcl.2021.11.025
- [4] T.S. Bestwick et al. 2021. DOI : 10.1186/s12891-021-04230-8
- [5] R. Martin et al. 2021. DOI : 10.2519/jospt.2021.0302
- [6] S. Chang et al. 2021. DOI : 10.5435/JAAOS-D-20-00145
- [7] S. Guillo et al. 2013. DOI : 10.1016/j.otsr.2013.10.009
- [8] S. Jolman et al. 2017. DOI : 10.1177/1071100716685526
- [9] S. Alshalawi et al. 2018. DOI : 10.1016/j.fcl.2018.07.008
- [10] N. Mainard et al. 2022. DOI : 10.3390/s22103840
- [11] R. Li et al. 2022. DOI : 10.3390/s22155865
- [12] M. Saleh et al. 2025. DOI : 10.1007/978-981-97-9294-8_3
- [13] J. Torniaainen et al. 2022. DOI : 10.1371/journal.pone.0263280
- [14] F. Taser et al. 2006. DOI : 10.1007/s00276-006-0112-1
- [15] J. Vega et al. 2020. DOI : 10.1007/s00167-017-4736-y
- [16] B. Hintermann et al. 2002. DOI : 10.1177/03635465020300031601
- [17] J. Vega et al. 2014. DOI : 10.1007/s00167-017-4736-y
- [18] S.L. Jacques. 2022. DOI : 10.1117/1.JBO.27.8.083002
- [19] W. Wang et al. 2024. DOI : 10.1109/ICECS61496.2024.10849250
- [20] A. Bashkatov et al. 2011. DOI : 10.1142/S1793545811001319

Toward a generic and flexible architecture for AI hardware

Mustafa Ibrahim
Nantes Université, IETR UMR 6164
Nantes, France
mustafa.ibrahim@univ-nantes.fr

Andrea Pinna
Sorbonne Université, LIP6
Paris, France
andrea.pinna@lip6.fr

Sebastien Pillement
Nantes Université, IETR UMR 6164
Nantes, France
sebastien.pillement@univ-nantes.fr

Abstract—This paper provides an overview of AI algorithms and hardware accelerators. We examine AI algorithms focusing on their computational needs and areas of application. Furthermore, we analyze various AI accelerators, highlighting their flexibility, and suitability for different AI workloads.

Index Terms—Artificial Intelligence, Embedded Systems, Hardware Design

I. INTRODUCTION

Artificial Intelligence (AI) algorithms are now fundamental to a variety of applications, including image recognition, natural language processing, speech recognition, and autonomous driving. Each algorithm has distinct computational and memory access patterns, posing challenges when deployed on hardware. This paper explores these challenges by outlining the requirements for designing a flexible edge architecture capable of efficiently supporting a wide range of AI applications.

II. AI ALGORITHMS

A Multilayer Perceptron (MLP) is a feedforward neural network consisting of an input layer, one or more hidden layers, and an output layer. It processes inputs by performing matrix-vector multiplications, essentially a series of dot products, followed by nonlinear activation functions. This allows the network to learn and represent complex patterns in the data. Convolutional Neural Networks (CNNs) [1] are neural networks specifically designed for image processing tasks. They use convolutional layers to apply filters that detect visual patterns such as edges and textures through dot product operations. Pooling layers, based on either comparisons or averaging, are used to reduce dimensionality while retaining important features. ReLU is the most commonly used activation function in CNNs. This structure effectively transforms raw images into compact feature representations suitable for tasks like classification.

Recurrent Neural Networks (RNNs) [2] are used for sequential data. Unlike MLPs, RNNs have a feedback loop, where each hidden layer receives input from both the previous layer and its prior state, allowing it to capture temporal patterns. This makes RNNs ideal for tasks like time-series analysis and language modeling.

This work is supported by France 2030 Priority research program and equipment for artificial intelligence PEPR AI, under the ref ANR-23-PEIA-0009.

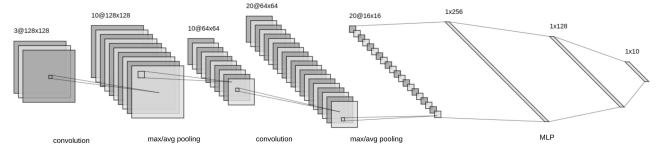


Fig. 1: Convolutional Neural Network

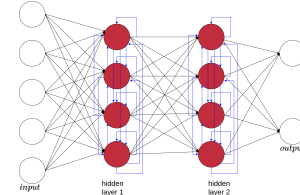


Fig. 2: Recurrent Neural Network

Long Short-Term Memory (LSTM) networks [2] enhance standard RNNs by effectively handling long-term dependencies in sequential data. They incorporate memory cells along with input, forget, and output gates, and use element-wise (Hadamard) multiplication to regulate information flow. This design allows the network to preserve important information across extended sequences. Common activation functions used in LSTMs include sigmoid and tanh.

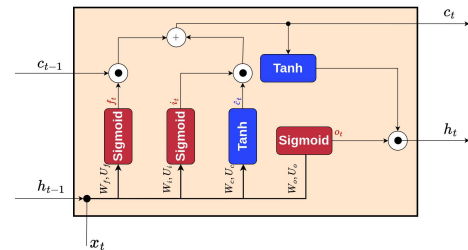


Fig. 3: Recurrent Neural Network

Transformers [3] are advanced neural network architectures designed for sequence-based tasks. Unlike RNNs and LSTMs that handle data sequentially, Transformers use self-attention

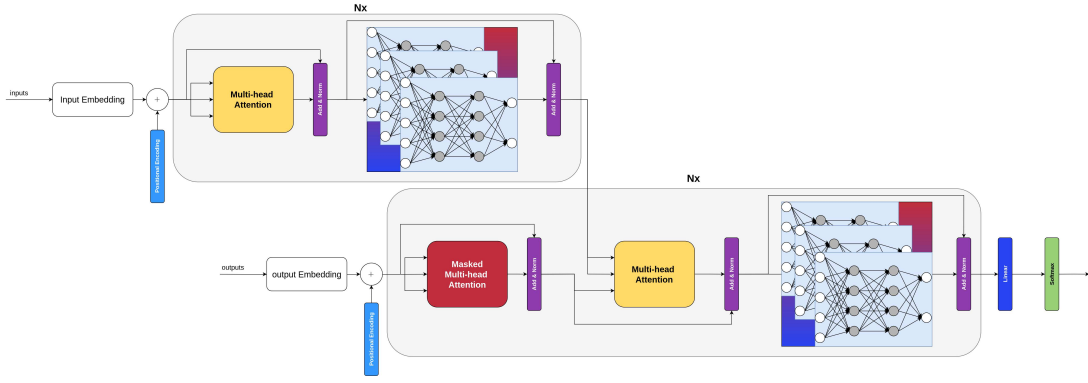


Fig. 4: Transformer neural network

mechanisms built on matrix multiplications and softmax operations to process entire input sequences in parallel. This makes them highly efficient for applications such as natural language processing.

III. HARDWARE ACCELERATORS FOR AI

NVIDIA's general-purpose GPUs offer strong flexibility and parallelism through their Single Instruction Multiple Thread (SIMT) architecture. Ampere based GPUs [4], for example, feature multiple Streaming Multiprocessors (SMs), each running 128 threads in parallel ideal for large scale, data parallel tasks. However, this performance comes at a cost. GPUs consume more power than specialized accelerators and, while efficient for batch processing, can introduce higher latency, making them less suitable for real-time inference.

Google's Tensor Processing Units (TPUs) [5] are AI accelerators built for efficient machine learning, using large systolic arrays for fast matrix operations. However, their fixed architecture can limit performance for algorithms with irregular computations. Early versions used hardwired activation functions and 8-bit data types, reducing flexibility. Newer TPUs address this with general-purpose vector processors and support for bfloat16, making them better suited for large-scale cloud and data center deployments.

Field-Programmable Gate Array (FPGA) SoCs provide high flexibility through reconfigurable fabric, including resources like LUTs, BRAMs, and DSPs, allowing bit-level customization and support for various data types. However, larger data types like float32 consume more resources and introduce higher latency. These devices also include multi-core CPUs and external memory controllers for hybrid hardware-software designs. Implementing an FPGA-based solution requires generating a bitstream via synthesis and place-and-route processes. Modern FPGA SoCs support Dynamic Partial Reconfiguration (DPR), enabling specific regions to be reconfigured at runtime without affecting the rest of the system. While this improves resource utilization and adaptability, DPR remains slow, typically taking several milliseconds due to limitations in current reconfiguration controller speeds.

A. Requirement for the proposed architecture

The target architecture should be able to dynamically adapt to a variety of AI algorithms, such as MLPs, CNNs, LSTMs, Transformers, etc., by utilizing fast and efficient dynamic reconfiguration to perform different arithmetic operations (e.g., matrix-vector multiplication, Hadamard product, pooling, softmax). It must also support a wide range of nonlinear activation functions (e.g., Sigmoid, Tanh, ReLU, GELU) through approximations with configurable error tolerances. Furthermore, the architecture must be compatible with multiple data types, including float32, float16, bfloat16, float8, fixed-point, and int8, to ensure computational flexibility. Lastly, it should be energy-efficient to facilitate integration into power-constrained edge environments.

IV. CONCLUSION

In conclusion, AI algorithms differ in their computational and memory patterns, posing challenges for hardware implementation. While GPUs and TPUs support a wide range of models, they are not ideal for edge environments. FPGAs offer reconfigurability but at the cost of high overhead. Existing edge accelerators lack the flexibility to support diverse algorithms and data types. This thesis aims to develop a generic, edge-focused AI accelerator that supports multiple AI models, activation functions, and data types

REFERENCES

- [1] Maria Vakalopoulou, Stergios Christodoulidis, Ninon Burgos, Olivier Colliot, and Vincent Lepetit. Deep learning: basics and convolutional neural networks (cnns). *Machine learning for brain disorders*, pages 77–115, 2023.
- [2] Robin M Schmidt. Recurrent neural networks (rnns): A gentle introduction and overview. *arXiv preprint arXiv:1912.05911*, 2019.
- [3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [4] R NVIDIA. Nvidia ampere ga102 gpu architecture, 2020.
- [5] Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th annual international symposium on computer architecture*, pages 1–12, 2017.

FPGA implementation of a multi-application Growing Neural Gas model Using Network-On-Chip

Florent DERUE, Slavisa JOVANOVIĆ, Hassan RABAH and Serge WEBER
Université de Lorraine, CNRS, IJL, F-54000 Nancy, France
 firstName.lastName@univ-lorraine.fr

Abstract—This work presents the design and implementation of a Growing Neural Gas (GnG) model using an array of neurons interconnected via a Network-on-Chip (NoC) for efficient inter-neuron communication. The proposed architecture is scalable and enables independent control of individual neurons, allowing the simultaneous processing of multiple application vectors. These features, combined with the GnG model, provide the system with Continual Learning capabilities allowing to adapt to statistical variations of incoming data. The design is validated through SystemC simulation and described in VHDL for hardware implementation.

Index Terms—Hardware implementation, unsupervised learning, vector quantization, continual learning, incremental learning, Growing Neural Gas (GNG).

I. INTRODUCTION

Machine Learning (ML) and Neural Networks (NN) are powerful tools for extracting insights from data but require significant computational resources, making them challenging to deploy in embedded systems. Additionally, traditional deep learning models struggle with changing data distributions, leading to performance degradation [1].

Continual learning methods, such as replay-based techniques, resource allocation, and regularization [2], [3], attempt to address this issue but often require extensive resources or reduce model adaptability. Prototype-based models, like Self-Organizing Maps (SOM) and their variants (GNG, GWR), offer a promising alternative by maintaining plasticity and mitigating catastrophic forgetting [4], [5].

Among these, dynamic models like Growing Neural Gas (GNG) and Growing-When-Required (GWR) excel in evolving environments but lack scalable hardware implementations. Existing FPGA-based solutions [6]–[8] are limited in neuron capacity or rely on simplified designs, restricting their use in continual learning.

This work introduces a scalable hardware architecture capable of hosting and training multiple GNG graphs simultaneously, enhancing neuron utilization and enabling continual learning. This paper details the proposed approach in Section II, the implementation in Section III, the validation and experimental results in Section IV, and concludes with future research directions in Section V.

II. CONTINUAL LEARNING WITH GROWING NEURAL GAS

The proposed work presents a continual learning approach summarized in Algorithm 1, where input data X is processed sequentially, detecting changes in its probability distribution $p_t(X, \theta_t)$ over time. The Growing Neural Gas (GNG) algorithm is used as a mapping function F (line 4), dynamically adjusting the network structure to evolving data patterns. Initially, data is mapped to a graph G_1 , which is continuously compared with new inputs. If significant changes are detected, either a new mapping is created or an existing one is recalled from a set of stored graphs Γ .

GNG, introduced by Fritzke in 1994 [9], a flexible self-organizing neural model [10], allows neurons to evolve dynamically, forming a graph structure rather than a fixed grid. It identifies the two closest neurons BMUs (Best Matching Units), updates the connections, removes outdated edges, and inserts new neurons in high-error regions to improve data quantization.

Algorithm 1: Continual learning approach

```

1 Input:  $X, p_t(X, \theta_t), \epsilon$ 
2 Output:  $\Gamma$ 
3  $\Gamma \leftarrow \emptyset$ 
4  $\Pi_{G_1}(X) = \{F(x_i) | x_i \in X, p(x_i) = p_{t_1}(X, \theta_{t_1})\}$ 
5  $\Gamma \leftarrow \Gamma \cup G_1$ 
6 while  $X$  do
7   if  $D(p_t(X, \theta_t), p_{t-1}(X, \theta_{t-1})) > \epsilon$  then
8     if  $\nexists \Pi_{G_j}(X), p_t(X, \theta_t) = p_j(X, \theta_j), G_j \in \Gamma$  then
9        $\Pi_{G_k}(X) = \{F(x_i) | x_i \in X, p(x_i) = p_t(X, \theta_t)\}$ 
10       $\Gamma \leftarrow \Gamma \cup G_k$ 
11 return  $\Gamma$ 

```

pdf - probability density function

X - input data set

$p_t(X, \theta_t)$ - time-variant pdf of X with time-variant parameters θ_t

F - a function mapping X onto graph G (i.e. GNG)

Γ - a set of graphs

$\Pi_{G_1}(X)$ - projection of X on graph G_1 built with the function F

$D(p_t(X, \theta_t), p_{t-1}(X, \theta_{t-1}))$ - metrics to compare p_t and p_{t-1}

ϵ - threshold indicating the change in X (based on the comparison of pdfs)

To support this continual learning framework, a hardware architecture is proposed for managing multiple GNG graphs in parallel thus, enabling real-time training and adaptation to dynamic environments.

III. MULTI GNG GRAPH HARDWARE ARCHITECTURE

The SWAP-GNG architecture is a NoC-based system designed for multi-GNG graph learning. Each neuron is connected through a NoC router, enabling flexible communication and all-to-all connectivity. A key feature is the ability to share neurons across multiple graphs through shared weight memory organization, allowing inference or training on multiple graphs simultaneously.

The architecture supports independent and parallel learning of multiple graphs while maintaining access to previously trained ones. It consists of three layers as shown in Fig. 1: the Neural Layer (NL), responsible for distance computation, BMU search, and edge management; the NoC layer, handling communication via a 2D mesh NoC using XY routing and messages acknowledgment for neuron occupancy; and the GNG Manager (GNGM), which supervises network initialization, input dispatching, and neuron growth.

Learning iterations involve sequential squared Euclidean distance computations, BMU searches, and edge updates, with execution times depending on network size and neuron count. The NoC efficiently routes messages for neuron interactions, while a unique graph code ensures correct memory access and offsets (Fig. 1) during computations.

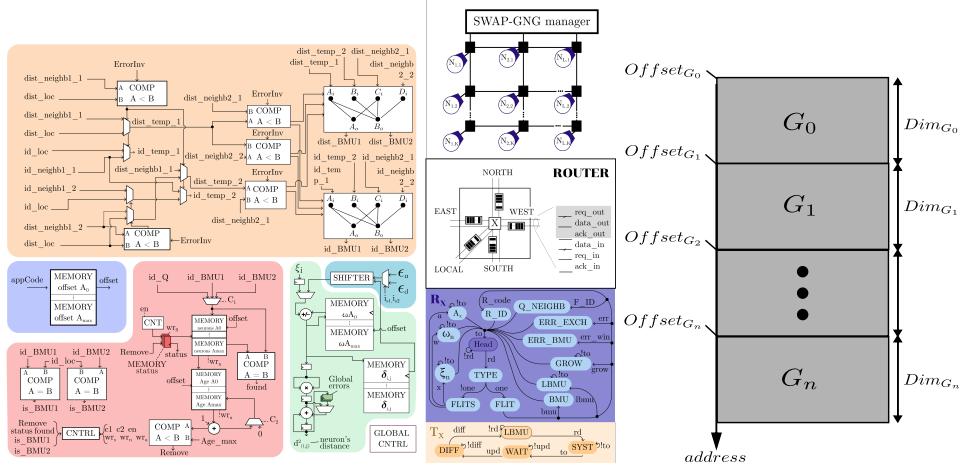


Figure 1: (a) SWAP-GNG architecture (b) Weight memory organization

IV. RESULTS

The proposed architecture is designed with VHDL at the RTL level for FPGA synthesis but also validated through SystemC simulations. A continual learning scenario involving three sequentially learned graphs is used to test its effectiveness, with input data representing geometric shapes: a rectangle, a triangle, and a circle. Data points are fed into the architecture sequentially, and neuron weights are adjusted dynamically to match the underlying data distributions.

Results in Fig. 2 show that neurons initially distribute sparsely but progressively align with the input data, capturing the geometric structures. Even though learning occurs sequentially, all trained graphs remain stored and can be retrieved for further learning or inference. The architecture, implemented as SWAP-GNG, achieves a maximum frequency of 66.6 MHz on a Xilinx XC7Z020CLG484-1 FPGA and efficiently utilizes BRAM for storage and DSP for computations (Table I).

The ability to handle multiple graphs simultaneously while adapting to evolving data distributions highlights the flexibility and efficiency of SWAP-GNG. It enables independent learning processes to start at any time and integrates new knowledge dynamically when statistical changes in data occur, making it ideal for continual learning environments.

V. CONCLUSION

This work presents a novel architecture utilizing a Growing Neural Gas (GNG) neural network as the foundation for continual learning. The design employs a memory-shared approach to minimize hardware overhead while enabling the implementation of multiple GNG graphs. It is highly connected, scalable, and distributed, allowing

simultaneous learning of diverse data distributions, making it well-suited for continual learning. The proposed architecture is validated through SystemC simulations and VHDL synthesis for performance analysis. Future work includes experimental validation and application in real-world scenarios as well as the exploration of advanced distance metrics for graph comparison.

Table I: Synthesis results on Xilinx XC7Z020CLG484-1 for one neuron/router pair.

Resources	Total	Total%	Proposed arch.	
			Neuron	Router
# of Slice LUTs	1853	0.03%	64%	36%
# of Slice FF	1261	0.01%	50%	50%
# of DSP	2	0.009%	100%	0
# of BRAM	2	0.014%	100%	0

REFERENCES

- [1] O. A. Mahdi, N. Ali, E. Pardede, A. Alazab, T. Al-Quraishi, and B. Das, "Roadmap of Concept Drift Adaptation in Data Stream Mining, Years Later," *IEEE Access*, vol. 12, pp. 21129–21146, 2024.
- [2] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, May 2019.
- [3] S. Tian, L. Li, W. Li, H. Ran, X. Ning, and P. Tiwari, "A survey on few-shot class-incremental learning," *Neural Networks*, vol. 169, pp. 307–324, Jan. 2024.
- [4] Y. Wei, J. Ye, Z. Huang, J. Zhang, and H. Shan, "Online prototype learning for online continual learning," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18718–18728, 2023.
- [5] S. Ho, M. Liu, L. Du, L. Gao, and Y. Xiang, "Prototype-Guided Memory Replay for Continual Learning," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–11, 2024.
- [6] G. La Tona, M. Luna, M. C. Di Piazza, M. Pucci, and A. Accetta, "Development of a high-performance, fpga-based virtual anemometer for model-based mppt of wind generators," *Electronics*, vol. 9, no. 1, 2020.
- [7] Z. Mohammadi, J. M. Kincaid, S. H. Pun, A. Klug, C. Liu, and T. C. Lei, "Computationally inexpensive enhanced growing neural gas algorithm for real-time adaptive neural spike clustering," *Journal of Neural Engineering*, vol. 16, 2019.
- [8] F. Flórez-Revuelta, J. M. García-Chamizo, J. García-Rodríguez, A. Fuster-Guilló, and J. Azorín-López, "A topology-preserving growing neural network for real-time representation of objects and their motion," *New Trends in Real-Time Artificial Intelligence (NTERTAIn)*, p. 44, 2006.
- [9] B. Fritzke, "A growing neural gas network learns topologies," *Neural Information Processing Systems*, vol. 7, 03 1995.
- [10] S. Jovanović and H. Hikawa, "A survey of hardware self-organizing maps," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 11, pp. 8154–8173, 2023.

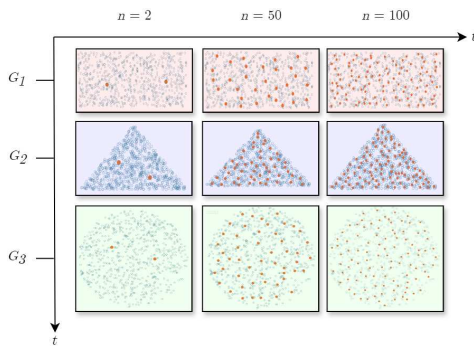


Figure 2: SWAP-GNG graphs for three different geometric-shaped data distributions.

Analyzing Embedded AI Vulnerabilities in Google’s LiteRT Through Fault Injection

Leonardo Alexandrino de Melo^{✉*}, Alberto Bosio^{✉*}, Rodrigo Possamai Bastos^{✉†}

^{*} Ecole Centrale de Lyon, INSA Lyon, CNRS, UCBL, CPE Lyon, INL, Ecully, France.

[†] Univ. Grenoble Alpes, CNRS, Grenoble INP, TIMA, 38000 Grenoble, France.

Abstract—Evaluating the reliability of deep neural network (DNN) applications is essential for deploying artificial intelligence (AI) on safety-critical embedded systems. This work presents a novel methodology for assessing reliability of DNN models deployed on Microcontrollers using software-level fault injection. A new tool, called MicroFI, was proposed and validated with two case studies, providing a comparative analysis of DNN reliability versus register and memory bit flips, induced by transient faults.

Index Terms—Neural Networks, Fault Tolerance, Fault Detection, Reliability Estimation, Embedded Systems

I. INTRODUCTION

Deep neural networks (DNNs) are among the most used predictive models in Machine Learning (ML), excelling in many complex computational tasks. These high demanding applications require processing large volumes of sensor data, posing significant challenges in power consumption, latency, and security when relying on cloud-based computation. To address these constraints, there is a growing shift toward data processing at the edge. Tiny machine learning (TinyML) has emerged as a promising field to integrate ML capabilities into resource-constrained embedded platforms.

The growing adoption of TinyML further drove the development of LiteRT for Microcontrollers (TFLM), a specialized version of TensorFlow (TF), designed to run ML models efficiently on resource-constrained devices by minimizing memory usage and processing demands, running computations on optimized C and assembly.

Faults caused by ionizing radiation occurring post-training can significantly disrupt system functionality [1]. They may result in prediction errors that degrade application performance, posing serious risks in critical scenarios [2]. Ensuring the reliability of HW-AI systems is therefore essential, particularly for safety-critical and mission-critical applications where failures could endanger human health.

The strong interest to assess the reliability of commercial off-the-shelf (COTS) boards, allied with the constraints from cloud-based computing systems, plays a critical role in enabling scalable and cost-effective HW-AI deployments. This work is motivated by the needs of the industry to evaluate and improve the reliability of AI applications on embedded platforms. The existing frameworks in the literature are designed for specific use cases and do not address general reliability assessment of DNN models deployed on TFLM. So we developed MicroFI, a fast and configurable framework for runtime Fault Injection (FI) on TFLM.

II. METHODOLOGY

TensorFlow models were defined and compiled using Keras operations in Python. Training was conducted on the MNIST dataset - where handwritten numbers are ranked in 10 classes. The target hardware platform is the ESP32C3-Mini-1, a modern single-core microcontroller from Espressif.

A. MemoryFI

MemoryFI is the first core feature of MicroFI, designed to inject transient faults into the FlatBuffer stored in the device’s DRAM. This enables targeted manipulation of parameters such as kernels, weights, biases, and other model components.

By leveraging the topology of the neural network, users can specify tensor names or indices as targets, and generate associated fault lists, streamlining the fault injection setup.

At runtime, MicroFI accesses the designated memory address corresponding to a specific tensor and introduces a transient fault - a single bit flip - immediately before the operation. After the operation is completed, the bit is restored to its original state to accommodate truthful transient injection for possible time redundant layers or parameter reuse, this operation monitoring functionality was achieved by inserting a small block of code on the TFLM micro interpreter library. The MemoryFI method is fully compatible with any TensorFlow Lite model, requiring no modifications to the model itself. Intrusion is minimal, and the runtime overhead was calculated to be around 182 CPU cycles, or $1.1\mu s$ for the ESP32C3’s 160MHz CPU, making it an effective and practical tool for fault injection in resource-constrained embedded systems.

B. RegisterFI

RegisterFI introduces transient faults directly during inference, targeting intermediate values, input tensors, output tensors, as well as control flow data structure at the level of the CPU registers.

In an embedded system, when a task is interrupted, the kernel saves its execution context, including register values, stack, and program counter. Upon resumption, its context is restored to ensure the task can continue from where it left off. The process of saving and restoring a task’s context is known as context switching. This mechanism is fundamental to multitasking systems, and a simplified version also applies to function calls. The stack pointer indicates the memory location where the function’s stack resides, allowing access to saved register values.

RegisterFI exploits context switching to introduce faults during inference. To achieve this, the `esp_timer` component from the ESP-IDF framework was used to configure a one-shot high-resolution timer. A one-shot timer is a timer that triggers an ISR after a predefined interval, executing its callback function only once.

The duration of each operation is profiled and saved during golden inferences. Operation monitoring was implemented in the micro interpreter, function responsible for managing the inference. The FI sequence initiates as soon as the target operation begins, RegisterFI starts a one-shot timer for a random time within the duration of the operation. When the timer expires, it triggers a context switch, interrupting the inference and executing the timer's callback function. The callback accesses the stack in memory where the last function's register are stored. It then flips a random bit in one of the 32-bit saved register values. The inference is resumed from the instruction it was interrupted, and the modified register value is reloaded into the CPU registers. The injected fault then propagates through the *operator*, the *graph*, and subsequent layers, simulating the real-world effects of a transient fault [1]. The measured runtime overhead for each injection, including operation monitoring, timer set up, bit flip, and context switching is around 1300 CPU cycles, or $8.1\mu s$ for the tested CPU.

To evaluate the behavior of a DNN under fault injection, MicroFI analyzes the predicted outputs of a faulty model against a fault-free baseline called golden model. Faulty predictions are classified as per the methodology adapted from [2], with few additions.

III. RESULTS AND DISCUSSION

The minimum overhead added by the framework - 1300 cycles or $8.1\mu s$ for RegisterFI and 182 cycles or $1.1\mu s$ for MemoryFI can be compared with other solutions, such as the GDB. The GDB overhead can be roughly estimated for ESP-PROG, Espressif's debugging board. It offers JTAG communication with the ESP32C3 and serial communication with the computer. MicroFI can be three orders of magnitude faster than debugging solutions.

In the first case study, we analyze LeNet5 [3], with 3 cConvolutional layers (Conv) and 2 fully connected layers (FC), using both MemoryFI and RegisterFI. Conv1 exhibited the highest proportion of Critical faults in both cases, highlighting its vulnerability during the early stages of feature extraction. However, on MemoryFI campaign, Conv1 showed about 4 times more faults than the other convolutional layers, while on RegisterFI campaign, Conv2 was almost as faulty as Conv1, which still had twice the number of faults compared to Conv3. From memory to register faults, Conv2 became less resilient than both FCs. It is worth noting that 41.9% of Observed faults in the memory were actually beneficial to model accuracy, with 42.1% classified as Acceptable. In contrast, 21.6% of the Observed register faults were classified as Critical. Register faults reduced accuracy to 96.04%, a drop of 3.16% from the golden accuracy. Memory faults caused

a smaller reduction, with a loss of 0.28% to a baseline of 98.00%.

This assessment illustrates the sensitivity of the CPU registers and highlights how FIs during calculations significantly impact the overall system reliability, when compared to FIs on the parameters. It is then imperative to inject faults in the registers to have a trustful evaluation of system reliability. In the next section, results compare a variety of DNNs on a RegisterFI campaign.

The second case study uses LeNet5 [3], LCNN [4] with 4 Conv and 2 FC, and a modified version of TinyVGG [5] with 6 Conv and 4 FC with reduced parameters.

Fault masking is extremely high for LCNN, with 99.89% or more faults being masked in all layers, showing strong inherent resilience to register faults. It is also the largest model, with 275KB, versus 127KB of M-TinyVGG, and 98KB of LeNet-5 for the `.tfLite` file. However, reliability is not only dictated by parameter count, as we can see with LeNet-5, which had an accuracy loss of 4.14% against 8.94% of the 30% larger M-TinyVGG, the number of layers play a role in increasing the failure rate for the last Conv layers of the model. M-TinyVGG also had more Critical faults, with 19.59% compared to 7.56% of LeNet-5. Comparing the LeNet-5 performance between both case studies, one can raise the hypothesis that its FC layers are more prone to fail in more difficult datasets, such as Fashion MNIST.

Crashes are substantial, and make up for 48% of all Observed faults, followed by Critical faults with 21%. FC layers are often more susceptible to Critical faults and Crashes than convolutional layers, likely due to the dense interconnections and reliance on precise register values. Another takeaway is that having fewer layers with more parameters leads to a more robust network.

In this study, we introduced a novel methodology for evaluating the reliability of DNN applications on TensorFlow Lite for Microcontrollers using SFI. This methodology was embodied in a specialized tool named MicroFI, designed to inject faults in both memory and CPU registers during runtime with minimum overhead. Results demonstrated how memory faults compares to register faults on a RISC-V ESP32C3, revealing the importance of a hardware-aware CPU fault injection to reliability assessment.

REFERENCES

- [1] G. D. Natale, D. Gizopoulos, S. D. Carlo, A. Bosio, and R. Canal, *Cross-Layer Reliability of Computing Systems*. The Institution of Engineering and Technology, 2020.
- [2] A. Ruospo, E. Sanchez, M. Traiola, I. O'Connor, and A. Bosio, "Investigating data representation for efficient and reliable convolutional neural networks," *Microprocessors and Microsystems*, vol. 86, p. 104318, 2021.
- [3] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [4] M. Begum, M. Hasan Shuvo, M. Kamal Nasir, A. Hossain, M. Jakir Hossain, I. Ashraf, J. Uddin, and M. A. Samad, "Lcnn: Lightweight cnn architecture for software defect feature identification using explainable ai," *IEEE Access*, vol. 12, pp. 55 744–55 756, 2024.
- [5] X. Fang, "Understanding deep learning via backtracking and deconvolution," *Journal of Big Data*, vol. 4, pp. 1–14, 2017.

Wireless, Energy-Autonomous Embedded System for Real-Time Monitoring of Intracranial Pressure

Ollie Chapman
LAAS-CNRS, Université de
Toulouse, CNRS
Toulouse, FRANCE
ollie.chapman@laas.fr

Gaël Loubet
LAAS-CNRS, Université de
Toulouse, CNRS, INSA
Toulouse, FRANCE
gael.loubet@laas.fr

Alexandru Takacs
LAAS-CNRS, Université de
Toulouse, CNRS, UPS
Toulouse, FRANCE
alexandru.takacs@laas.fr

Daniela Dragomirescu
LAAS-CNRS, Université de
Toulouse, CNRS, INSA
Toulouse, FRANCE
daniela.dragomirescu@laas.fr

Abstract—This paper presents a novel system for reading a resonant intracranial passive sensor dedicated to monitoring the intracranial pressure. This sensor is remotely excited by a Voltage-Controlled Oscillator and its impedance is accurately read via a coupled coil, that leads back to the pressure value. The proposed system utilizes a fully analog, low-cost design, making it a promising solution for future medical applications requiring real-time intracranial pressure monitoring.

Index Terms—Intracranial Pressure, LC Sensor, Impedance Reader, Embedded System, Wireless Sensing

I. INTRODUCTION

Intracranial pressure (ICP) is the pressure exerted by the cerebrospinal fluid, blood and brain tissue on the rigid skull [1]. An increase in ICP is a life threatening problem. It can occur after traumatic brain injury, brain cancer, stroke, in neurodegenerative disease or even during exposure to changes in gravity such as spaceflight. Normal ranges of ICP is 5-15 mmHg (7-20 mbar) [2], whereby the Brain Trauma Foundation recommends treatment for ICP above 20 mmHg. ICP values are not stable but form pulsatile waves (amplitude around 1 mmHg). It also depend on the head position, body temperature and even oxygenation status. Continuous and sensitive ICP monitoring is critical for better diagnosis and surveillance. It facilitates brain injury evaluation, treatment efficacy, and patient survival.

Today, ICP measurement requires the implantation of a wired pressure sensor directly in the skull [3]. Indirect methods – based on external measurements – do not provide information sufficiently relevant or accurate for diagnosis [4]. Gold standard ICP sensors are based on a wired piezoelectrical pressure sensor that are bulky and highly invasive for implantation. These are important constraints for clinical use, and -to date- are unsuitable for use in research models that often employ rodents. In this last case, a less invasive implantable ICP sensor of a maximum size of 9 mm² is required. These researches would enable better comprehension of the functioning and regulation of ICP.

Our main objective in the framework of the ANR WISPerS project is to design an external autonomous and battery free, cutaneous wearable patch to interrogate a passive transducer wirelessly.

II. INTRACRANIAL PRESSURE MEASUREMENT

A. Sensor principle

Micro-electromechanical system (MEMS) are ubiquitous in many applications, but are economically viable only in large-scale production. With emerging 3D-printing technologies, MEMS rapid prototyping can be done, using various principles (e.g. fused material deposition, stereo-lithography, 2-photon stereo-lithography, etc.). The targeted passive transducer, composed of a planar coil, two electrodes in a capacitor configuration and bundled in a bio-compatible substrate, will act as an LC sensor operating at its modulated resonant frequency, which leads back to the ICP.

Keeping the resonant frequency, defined by $f_{res} = \frac{1}{2\pi\sqrt{LC}}$, under 100 MHz allows signals to pass through the skull and skin more easily than at higher frequencies, increasing its resilience. Unfortunately, constraints in size and working frequency leads to a coil low quality factor (Q), meaning optimization on its design and readout system is crucial.

B. Inductive reader system principle

To interrogate this passive transducer using coupled coil mechanisms, different techniques can be used, such as phase [5] or real [6], [7] part measurement of reader coil impedance, a combination of both [8], as well as time-domain measurements [9], [10].

As our system needs to be independent from the coupling factor k (see Fig.1) because of changes in distance and misalignment between sensor's and reader's coils, measurement of the real part at reader's coil is well suited as it only acts as an amplitude factor:

$$Re(Z_{reader})_{max} = R_s + 2\pi f_{res} L_{sensor} k^2 Q \quad (1)$$

III. LC TANK RESONANT FREQUENCY MEASUREMENT

A. Impedance measurement

Most readout circuit nowadays operates on lower/higher frequency range [11], [12], or have too high power consumption [6]. We propose a full-analog impedance reader, based on a voltage controlled oscillator (VCO), a low noise amplifier (LNA), a low-pass filter (LPF) and a peak detector, tested with an Arduino with integrated analog-to-digital (ADC) and

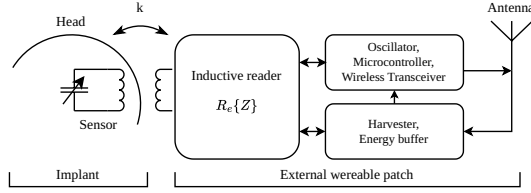


Fig. 1: Project's architecture overview

digital-to-analog (DAC) for a proof-of-concept (see Fig.2). Tests were done using commercial, fixed k , coupled coils with different capacitor and compared with reference measurements obtained with a vector network analyzer (VNA) at the reader's coil ports (see Fig.3).

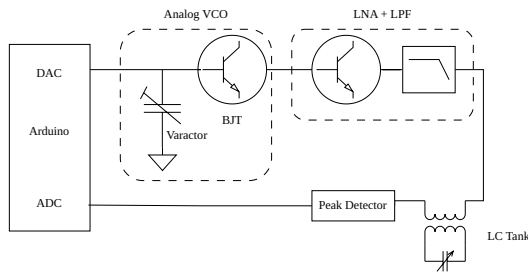


Fig. 2: Analog readout system connected to an Arduino

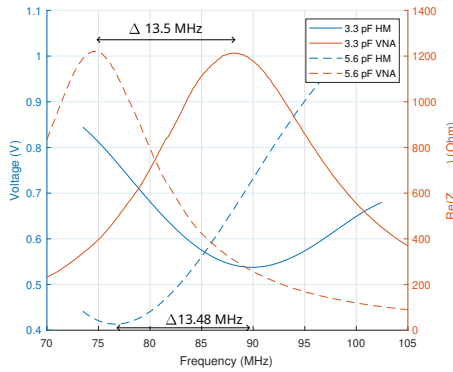


Fig. 3: Comparison between the variation of the voltage measured with the home-made (HM) analog readout system (blue) and $Re(Z_{reader})$ measured with a VNA (orange) at the output ports of the reader coil for different capacitor values

Experimental results have confirmed that the voltage measurements obtained with the proposed analog readout system and $Re(Z_{reader})$ acquired with the VNA have the same shape, but inverted, with a shift caused by coils differences used on home-made and reference board. The frequency differential between the two systems exhibits a ~ 20 kHz deviation (see Fig.3), leading to a relative precision of 0.1 mmHg with the final sensor theoretical characteristics (200 kHz / mmHg).

Custom coils with a constant inductance across the specified frequency range must be fabricated to facilitate the testing

of absolute value measurements, as opposed to relative measurements. Additionally, enhancements in the VCO frequency range, while keeping a high step resolution, are required to achieve an expanded ICP measurement. Integration on a low-power micro-controller that uses energy harvesting and low energy data transmission protocol are the next steps for this system.

B. Coil optimization

Current works are done on planar coil design optimization, in order to test our system with different k and Q factors, between multiple medium to simulate skull and skin permittivity.

Self resonant frequency (SRF) of sensor and reader prototype coils needs to be as high as possible, while complying within our constraints of size and manufacturing process. As f_{res} is a function of L and C , minimizing variations in L across the frequency range is essential for precise ICP retrieval.

IV. CONCLUSION

Low power embedded readout sensor for low coupling capacitive sensor used in biomedical applications allows patients to have more freedom during healing process. Our system, coupled with optimized data transmission and energy harvesting, could achieve this, leading to further research on brain injuries and regulatory mechanisms. Further work will be done on a more compact integration (flexible substrate, ASIC, etc.) and better reliability under low coupling conditions.

REFERENCES

- [1] R. M. Chesnut *et al.*, "A Trial of Intracranial-Pressure Monitoring in Traumatic Brain Injury," *New England Journal of Medicine*, vol. 367, no. 26, pp. 2471–2481, Dec. 2012.
- [2] D. S. Nag *et al.*, "Intracranial pressure monitoring: Gold standard and recent innovations," *World Journal of Clinical Cases*, vol. 7, no. 13, pp. 1535–1553, Jul. 2019.
- [3] P. H. Raboel *et al.*, "Intracranial Pressure Monitoring: Invasive versus Non-Invasive Methods—A Review," *Critical Care Research and Practice*, vol. 2012, pp. 1–14, 2012.
- [4] M. Khan *et al.*, "Noninvasive monitoring intracranial pressure – A review of available modalities," *Surgical Neurology International*, vol. 8, no. 1, p. 51, 2017.
- [5] M. Nowak *et al.*, "Sensitivity analysis of a passive inductive telemetry system for a capacitive sensor," in *2006 Ph.D. Research in Microelectronics and Electronics*, Jun. 2006, pp. 273–276.
- [6] M. Simić *et al.*, "A Portable Device for Passive LC Sensors Readout With Low-Coupling Enhanced Sensitivity," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–12, 2023.
- [7] S. Roy *et al.*, "Low-Cost Portable Readout System Design for Inductively Coupled Resonant Sensors," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2022.
- [8] B. Lin *et al.*, "Temperature and Pressure Composite Measurement System Based on Wireless Passive LC Sensor," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, 2021.
- [9] M. Demori *et al.*, "Interrogation Techniques and Interface Circuits for Coil-Coupled Passive Sensors," *Micromachines*, vol. 9, no. 9, p. 449, Sep. 2018.
- [10] M. Masud *et al.*, "Measurement Techniques and Challenges of Wireless LC Resonant Sensors: A Review," *IEEE Access*, vol. 11, pp. 95 235–95 252, 2023.
- [11] J. Coosemans *et al.*, "A readout circuit for an intra-ocular pressure sensor," *Sensors and Actuators A: Physical*, vol. 110, no. 1-3, pp. 432–438, Feb. 2004.
- [12] F. Wang *et al.*, "A Novel Intracranial Pressure Readout Circuit for Passive Wireless LC Sensor," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 11, no. 5, pp. 1123–1132, Oct. 2017.

A 20.6-24.9 TOPS/W Multiplier-Less Digital In-Memory Computing Macro with low-cost Multi-Layer Inference in 28nm FDSOI for edge AI.

Antoine Gautier, Benoît Larras, Antoine Frappé

*Univ. Lille, CNRS, Centrale Lille, Junia, Univ. Polytechnique Hauts-de-France, IEMN UMR8520
antoine.gautier@junia.com

Abstract—In this paper, a low-power multiplier-less artificial neural network implementation is proposed. Reduced area and power consumption are achieved through the combination of a multiplier-less neuron unit, the use of a digital in-memory computing (DIMC) architecture of the neuron, and the implementation of a multi-layer network as a single iterative layer scheme. A $32 \times 32 \times 32 \times 32$ (4 layers of 32 neurons) feedforward fully connected network (FNN) has been fabricated in 28nm FDSOI CMOS. Measurement results show up to 24.9 8-b TOPS/W performance for a 1-layer inference and 20.6 TOPS/W for a 4-layer inference as well as a 1.2 8-b TOPS/mm².

I. INTRODUCTION

The implementation of Artificial Intelligence (AI) solutions is a global trend in the development of biomedical devices and systems or Intelligence of Things (AIoT) systems, where numerous devices and data communications are involved. Artificial Neural Networks (ANN) and their many flavors are a popular and well-covered AI solution in this context. However, while the development of the performance and complexity of ANNs is growing at a high rate, there is a need for solutions to reduce the on-chip hardware implementation cost [1]. One major trend is the use of an in-memory computing architecture that allows high TOPS/W energy performance by limiting the movement of memory data [2]. The objective of this work is to integrate a low-power artificial neural network processor on-chip for long-term health monitoring applications. To achieve this, a digital in-memory computing architecture is proposed, combined with an iterative layer scheme for multi-layer inference. A $32 \times 32 \times 32 \times 32$ (4 layers of 32 neurons) fully connected feedforward network (FNN) has been designed and fabricated in 28nm FDSOI CMOS.

II. IN-MEMORY COMPUTING SCHEME

In a conventional Von-Neumann architecture most of the power consumption is implied by the memory data movements: the data are fetched from the memory array and provided to the processing elements [3], which is time and power consuming. In a compute-in-memory architecture (CIM), each memory point has direct access to a processing element integrated within the memory array. CIM addresses the problem of weight movement with a static weight scheme allowing significant power reduction and faster operation due to the absence of fetch operation and weight movement. CIM can be implemented as analog schemes, which achieve higher energy

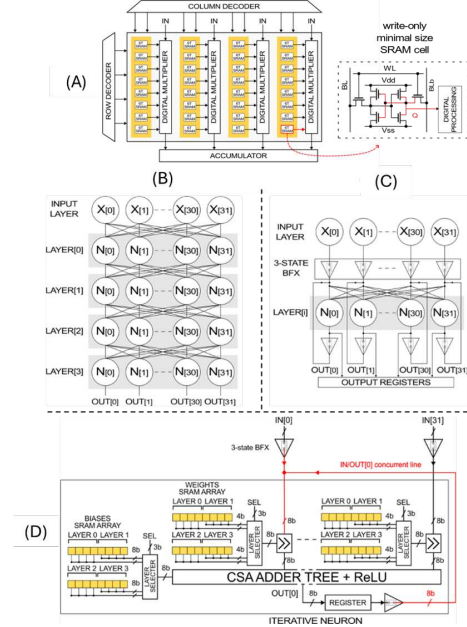


Fig. 1: (A) Proposed DIMC macro and write-only SRAM cell (B) Cascaded and (C) iterative layer scheme for multi-layer inference; (D) details of the iterative layer

efficiency but are prone to PVT variations and arduous design challenges [2], [3], or digital in-memory computing schemes [4], [5], [6] which are less energy efficient but more resilient to PVT variations.

In this paper, we propose a DIMC neuron macro with 32 8-bit inputs implemented as a SHIFT+ADD processing architecture. Input activations are fed in parallel to the macro allowing higher operating frequency and throughput at the cost of an increased area. The main contributor to complexity is the weight multiplication. A solution is to use non-uniform logarithmic power-of-two quantization of the weights [7], allowing the replacement of multipliers by shifters as well as the reduction of the weights bit-width and the weight memory size, leading to a reduction in the complexity of the neuron implementation cost. The ANNs weights are quantized using a single POT scheme on 8-bit resolution. Weights-activations products are then fed to a carry-save adder tree to process the final accumulated value in a single clock cycle. Minimal-size SRAM cells are implemented as write-only cells where the stored values of the weights are directly connected to

the processing elements integrated as close as possible to the SRAM cells. The proposed DIMC macro is illustrated in Fig. 1.

III. ITERATIVE LAYER APPROACH

Most of FNN application cases necessitate multi-layer inference. Implementing all the layers on-chip as shown in Fig. 1 (B) is area and leakage inefficient. An alternative is to implement only one layer and reload the memory weights for each layer after each computation, undermining the benefits of the in-memory computing scheme by re-introducing costly data movements, thus when computing multiple layer the performance of such solution would degrade as shown in Fig. 2 (A). We propose to overtake this problem by sharing each processing element with 4 different memory points related to 4 different layers, and to provide the ability to sequentially compute the 4 layers without any memory movement, saving both area and power, as illustrated by Fig. 1 (C) and (D). More precisely, the iterative layer scheme is based on tri-state logic buffers on concurrent input/output lines. During the first layer inference, input tri-state buffers connect the activation inputs to the neurons. After inference, the neuron's output results are stored in a buffer register. The output value of the first layer is connected to the next layer, by activating the tri-state buffers and deactivating the input tri-state buffers. Area overhead of proposed multi-layer scheme degrades TOPS/mm² by 15 %.

IV. MEASUREMENT RESULTS AND SOTA COMPARISON

Measurement results of the fabricated integrated circuits Fig. 2 (A) highlights the inference performance in both area and energy efficiency for single and multi layer inference. The best performance for 1-layer inference is measured for a clock frequency of 150MHz and a VDD of 0.6V with 24.97 TOPS/W and 1.17 TOPS/mm². In the same conditions performing successive layers degrades lightly the TOPS/W performance by 20% for computing up to 4 successive layers, downgrading to 20.6 TOPS/W, as shown in Fig. 2 (B). A TOPS/W choropleth scale Schmo plot of the performed measurements is provided in Fig. 2 (C). Comparison with state-of-the-art has been made by gathering multiple DIMC references between 16nm-55nm. All DIMC macros from literature provide performances for 1-layer inference, to estimate their actual performance when dealing with multiple layers, we considered the case that SOTA are implemented as a single macro and it is needed to update iteratively the weights layer after layer and we recalculated both TOPS/W and TOPS/mm² by taking into account the memory access cost leading to a degradation of SOTA performance. Optimum measured results in 8-bit TOPS/W and 8-bit TOPS/mm² of the fabricated chips for both 1-layer and 4-layer inference are shown in Fig. 2 (D) along the 1-layer and estimated 4-layer performances of 9 DIMC SOTA references from literature. While the 1-layer TOPS/W performance reaches the numbers of similar SOTA references, the solution becomes competitive when dealing with multi-layer networks, which are needed for most applications, and even outperforms SOTA for 4-layer inference.

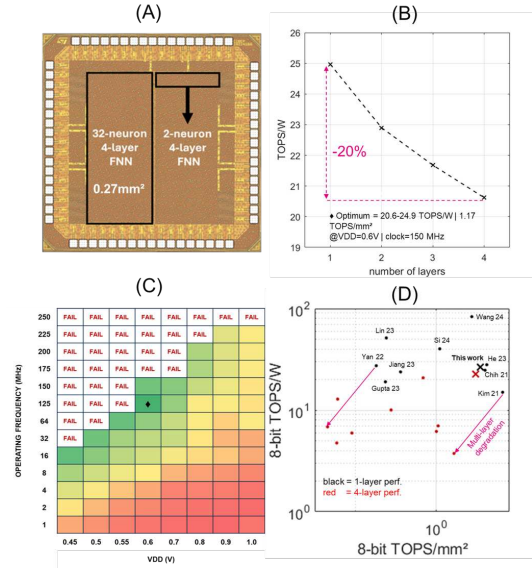


Fig. 2: (A) Cascaded and (B) iterative layer scheme for multi-layer inference; (C) details of the iterative layer computation

V. CONCLUSION

In this paper, the combination of POT quantization and the use of an iterative single layer in-memory architecture are shown as relevant to reduce neural network implementation cost while keeping high inference performance. A FNN network of dimensions 32x32x32x32 has been designed in 28nm FDSOI and integrated on-chip. Proposed DIMC macro performances are 24.9 TOPS/W inference energy efficiency which reach the levels of similar CIM neuron macros from the literature while providing flexibility on the network dimensions as well as the possibility to compute multi-layer networks without major degradation of the energy cost with a reduction of only 20% of the TOPS/W when computing up to 4 successive layers and a reduction of 15% of the TOPS/mm² outperforming SOTA for multi-layer applications.

REFERENCES

- [1] Mostafa Rahimi Azghadi et al., "Hardware Implementation of Deep Network Accelerators Towards Healthcare and Biomedical Applications", IEEE Transactions on Biomedical Circuits and Systems, 2020
- [2] Jae-sun Seo et al., "Digital Versus Analog Artificial Intelligence Accelerators: Advances, trends, and emerging designs", IEEE Solid-State Circuits Magazine Volume: 14, Issue: 3, Summer 2022
- [3] N. Verma et al., "In-Memory Computing: Advances and Prospects," in IEEE Solid-State Circuits Magazine, vol. 11, no.3, Summer 2019
- [4] Yifan He and al., "A 28nm 38-to-102-TOPS/W 8b Multiply-Less Approximate Digital SRAM Compute-In-Memory Macro for Neural-Network Inference", ISSCC 2023
- [5] Y.-D. Chich et al., "An 89 TOPS/W and 16.3 TOPS/mm² All-Digital SRAM-Based Full-Precision Compute-In Memory Macro in 22nm for Machine-Learning Edge Applications", ISSCC 2021
- [6] B. Yan et al., "A 1.041-Mb/mm² 227.38-TOPS/W Signed-INT8 Dynamic-Logic-Based ADC-less SRAM Compute-in-Memory Macro in 28nm with Reconfigurable Bitwise Operation for AI and Embedded Applications", ISSCC 2022
- [7] D. Przewlocka-Rus et al., "Powers-of-Two Quantization for Low Bitwidth and Hardware Compliant Neural Networks", tinyML Research Symposium, 2022

Reliability Under Stress: The Impact of Localized Aging on RO-PUF Architectures in FPGAs

Aghiles Douadi^{*†}, Elena-Ioana Vatajelu^{*}, Paolo Maistri^{*}, Vincent Beroulle[†], Giorgio Di Natale^{*}

^{*}Univ. Grenoble Alpes, CNRS, Grenoble INP¹, TIMA, 38000 Grenoble, France

[†]Univ. Grenoble Alpes, Grenoble INP¹, LCIS, 26000 Valence, France

Abstract—Physical Unclonable Functions (PUFs) have emerged as a promising security solution for electronic devices, but their reliability remains a critical challenge. This paper investigates the impact of localized aging on the reliability of Ring Oscillator-based PUFs (RO-PUFs) implemented on FPGAs. Using a simple and reproducible aging method, we subjected multiple FPGAs to controlled heating experiments, targeting regions with active and inactive Ring Oscillators rather than genuine RO-PUF configurations. Over 11 weeks of heating, significant and distinguishable aging effects were observed, leading to instability levels up to two times higher than normal in the RO-PUFs.

I. INTRODUCTION

The emergence of advanced physical attacks, such as X-ray [1], electromagnetic (EM) [2] [3], and thermal attacks, has significantly increased the vulnerability of electronic devices necessitating the use of more secure and extremely costly memory systems to store critical information. One of the most concerning threats is the possibility of cloning or replicating a device for malicious purposes. To address this growing threat, a new solution has emerged: Physical Unclonable Functions (PUFs). PUFs can be integrated into any circuit and provide a unique identification key for each chip without requiring stored information. This is achieved by exploiting manufacturing process variations to generate the key [4]. However, while PUFs are considered unclonable, they are not without vulnerabilities. Environmental factors such as extreme temperatures, electronic noise, power supply variations, and aging can negatively impact their reliability. To mitigate these issues, PUFs often incorporate embedded solutions to enhance stability. One common countermeasure involves filtering out unreliable bits during the enrollment phase. This paper demonstrates that, with minimal resources, it is possible to induce sufficient aging in a circuit to render these countermeasures ineffective. Specifically, we conduct a thermal attack on an FPGA where a RO-PUF is implemented. Our attack combines two bitstreams: a trusted one, containing the original PUF, and a malicious one, integrating a heating module and a replica of the PUF. We show that localized aging can be induced by manipulating the state of the ROs in the malicious bitstream. The Section II details the experimental setup, including the FPGA and attack method. Section III presents experimental observations. Section IV summarizes the results.

¹Institute of Engineering Univ. Grenoble Alpes

This work was supported by a research grant from the French Agence Nationale de la Recherche (POP project, ANR-21-CE39-0004).

II. EXPERIMENTAL SETUP

A. FPGA and RO-PUF

In this study, a 256-bit RO-PUF is implemented, where each RO comprises 8 stages: a NAND gate, 6 inversion stages, and a buffer for output. In an FPGA, traditional inverters are replaced by LUTs (Look-Up Tables) configured to perform inversion and NAND operations. The frequencies are measured using internal 31-bit counters, with measurements taken over 0.01-second intervals. The state machine that manages communication transmits 64 bits: one bit is used for the comparison result between the two ROs, while the remaining bits carry the counter values of the two ROs being compared. The 256 bits of the RO-PUF are generated by comparing two interdigitated groups of 256 ROs each. Each oscillator in the first group is compared with its counterpart in the second group, ensuring that comparisons are made between nearby ROs.

B. Attack Model

In our attack model, two bitstreams are used: one for the original PUF and another, the malicious bitstream, which includes 4,600 SIROs and a replica of the original PUF. In the replica, group 1 ROs oscillate normally, while group 2 ROs (red) are blocked. This approach, inspired by [5], is compatible with all FPGA architectures. Unlike prior work focused on cloning, our goal is to assess the PUF's reliability and the effectiveness of filtering techniques under thermal attack. For the experiment, three FPGAs from the same family were used: two as reference devices and one subjected to the attack. The targeted FPGA was exposed to high temperatures for one week, while the reference FPGAs remained powered off in the same conditions. Frequencies and responses were recorded weekly, once the attacked FPGA returned to its normal temperature, allowing for the observation of thermal effects over time (Figure 1a).

III. RESULTS AND DISCUSSION

A. Evolution of the PUF frequencies

To improve clarity and highlight the impact of the attack on RO frequencies, we chose to represent the average RO frequencies across the three FPGAs. This approach simplifies the presentation by using averages, making the analysis more accessible while preserving the data from all FPGAs. In Figure 1b, the reference FPGAs, which were not attacked, show stable frequencies. In contrast, the attacked FPGA exhibits

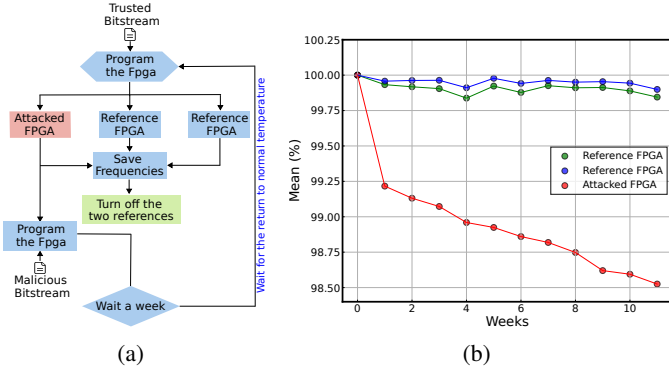


Fig. 1: (a) The methodology involves subjecting one FPGA to a malicious bitstream attack, while using two other FPGAs as references. Frequencies are measured weekly. (b) Shows the evolution of the normalized mean frequencies of the RO-PUF across the three FPGAs after each week of testing.

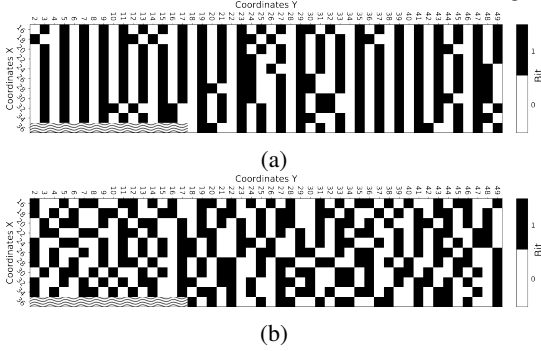


Fig. 2: Value of 1 is assigned if the $|\Delta F|$ of a RO is greater than that of the one it is compared to, and a value of 0 is assigned otherwise, shown for the attacked FPGA (a) and for one of the reference FPGAs (b).

a significant decrease in frequencies, reaching up to 1.5% lower than the initial measurement. The same figure also shows a slight drop in frequencies for the reference FPGAs between weeks 4 and 6. These variations, likely due to room temperature changes.

B. Impact on the RO-PUF reliability

The two groups of ROs in the malicious bitstream were programmed differently (as explained in section II-B). Group 1 continued oscillating during the entire attack period, while the ROs in group 2 were frozen and did not oscillate. This difference in behavior results in non-uniform aging across the ROs. Additionally, the temperature generated by the SIROs is not uniformly distributed, emphasizing the crucial role of the ROs' placement in the system's overall behavior. A binary heatmap was generated. The principle is as follows: for each pair of ROs, if the frequency decrease of the oscillating RO (e.g., X16Y49) is greater than that of the static RO (X16Y48), the RO is classified as 1; otherwise, it is classified as 0. This comparison process is applied to all pairs of ROs, allowing the differences in aging to be visualized in the form of a binary map. In Figure 2a, corresponding to the attacked FPGA, distinct lines are visible, indicating areas of the circuit that have undergone more significant aging. These lines mark regions where the ROs remained active during the attack phase, showing greater degradation, while areas with less activity are less affected. In contrast, Figure 2b, representing one of the reference FPGAs, shows a uniform distribution of bits

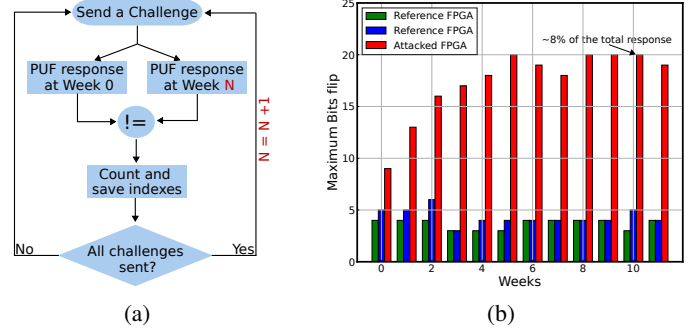


Fig. 3: (a) Methodology used to calculate the number of bit flips and their corresponding indexes, using the initial measurement from week 0 as a reference. (b) Maximum number of bit flips recorded over the testing weeks for the three FPGAs.

without any distinct lines. This clearly illustrates that the aging effect is mostly localized, although not perfectly so. The slight inconsistencies observed are likely due to the relatively short aging period of 11 weeks. While 11 weeks is significant in an accelerated environment, natural aging processes typically span years, and while high temperatures speed up this process, complete and uniform aging still requires more time. This localized aging can significantly impact the reliability of our RO-PUF. To verify this, we applied the methodology outlined in Figure 3a to track the number of bit changes in the PUF responses after each week of testing across the three FPGAs used in the experiments. As shown in Figure 3b, the bit flip rate is monitored over time. Initially, the attacked FPGA shows a bit flip rate of 3%, but after the thermal attack, this rate more than doubles.

IV. CONCLUSION

In this article, we demonstrated that exposing an FPGA to high temperatures, combined with the strategic use of internal resources (e.g., oscillating vs. non-oscillating structures), leads to targeted aging, altering the chip's intrinsic characteristics. While previous literature suggests that inactive oscillators age faster than active ones, our experiments revealed the opposite. This discrepancy can be attributed to the architecture of FPGAs, where limited knowledge of the internal structure of LUTs hinders definitive conclusions.

REFERENCES

- [1] Laurent Maingault and al. Laboratory x-rays operando single bit attacks on flash memory cells. In *International Conference on Smart Card Research and Advanced Applications*, pages 139–150. Springer, 2021.
- [2] Mathieu Dumont and al. Modeling and simulating electromagnetic fault injection. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 40(4):680–693, 2020.
- [3] Alexandre Proulx and al. Investigating the effect of electromagnetic fault injections on the configuration memory of sram-based fpga devices. In *2023 IEEE Physical Assurance and Inspection of Electronics (PAINE)*, pages 1–7, 2023.
- [4] Alireza Shamsoshoara and al. A survey on physical unclonable function (puf)-based security solutions for internet of things. *Computer Networks*, 183:107593, 2020.
- [5] Hayden Cook and al. Cloning the unclonable: Physically cloning an fpga ring-oscillator puf. In *2022 International Conference on Field-Programmable Technology (ICFPT)*, pages 1–10. IEEE, 2022.

Une solution par Lockstep pour une architecture RISC-V sur FPGA résistante aux radiations

H. Closquinet^{1,3}, F. Miller¹, L. Noizette¹, Y. Helen², P. Girard³, T. Vayssade³, A. Virazel³

¹ Nuclétudes, Les-Usis, France

² Direction Générale de l'Armement, Bruz, France

³ LIRMM – Univ. de Montpellier / CNRS, Montpellier, France

Abstract— Cet article présente une architecture RISC-V qui met en œuvre de la redondance par Lockstep. Cette méthode garantit une détection et une correction efficace des fautes dans des environnements radiatifs sévères tout en optimisant l'utilisation des ressources. Une injection de fautes par émulation a permis de valider cette approche.

Mots Clés— Lockstep, tolérance aux fautes, injection de fautes, processeurs RISC-V, SEE.

I. INTRODUCTION

Dans les systèmes électroniques modernes, de plus en plus de composants Commerciaux Off-The-Shelf (COTS) sont utilisés dans des environnements exposés aux radiations. Malgré leur coût avantageux, ces systèmes sont particulièrement vulnérables aux erreurs induites par les radiations, et notamment aux effets d'événements uniques, les Single Event Effects (SEE). Si des solutions comme la redondance matérielle ou les Error Corrector Code (ECC) permettent d'y répondre, elles ont un impact sur les ressources et les performances du système.

Pour limiter leur impact, le Lockstep est un bon compromis entre efficacité de protection, performance du système et ressources utilisées. Il consiste à exécuter le même code sur deux cœurs processeurs en parallèle (pouvant fonctionner en décalage de plusieurs coups d'horloge) et compare les sorties des cœurs processeurs afin de détecter des erreurs. Il intègre également des mécanismes logiciels de Checkpoint/Rollback visant à restaurer un état opérationnel en cas de faute. Cependant, les travaux autour du Lockstep, comme [1], [2] et [3], incluent une utilisation plus importante des ressources, une complexité et une consommation d'énergie accrue.

Ce travail propose une architecture de Lockstep innovante implémentée sur FPGA, combinant détection et correction de fautes avec une utilisation minimale des ressources. Cette architecture utilise deux cœurs RISC-V avec un décalage de deux cycles et des mécanismes de Checkpoint/Rollback. L'approche proposée a été validée par injection de fautes par émulation.

Ce papier est structuré comme suit : la Section II présente l'approche de Lockstep proposée, tandis que la Section III décrit les campagnes de tests menées et l'analyse des résultats.

II. APPROCHE PROPOSEE

Le module Lockstep est composé de plusieurs sous-modules VHDL, comme le montre la figure 1. Malgré un décalage de deux cycles entre les cœurs, deux buffers garantissent un stockage temporaire des signaux AHB des processeurs sur deux cycles, ce qui permet la synchronisation des données. En parallèle, les requêtes du cœur "maître" continuent de communiquer avec le reste de l'architecture,

afin de ne pas affecter les performances du système. En sortie des deux buffers, une Machine à Etats Finis (FSM) compare les requêtes des deux cœurs. Si elles sont identiques, un signal d'interruption sauvegarde périodiquement, toutes les 15ms, l'ensemble des General Purpose Registers (GPR), certains Control and Status Registers (CSR) et 2 KB de pile des applicatifs. En cas de faute, une interruption tente de restaurer le dernier état fonctionnel valide. Si l'erreur persiste, un sous-module isole les cœurs pour éviter la propagation de la faute. Un dernier sous-module, un buffer, gère le retour des données vers les cœurs tout en maintenant le décalage de deux cycles.

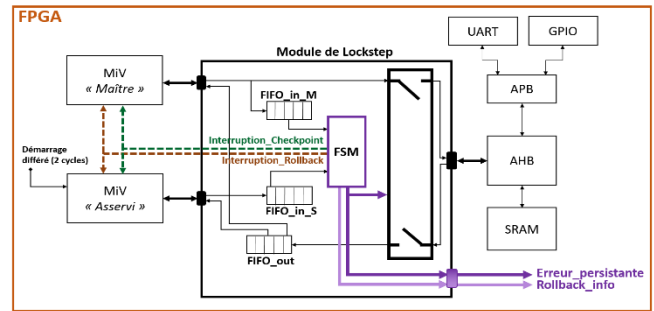


Fig. 1. Architecture en Lockstep avec le détail de ses sous-modules

Cette étude compare deux architectures RISC-V, en Lockstep et classique, toutes les deux implémentées sur un FPGA PolarFire de Microchip [4]. Les ressources utilisées par chaque architecture sont présentées dans le Tableau I.

TABLEAU I
UTILISATION DES RESSOURCES LOGIQUES DES DEUX ARCHITECTURES

	Processeur(s)		Architecture		Ressources FPGA		
	4LUT	DFF	4LUT	DFF			
Architecture RISC-V classique	8 014 (2,68%)	3 156 (1,03%)	20 092 (6,71%)	13 364 (4,46%)	299 544 (100%)		
Architecture RISC-V en Lockstep	8 175 Cœur 0 (2,73%)	7 033 Cœur 1 (2,33%)	3 173 Cœur 0 (1,06%)	2 496 Cœur 1 (0,83%)	27 647 (9,23%)	16 177 (5,40%)	299 544 (100%)
Surcoût en ressources	+ 161 Cœur 0 (0,03%)	+ 981 Cœur 1 (0,33%)	+ 17 Cœur 0 (~0%)	+ 660 Cœur 1 (0,22%)	+ 7 553 (37,60%)	+ 2 813 (21,64%)	-

Trois applicatifs de la suite MiBench [5] ont été utilisés avec deux niveaux d'optimisation : o0 (utilisation limitée des registres) et o3 (utilisation complète des registres). QSort, trie les données, Bitcount effectue des opérations de comptage de bits et MxM exécute des opérations de calcul matriciel. Les performances des deux architectures sont résumées dans le Tableau II.

TABLEAU II
PERFORMANCES RISC-V: CLASSIQUE VS LOCKSTEP

Empreinte d'exécution par benchmark en ms :	optimisation de la compilation	Bitcount	MM	QSort
Architecture RISC-V classique (sans Checkpoint)	o0	73,8	79,2	253,9
	o3	24,9	24,6	91,8
Architecture RISC-V en Lockstep (avec Checkpoint)	o0	114,6 (+ 55,2%)	101,2 (+ 27,8%)	392,5 (+ 54,6%)
	o3	41,8 (+ 67,9%)	26,2 (+ 7,4%)	147,6 (+ 60,8%)
Nombre de Checkpoint(s) (toutes les 15 ms)	o0	7	6	26
	o3	2	1	9
Empreinte d'exécution en ms pour 1 Checkpoint (stockage : GPR + CSR + 2 KB de pile)	o0	4,1	1,6	4,4
	o3	3,8	1,3	4,1

III. EVALUATION DE CAMPAGNES D'INJECTIONS DE FAUTES

Pour évaluer l'approche de Lockstep, notre groupe a développé une méthode d'injection de fautes par émulation, décrite dans [6]. Elle utilise SmartDebug, un outil de débogage fourni par Libero, l'environnement de conception pour les FPGAs PolarFire. Les campagnes d'injection de fautes réalisées ont ciblé les D flip-flops (DFFs) et les GPRs du cœur MiV "maître", avec 5000 fautes injectées dans les DFFs et 3000 dans les GPRs basés sur un calcul statistique d'injection de fautes de la référence [7]. Plusieurs campagnes ont été menées sur les applicatifs (QSort, Bitcount, MxM) et les deux niveaux d'optimisation (o0 et o3), avec des résultats classifiés en quatre catégories. Les fautes **UNACE** (**UnNecessary for Architecturally Correct Execution**) sont sans impact ou masquées, les fautes de type **Reset** nécessitent une réinitialisation pour ramener le système à un état stable, les **Crashes** rendent le système non fonctionnel, et les **Fautes Corrigées** sont récupérées par le rollback du Lockstep.

Les résultats montrent que la méthode Lockstep a atteint un taux de détection compris entre 99,5% et 100%. Pour l'applicatif Bitcount en o0, le Lockstep corrige 22,8% des erreurs DFFs et 58,9% des erreurs GPRs. Les taux de crashes ont diminué de 21,6% pour les DFFs et de 41,4% pour les GPRs, comme vu en Fig. 2. Des résultats similaires sont observés pour les applicatifs MxM en Fig. 3 et QSort en Fig. 4. MxM corrige 24,8% des DFFs et 66,9% des GPRs avec une réduction des crashes de 15,7% et 88,5% respectivement, pour QSort 21,8% des DFFs et 53,7% des GPRs sont corrigés avec une réduction des crashes de 36,7% et 51,6%.

Sous l'optimisation o3, le taux de crashes est plus élevé, cela est dû à une utilisation plus importante des registres, mais le mécanisme de correction du Lockstep fonctionne correctement. Les taux de correction sont les suivants : Bitcount : 20,8% (DFF), 52% (GPR), MxM : 18,1% (DFF), 52,4% (GPR), QSort : 19,6% (DFF), 47,1% (GPR).

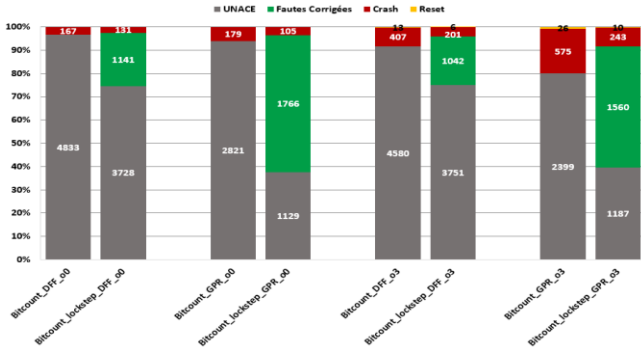


Fig. 2. Résultats de l'injection de fautes pour Bitcount : 5000 fautes injectées dans les DFFs et 3000 dans les GPRs.

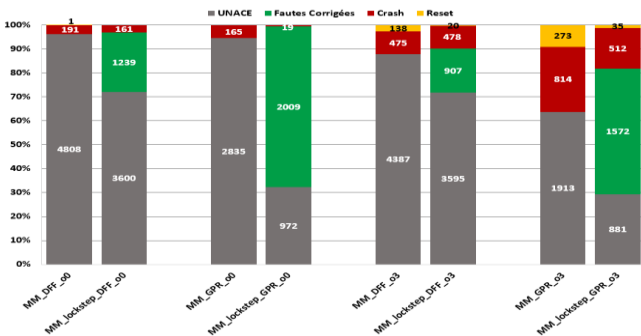


Fig. 3. Résultats de l'injection de fautes pour MxM : 5000 fautes injectées dans les DFFs et 3000 dans les GPRs.

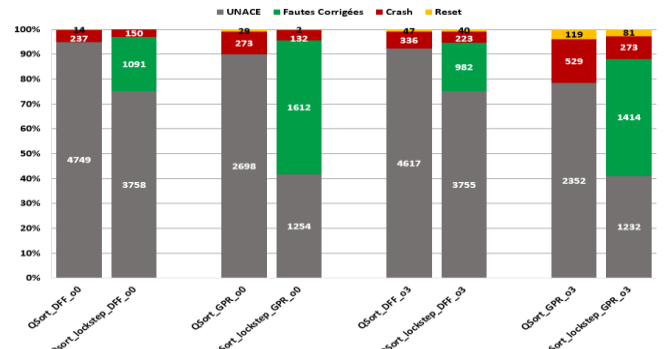


Fig. 4. Résultats de l'injection de fautes pour QSort : 5000 fautes injectées dans les DFFs et 3000 dans les GPRs.

IV. CONCLUSION

Cet article présente une technique de Lockstep mise en oeuvre sur une architecture RISC-V et implémentée sur un FPGA PolarFire. Les résultats des injections de fautes menées ont montré que la méthode proposée détecte entre 99,5% et 100% des fautes. En termes de correction, le Lockstep se montre efficace, avec jusqu'à 69% des fautes corrigées en o0 et 52,4% en o3. L'utilisation de ressources supplémentaires pour l'ajout de la méthode de Lockstep se limite à 37,6% en 4LUT et 21,04% en DFFs, tandis que l'impact sur les performances du système représente 55 % pour les applicatifs en o0 et 60 % pour les applicatifs en o3.

Les travaux futurs étendront cette méthodologie par Lockstep sur d'autres systèmes intégrés complexes tels que les SoC-FPGAs. Cela ouvre la voie à l'adaptation du Lockstep pour des applications de Calcul à Haute Performance (HPC), afin de répondre aux exigences croissantes de systèmes embarquant des calculs de plus en plus puissants.

ACKNOWLEDGMENT

Ce travail a été financé par l'Agence de l'Innovation de Défense (AID) dans le cadre d'une thèse de doctorat.

REFERENCES

- [1] S. Kasap, et al., "Novel Lockstep-based Approach with Roll-back and Roll-forward Recovery to Mitigate Radiation-Induced Soft Errors," *IEEE Nordic Circuits and Systems Conference*, Oslo, Norway, 2020.
- [2] P. M. Aviles, et al., "Supervised Triple Macrosynchronized Lockstep (STMLS) Architecture for Multicore Processors," *IEEE Access*, 2023.
- [3] I. Marques, C. Rodrigues, A. Tavares, S. Pinto, T. Gomes, "Lock-V: A heterogeneous fault tolerance architecture based on Arm and RISC-V," *Microelectronics Reliability*, Volume 120, 2021.
- [4] Microchip Technology Inc, Available: <https://www.microchip.com/en-us/products/fpgas-and-plds/fpgas/polarfire-fpgas/polarfire-mid-range-fpgas>, Accessed on: April. 9, 2025.
- [5] M. R. Guthaus, et al., "MiBench: A free, commercially representative embedded benchmark suite," *IEEE International Workshop on Workload Characterization*, Austin, USA, pp. 3-14, 2001.
- [6] L. Noizette, et al., "Understanding the Link Between Complex Digital Devices Soft Error Rate and the Running Software," *IEEE Transactions on Nuclear Science*, vol. 70, no. 8, pp. 1747-1754, Aug. 2023.
- [7] R. Leveugle, et al., "Statistical fault injection: Quantified error and confidence", *Proc. Design Autom. Test Eur. Conf. Exhib.*, pp. 502-506, Apr. 2009.

Towards Automated Hardware Generation for Spiking Neural Networks: A Modular Design Flow Approach

Xindan Zhang

LIP6, Sorbonne Université, CNRS

Paris, France

Email: xindan.zhang@lip6.fr

Abstract—Neuromorphic computing takes inspiration from how biological neural systems operate, with the goal of building systems that process information using low-power and discrete spikes [1]. In recent years, several tools have been developed to help convert spiking neural network (SNN) models into hardware designs. This PhD project, which began in February 2025, focuses on learning from existing solutions and gradually shaping a workflow that supports experimentation with different hardware configurations, with particular attention to reliability aspects. As a first step, we implemented a simple LIF-based SNN on FPGA.

I. INTRODUCTION

Neuromorphic computing is a growing field that tries to build systems inspired by how the brain works. In particular, SNNs are interesting because they can use less energy and work with spikes over time instead of continuous signals.

Tools like ModNEF [4] (Université de Lille) and Qualia [5] (Université Côte d’Azur) can already convert SNN models into hardware. This PhD project began recently. Our initial goal is to better understand how such frameworks operate, what constraints they have, and whether a more adaptable design flow could be developed. The long-term perspective is to gradually build a design flow that is easier to adapt and more suitable for exploring different hardware configurations.

The rest of this paper briefly introduces SNNs and their specific features, presents related tools, and describes the current progress and directions for this work.

II. SPIKING NEURAL NETWORKS AND THEIR ADVANTAGES

SNNs are a type of neural network that uses spikes to send information, more like how real neurons work. Instead of using numbers that change smoothly, they use simple signals (spikes) that happen at certain times.

Compared to traditional artificial neural networks (ANNs), SNNs can be more energy-efficient, because the neurons exhibit sparse activity due to event-driven computation. This is useful for devices that run on batteries or have limited computing power, particularly in embedded and edge computing scenarios. SNNs are also good for working with time-based data, like gestures or sounds.

However, training SNNs is more difficult than training regular neural networks. Also, building efficient hardware for them is not easy. That’s why researchers are trying to create

tools that can help design and test SNN-based systems more easily.

III. RELATED WORKS

In the past years, several hardware platforms have been developed to support the implementation and experimentation of SNNs. Some of them are designed specifically to run large-scale SNNs efficiently and have been widely used in neuromorphic research projects.

Loihi [2], developed by Intel, is a digital neuromorphic chip that supports on-chip learning and asynchronous SNNs. It features programmable neuron models, event-driven computation, and dynamic sparse connectivity, enabling efficient implementations for robotics, edge AI, and autonomous systems. While the hardware is not open-source, Loihi provides a full-stack development environment with APIs and simulation tools (including the Lava framework), making it a widely adopted platform in neuromorphic research.

SpiNNaker [3], designed by the University of Manchester, is a massively parallel neuromorphic system built with ARM cores interconnected via a packet-switched network. It simulates large-scale neural networks in real time, prioritizing software flexibility and biological plausibility. SpiNNaker has been used in cognitive modeling, neuroscience studies, and projects like the Human Brain Project.

Other academic efforts have focused on lightweight and re-configurable toolchains. For example, ModNEF [4] generates VHDL code from configurable SNN architectures, supporting various neuron models. Qualia [5] is a flexible framework for training, quantizing, and deploying neural networks on embedded targets. It is typically used with models from SpikingJelly [6], and supports deployment on microcontrollers such as STM32L4 as well as the custom neuromorphic hardware platform SPLEAT [7], a low-power accelerator for event-based image classification in satellite applications.

IV. INITIAL WORK ON SNN HARDWARE DEPLOYMENT

To explore the full workflow from training to deployment, we built a simple SNN and implemented it on FPGA (ZCU104). The model has two fully connected layers (784-64-10) with leaky integrate-and-fire (LIF) neurons and uses 10 time steps for temporal processing. It was trained on

the MNIST dataset using the *snnTorch* [8] framework, and achieved about 95% accuracy on the test set.

Quantization was performed using quantization-aware training (QAT), enabling 8-bit integer representation of weights. By avoiding floating-point operations, we reduced memory and computational requirements, making it easier to implement both the weight storage and processing logic within the BRAM and DSP resources available on the ZCU104 FPGA. After synthesis, the quantized version reduced BRAM and DSP usage by over 50% compared to the original floating-point model, and also significantly lowered flip-flop and LUT counts. This optimization came with a minor drop in accuracy (around 2%) when running on hardware.

We rewrote the model in C++, including matrix multiplication with quantized weights, spike-driven voltage updates, classification logic, and neuron dynamics with refractory periods. We also designed the structure to be synthesis-friendly. The weights were defined as static arrays and synthesized together with the logic. We applied pragmas such as 'ARRAY_PARTITION' for loop-level parallelism. Then the C++ model was compiled with Vitis HLS to generate hardware description (RTL). It was then packaged as a Vivado IP core and integrated into a Vivado block design targeting the ZCU104 board, based on Xilinx's Zynq UltraScale+ MPSoC platform.

To test the performance, we wrote a simple C program to send input data and read predictions. Despite this optimization, the classification accuracy remained close to the original model.

V. OBJECTIVES AND METHODOLOGY

Building on this initial work, the broader goal of the PhD is to explore how trained SNN models can be converted into efficient hardware architectures using a flexible and modular design flow.

We started by reviewing the main research in neuromorphic computing, SNN learning methods, and hardware accelerators. This helps us understand how different tools approach the problem, what models they support, and how their design flows are structured.

Beyond tool evaluation, we plan to experiment with architecture-level parameters, such as numerical precision, memory layout, and neuron models to observe their impact on performance, energy usage, and resource consumption on FPGA.

Later in the project, we plan to build a simple and modular design flow. The idea is to let users try out different architecture choices, such as changing numerical precision, memory organization, or network structure, and see how these changes affect performance, energy use, and hardware size.

We are also interested in exploring how hardware reliability and data protection can be considered during the design process. For instance, we want to look at simple strategies like adding redundancy or isolating sensitive components.

Ideally, the design flow should be compatible with multiple training frameworks and not depend on a single ecosystem.

We will try to keep it open and easy to reuse or extend, so that others can test new ideas without starting from zero.

VI. EXPECTED CONTRIBUTIONS

This work is still at an early stage, but we hope it will lead to:

- A simple and flexible design flow for turning SNN models into hardware;
- A better understanding of how design choices (like precision or memory) affect performance and resource usage;
- Some basic tools for testing different architecture options;
- Ideas on how to include reliability and security in neuromorphic designs.

We also plan to share parts of the code or scripts that could be useful for other researchers working on similar topics.

VII. CONCLUSION

This work represents the starting point of a doctoral research project that aims to explore the automated generation of hardware architectures for SNNs. We will develop a modular design flow that enables testing of different hardware architectures for SNNs, with particular attention to resource usage and reliability.

While several open questions remain, especially regarding the generalization of such a flow and its applicability across different training frameworks, we believe that this investigation will provide valuable insights into the co-design of neuromorphic models and their hardware realizations.

Next steps are refining the design flow, testing use cases, and evaluating trade-offs in cost, performance, and robustness.

REFERENCES

- [1] C. D. Schuman, S. R. Kulkarni, M. Parsa, J. P. Mitchell, P. Date, and B. Kay, "Opportunities for neuromorphic computing algorithms and applications," *Nature Computational Science*, vol. 2, no. 1, pp. 10–19, 2022. [Online]. Available: <https://doi.org/10.1038/s43588-021-00184-y>
- [2] M. Davies et al., "Loihi: A Neuromorphic Manycore Processor with On-Chip Learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018. doi: 10.1109/MM.2018.112130359
- [3] S. B. Furber et al., "Overview of the SpiNNaker System Architecture," *IEEE Trans. Comput.*, vol. 62, no. 12, pp. 2454–2467, 2013. doi: 10.1109/TC.2012.142
- [4] B. Miramond et al., "ModNEF project repository," Université de Lille, [Online]. Available: <https://gitlab.univ-lille.fr/bioinsp/ModNEF/-/blob/main>
- [5] N. Abderrahmane, B. Miramond, and E. Kervennic, "Focus to Learn More: Qualia, a Configurable and Open-Source Framework for the Design and Deployment of Neuromorphic Systems," *Sensors*, vol. 21, no. 9, p. 2984, 2021. [Online]. Available: <https://doi.org/10.3390/s21092984>
- [6] W. Fang et al., "SpikingJelly: An open-source machine learning infrastructure platform for spike-based intelligence," *Science Advances*, vol. 9, eadi1480, 2023. doi: 10.1126/sciadv.adi1480
- [7] N. Abderrahmane, B. Miramond, E. Kervennic, and A. Girard, "SPLEAT: SPiking Low-power Event-based ArchiTecture for in-orbit processing of satellite imagery," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2022, pp. 1–10. doi: 10.1109/IJCNN55064.2022.9892277
- [8] J. H. Eshraghian, "snnTorch: An Open-Source Library for Spiking Neural Networks in PyTorch," [Online]. Available: <https://snntorch.readthedocs.io/en/latest/>

Network Folding: A Resource-Efficient Approach for DNN Streaming Architectures

Van-Quan Pham, Adrien Prost-Boucle, Olivier Muller, Frédéric Pétrot

Laboratory TIMA/SLS, Grenoble INP

University of Grenoble Alpes

Grenoble, France

GivenName.Surname@univ-grenoble-alpes.fr

Abstract—Most Deep Neural Network (DNN) implementations on hardware focus on the inference stage for relatively light-weight models, primarily due to limitations in resources, memory capacity, energy consumption, and performance requirements. Streaming architectures, also known as dataflow architectures, significantly enhance performance—measured in Frames Per Second (FPS) by optimizing computation pipelines tailored to custom DNN models. However, as networks grow larger, the demand for computational resources increases substantially, making hardware implementation more challenging. By using a folding network architecture, we enable the deployment of large DNN models on smaller FPGAs with acceptable throughput degradation.

Index Terms—Deep Neural Network, Streaming Architecture, Network Folding, FPGA, resource constraints.

I. INTRODUCTION

The streaming architecture has demonstrated exceptionally high performance for feed-forward DNN models, which do not require frequent memory write-backs compared to systolic architecture. Numerous projects have been dedicated to optimizing streaming architectures, including: NNAWAQ [1], FINN [2], NeuroCorgi [3], fpgaConvNet [4], HLS4ML [5], H2PIPE [6]. However, for large-scale networks, the significant resource constraints associated with designing and implementing a vast number of parameters and operations remain a challenge for researchers. To address this issue, folding techniques have been explored to efficiently rescale the architecture, including: fpgaConvNet [4] that using Design Space Exploration (DSE) to design and reconfigure a FPGA or using multiple FPGAs, FINN [2] and NNAWAQ [1] that manipulate the granularity of input and output folding—using a SIMD and PE-based approach in FINN, and a combination of parallelism and time-multiplexing in NNAWAQ.

In various DNN models, many layer blocks are repeatedly executed multiple times, as seen in network topology of MobileNet, ResNet, DenseNet, GoogleNet, and Transformers, etc. Implementing these models using a conventional streaming architecture demands a massive amount of computational resources and a large hardware footprint. To address this challenge, we propose a folding network architecture at the block-layer level, optimizing resource utilization while maintaining performance efficiency.

In this work, we take into account the folding network architecture approach across two main ResNet vari-

ants [7], including ResNet18-ResNet34 and ResNet50-ResNet101-ResNet152. Where, the general network topology of ResNet50-ResNet101-ResNet152 is illustrated in Figure 1, with $\{N1, N2, N3, N4\}$ representing the number of iterations (or loops) for each residual block. Specifically, the values of $\{N1, N2, N3, N4\}$ are $\{3, 4, 6, 3\}$ for ResNet50, $\{3, 4, 23, 3\}$ for ResNet101, and $\{3, 8, 36, 3\}$ for ResNet152. For the shallower variants, ResNet18 and ResNet34, the concept of residual block iteration remains consistent, although the number of convolutional layers within each block and the filter sizes differ slightly.

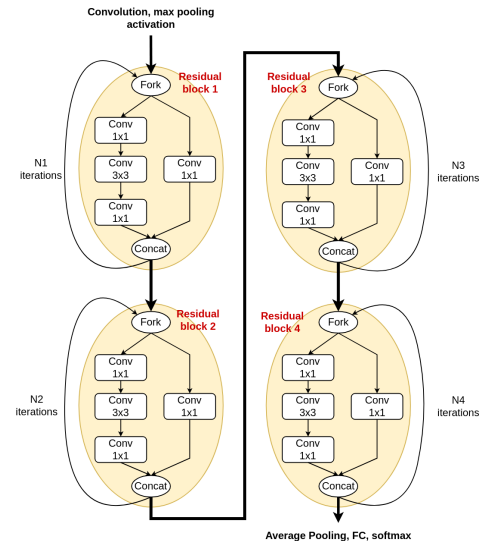


Fig. 1. General network topology for ResNet family.

In the subsequent sections, we will provide a detailed explanation of the methodology behind our approach (section II). Additionally, in section III, we will identify the required implementation resources, and then analyze key performance metrics, including throughput and latency, both pre- and post-folding.

II. FOLDING NETWORK ARCHITECTURE

The implementation of the folding system builds upon our work previously done for the conventional streaming architecture, as detailed in our earlier paper [1]. In that work, each

layer in the network was processed by two primary modules: the Sliding Window Layer (SWL) for parallel data streaming, and the Neuron Layer (Neu), which serves as the computation engine.

To facilitate folding, we retain the original structure for the first convolution, max-pooling, average-pooling, and fully-connected layers. However, the combination of layers within the residual blocks is restructured by adding a Residual Controller module (Res_Controller) to manage data flow across iterations (Figure 2). Additionally, each weight buffer in the neuron layers inside the residual block is expanded to accommodate the weight sets for all iterations. If the quantized weights of DNN model do not fit within the internal memory, the weight buffer size is adjusted to the maximum weight set size across all iterations. And then, for each iteration, the new weight set need to be fetched from off-chip memory to weight buffers (on-chip memory) located inside the neuron layers. Furthermore, a Residual Sliding Window (Res_SWL) module is integrated to store the intermediate activations during iterations (SWL_ITER), as well as to provide the output activation to the subsequent residual block (SWL_OUT).

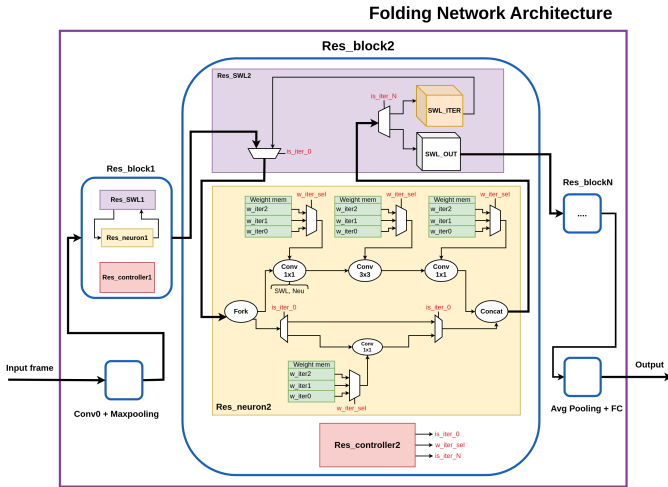


Fig. 2. A comprehensive design for folding network architecture.

III. RESULTS AND EVALUATION

For preliminary evaluation, we implemented the system on the FPGA Xilinx VCU128 running at 250 MHz.

The platform NNAWAQ [1] is utilized for performance estimation. In table I, ResNet-18 with a folding factor of $\{2, 2, 2, 2\}$ experiences a $2\times$ degradation in throughput. In contrast, ResNet-50 with a folding factor of $\{3, 4, 6, 3\}$ suffers from a $6\times$ reduction. This performance degradation is primarily influenced by the maximum folding factor across all residual blocks in the network. However, there is potential for improvement in the future by leveraging the resources saved through folding to optimize parallelism and synchronization among these residual blocks.

The synthesis resource utilization is summarized in Table II. Notably, the LUTs and LUTRAM usage is significantly

reduced in the folded architecture, due to fewer logic gates being required.

TABLE I
EVALUATION ON VCU128

Evaluation metrics	DNN architectures			
	Resnet18	Resnet18*	Resnet50	Resnet50**
Throughput	2214 fps	1107 fps	1107 fps	184 fps
Latency	1.6 ms	2.1 ms	4.9 ms	9.4 ms

* : folding factor of $\{2, 2, 2, 2\}$
** : folding factor of $\{3, 4, 6, 3\}$

TABLE II
DEPLOYMENT OF FOLDING NETWORK ON FPGA VCU128

Design resources	DNN architectures			
	Resnet18	Resnet18*	Resnet50	Resnet50**
LUTS	560840 (43.02%)	352696 (27.05%)	520636 (39.94%)	207622 (15.92%)
LUTRAM	281437 (46.83%)	223380 (37.17%)	165188 (27.49%)	50184 (8.35%)
BRAM	1185 (58.78%)	1000 (47.60%)	1719 (85.27%)	1417 (69.52%)
URAM	0	0	342 (35.63%)	328 (34.17%)
DSP	92 (1.02%)	64 (0.92%)	216 (2.39%)	96 (1.06%)

IV. RESEARCH PERSPECTIVES

With the potential in research orientation, we aim to design a Design Space Exploration (DSE) for selecting the folding factors as well as the granularity in the parallelisms of computation engines. The Front-End will be considered to be integrated for facilitating the importation and pre-optimize in the immediate representation (IR) of DNN model. For more convincing, the bigger network such as ResNet152, DenseNet, and Transformer aimed to be deployed in the future.

ACKNOWLEDGEMENTS

This work is funded by the Holigrail project within the PEPR-IA program, which actively contributes to the research and development of frugal SoC systems for AI.

REFERENCES

- [1] A. Prost-Boucle, A. Bourge, and F. Pétrot, "High-efficiency convolutional ternary neural networks with custom adder trees and weight compression," vol. 11, no. 3, pp. 1–24. [Online]. Available: <https://dl.acm.org/doi/10.1145/3270764>
- [2] Y. Umuroglu, N. J. Fraser, G. Gambardella, M. Blott, and et al, "FINN: A framework for the fast, scalable binarized neural network inference," in *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pp. 65–74.
- [3] I. Miro-Panades, V. Lorrain, L. Billod, and et al, "A 772J/frame ImageNet feature extractor accelerator on HD images at 30fps," in *2024 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*. IEEE, pp. 25–29. [Online]. Available: <https://ieeexplore.ieee.org/document/10808841/>
- [4] S. I. Venieris and C.-S. Bouganis, "fpgaConvNet: Mapping regular and irregular convolutional neural networks on FPGAs," vol. 30, no. 2, pp. 326–342. [Online]. Available: <https://ieeexplore.ieee.org/document/8401525/>
- [5] T. Aarrestad, V. Loncar, and et al, "Fast convolutional neural networks on FPGAs with hls4ml," vol. 2, no. 4, p. 045015.
- [6] M. Doumet, M. Stan, M. Hall, and V. Betz, "H2pipe: High throughput CNN inference on FPGAs with high-bandwidth memory," in *2024 34th International Conference on Field-Programmable Logic and Applications (FPL)*, pp. 69–77, ISSN: 1946-1488.
- [7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 770–778.

OnLine Unsupervised Learning of Self-Organizing Maps for SSL: a Hardware Implementation

Alexandre MASSON
LAAS-CNRS, Université de
Toulouse, CNRS, INSA
Toulouse, France
amasson@insa-toulouse.fr

Gaël LOUBET
LAAS-CNRS, Université de
Toulouse, CNRS, INSA
Toulouse, France
gael.loubet@laas.fr

Patrick DANÈS
LAAS-CNRS, Université de
Toulouse, CNRS, UPS
Toulouse, France
patrick.danes@laas.fr

Daniela DRAGOMIRESCU
LAAS-CNRS, Université de
Toulouse, CNRS, INSA
Toulouse, France
daniela@laas.fr

Abstract—This paper addresses the design of a Sound Source Localisation system using Self-Organizing Maps and its embedded, constrained hardware implementation on FPGA. A modified SOM is trained on the Sound Source Localisation for Robot dataset with GCC-PHAT as input features. Its implementation enables online unsupervised learning and real-time inference (latency < 1 ms), at low-power (< 150 mW) and light footprint. A single sound source can be classified to within 10°.

Index Terms—Edge AI, Self-Organizing Maps (SOM), Field-Programmable Gate Arrays (FPGA), Sound Source Localization (SSL), Algorithm-Architecture Adequation (AAA)

I. INTRODUCTION

Self-Organizing Map (SOM) learning is an unsupervised vector quantization algorithm widely used for multidimensional categorization [1]. It demonstrates its effectiveness across various applications such as ECG clustering [2] or anomaly detection [3]. Today, its use in Sound Source Localisation (SSL) is little studied [4].

An embedded hardware implementation based on the SOM on an FPGA of an SSL system is proposed. The key design objectives are: online unsupervised learning, consumption minimization, and compact footprint. Embedded, hardware implementation of SOM has already been proposed for image compression [5] or QAM demodulation [6], but never for SSL. The Sound Source Localisation for Robots (SSLR) speech dataset –recorded on a Pepper robot– is used for training [7].

II. THE SELF-ORGANIZING MAPS ALGORITHM

The SOM algorithm reduces N-dimensional input data to an M-dimensional space using a map of M nodes, each representing a potential cluster [1]. The SOM algorithm consists of two phases: learning and inference.

During the learning phase, initial node weights are set randomly. Weights are then updated by moving them toward the features of an N-dimensional input vector. By evaluating a distance between the input vector and each node, the “winner” node –the closest to the input features vector– can be selected. The neighborhood function, often a Gaussian centered on the winner node, determines how much and with what intensity each node moves toward the input features vector.

During inference, the distance between the input data and each node is calculated, identifying the “winning” node that indicates the estimated input group. Analyzing clusters from

the training dataset enable their labelling, allowing predictions on new input data. Therefore, while not a classification algorithm, SOM can be used to classify data into labeled clusters.

III. THE SSLR DATASET

The dataset includes audio files delivered by the 4 microphones of a Pepper robot. Each audio snapshot is labeled with the azimuth of the sound source [7]. The inter-aural model of Pepper’s head is complex, thus the machine learning approach seems more relevant than an analytical one.

The signals are divided into 8192-sample frames (about 170 ms for a 48 kHz sampling frequency). Generalized Cross Correlations with PHase Transform (GCC-PHAT) estimates the Time Difference Of Arrival (TDOA) between 2 microphones, providing audio cues related to the spatial origin of the sound. For microphones spaced at a maximum of 11 cm apart, sound takes about 0.32 ms to travel, *i.e.*, 16 samples. This means that only 32 points (16 before and 16 after the central point) are needed to identify the correlation peak, which represents the TDOA between 2 microphones. GCC-PHAT is applied to each of the 6 pairs built up with 4 microphones, giving 6 32-points GCC-PHAT outputs. This yields a total dimension of 192, matching the dimensionality of the SOM nodes.

IV. REVISITED SOM FOR HARDWARE IMPLEMENTATION

To optimize its hardware implementation, the original SOM algorithm was modified.

Firstly, the Euclidean distance was replaced by the Manhattan distance –a common strategy–, because it avoids the need for square and square root calculations [5].

Secondly, the learning rate is used as neighborhood function to lighten the SOM algorithm by avoiding heavy operations like exponential and divisions. Its value decays over iterations, ensuring a monotonic decrease as a function of the regression steps, as required by the SOM [1]. This method adjusts only the “winner” node for each new input. However, this solution shows limitations: after the first input, only the same winning node updates its weights causing it to dominate subsequent training data. To address this issue, the neighborhood function adjusts all node weights during the first iteration –enabling them to resemble the various input data–, then in subsequent iterations, only the winning node’s weights are adjusted.

V. SIMULATIONS

The designed model must differentiate sound source's spatial origins and detect whether the source is active or not. Evaluating this unsupervised model's performance is complex, necessitating various metrics.

Visual analysis of SOM clusters offers insights into SSL performances by assessing whether training clusters gather similar azimuths. With a homogeneous dataset and training on 19 nodes, we expect clusters to represent angle ranges of 20° (1 for inactivity and 18 for each 20° angular sector). Fig. 1 shows clusters from SOM training with a learning rate of 0.05 over 10 iterations. Although clusters are distinct, some overlap and size disparity exist, as well as a cluster containing all inactive sources, not just those labeled as such. This suggests that the model has successfully clustered the input data according to its spatial origin.

The SOM algorithm is then tested on new, unseen data. The results demonstrate the model's ability to group input data with similar azimuths into the same cluster during online inference.

Similar results are achieved by training and testing a SOM with 37 nodes (1 for inactivity and 36 for each 10° angular sector) over 10 iterations with a learning rate of 0.05. This learning rate is optimal: both higher and lower rates result in fewer or less distinct clusters; and results stabilize after 10 iterations. Additional experiments with 73 nodes (5° angular range) show complete clusters overlap, highlighting the limitations of the SOM algorithm for this application.

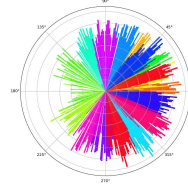
Since the SOM is unsupervised, it is tedious to calculate a precise accuracy. Since SLLR is not perfect, it is assumed that some micro-pauses in the recorded voice may have been labeled as active rather than inactive. With this hypothesis every input predicted in the inactive cases cluster is considered a true result to compute a Revised True Prediction Rate (RTPR). RTPR of 0.929 and 0.914 were obtained, respectively, for SOM of 19 and 37 clusters.

Another common metric was used: the Silhouette Coefficient (SC); which evaluates over $[-1; 1]$ the degree of similarity of an object to its own cluster compared to others, with higher values indicating better-defined groups. SC of 0.176 and 0.119 were obtained, respectively, for SOM of 19 and 37 clusters.

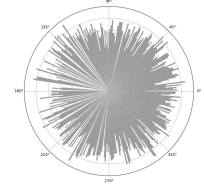
VI. HARDWARE IMPLEMENTATION

A trade-off between resources use and latency for real-time use was achieved to embed the SOM for SSL on FPGA. The designs were created in VHDL using Vitis HLS in C, then implemented and analyzed –for timing, energy consumption, and resource usage– on a Basys 3 board via Vivado.

"Serial" and "parallelized" implementations were tested. The "serial" implementation minimizes resources consumption by prioritizing hardware efficiency over execution speed, making it suitable for FPGA with limited resources or small problem sizes. The "parallelized" implementation minimizes the latency at the cost of increased resources usage. In summary, the "serial" implementation is resource-efficient due to reduced parallelism and staggered operations, and offers flexibility for adapting to various use cases.



(a) Histogram of the angles present in each clusters (each color represent one of the 18 clusters of spatial origins).



(b) Histogram of the angles belonging to the inactive case cluster.

Fig. 1. All the clusters formed during the training of the SOM on 19 nodes.

The "serial" implementation of the SOM of 19 and 37 clusters shows, respectively, a latency of 230 and 505 μ s, by using 2,488 and 2,612 LUT, representing an used area of 11.96 and 12.56 % of the xc7a35t FPGA, and consuming 133 and 129 mW; while the "parallelized" implementation of the SOM of 19 and 37 clusters shows, respectively, a latency of 78.25 and 131 μ s, by using 18,015 and 18,155 LUT.

VII. CONCLUSION AND PERSPECTIVES

The hardware implementation on a small FPGA of an SSL system based on a modified SOM algorithm was demonstrated. It enable online training and real-time inference, with a low power consumption, a high hardware efficiency, and a significant accuracy. Prospects include improving angular accuracy, processing features online –on the FPGA– from the microphone outputs, and testing on a Pepper robot.

VIII. ACKNOWLEDGMENT

This research used the SSLR Dataset made available by the Idiap Research Institute, Martigny, Switzerland, to whom the authors would like to express their sincere thanks [7].

REFERENCES

- [1] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.
- [2] J. Kim and P. Mazumder, "Energy-efficient hardware architecture of self-organizing map for ecg clustering in 65-nm cmos," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 9, pp. 1097–1101, 2017.
- [3] N. Li, K. Jiang, Z. Ma, X. Wei, X. Hong, and Y. Gong, "Anomaly detection via self-organizing map," in *2021 IEEE International Conference on Image Processing (ICIP)*, 2021, pp. 974–978.
- [4] E. Berglund and J. Sitte, "Sound source localisation through active audition," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 653–658.
- [5] G. C. Cardarilli, L. Di Nunzio, R. Fazzolari, M. Re, and S. Spanò, "Aw-som, an algorithm for high-speed learning in hardware self-organizing maps," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 2, pp. 380–384, 2020.
- [6] L. Canese, G. C. Cardarilli, L. Di Nunzio, R. Fazzolari, M. Re, and S. Spanò, "A hardware-oriented qam demodulation method driven by aw-som machine learning," in *2023 57th Asilomar Conference on Signals, Systems, and Computers*, 2023, pp. 937–941.
- [7] W. He, P. Motlicek, and J.-M. Odobez, "Deep Neural Networks for Multiple Speaker Detection and Localization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018.

Power of two fine-tuning for patient-specific cardiac arrhythmia recognition

Mathieu Chêne, Benoit Larras, Andreas Kaiser, Antoine Frappé

Univ. Lille, CNRS, Centrale Lille, Junia, Univ. Polytechnique Hauts-de-France, UMR 8520 - IEMN

Lille, France

mathieu.chene1@junia.com

Abstract—Deep learning is one of the main solution in cardiac arrhythmia classifier for its excellent performances. Research aims to embed these classifiers as near as possible from the sensors to process the data in real time and get ride of the wireless communication drawbacks. This work investigates the performance of a hardware-friendly quantified learning framework that addresses the issue of data confidentiality. It achieves the same performance than the state of the arts without data augmentation. It is also compatible with fine-tuning increasing the state of the art performances by up to 11 points with a model $\times 25000$ smaller.

I. INTRODUCTION

Cardiovascular diseases are one of the main causes of death in the world, with according to the World Health Organization an estimation of about 17.9 million deaths per year. With the increase of wearable electrocardiogram (ECG) sensors such as ECG belt, or ECG patches, it is now possible to conduct in-home and real-time heart condition monitoring which has been shown to effectively reduce heart failure hospitalization rate [1]. To process and analyse sensors measurement, research focuses on the use of Artificial Intelligence (AI) to embed Cardiac Arrhythmia Classifier (CAC) in devices. To comply with battery lifetime requirement, State of the Art (SoA) proposes multiple solutions for Ultra Low Power (ULP) CACs such as binary classifier [2], antictionary [3] or simple fully-connected neural network (NN) with only one hidden layer [4], [5]. However all these propositions are made for inference only and the model need to be trained or initialized offline using a server before deployment. Therefore, patient data must be shared, which raises confidentiality issues, or a general model must be integrated, but this is known to be less efficient due to physiological differences from one patient to another. Some studies suggest using transfer learning or fine-tuning (or both) [6], [7]. However, most of these works use a complex architecture of NNs (MobileNet, DenseNet) whose inference is energy-intensive. Furthermore, they are also trained online with full precision using 32-bit floating-point quantization (FP32), making them incompatible with ULP on-chip training. Nor do these solutions solve the problem of data privacy. In a previous work [8] a hardware-friendly framework for Quantized Training (QT) using INT8 and 8 bits power of two (POT8) gradients has been introduced. This paper proposes to extend the use of QT on MITBIH dataset. The second section presents the impact of the proposed QT approach on a simple, fully connected architecture. The third section studies

the performance of the same network after quantized fine-tuning using a general model trained in FP32.

II. PATIENT SPECIFIC QUANTIZED TRAINING FROM SCRATCH

In order to be compatible with on-chip training in constrained environment it is mandatory to quantize the backward path during training. In [8], the proposed framework quantizes Inference and Gradient Calculation steps as INT8 using General Matrix Multiplication Low Precision (GEMMLOWP) and the weight update (WU) operation is performed using gradient encoded as POT8. That quantization scheme makes the WU operation compatible with in-memory computing solving the standard SRAM read and write operation bottleneck and reducing the energy needed for this step by 13.7 %. As in [4] and [5] this work study the performances of fully connected NN with 1 hidden layer of 16 neurons. This network will be referred as MITBIHNet. This work will also use the same inputs than reference [2]. Heartbeats are represented with 11 features: the distance of the current R-peak with the previous and the next one, the distances of P,Q,S,T peaks with the current R-peak and the amplitude of P,Q,R,S,T peaks. Training is performed patient-specific with 70% of training data are used for training and 30% for the test. Table I compares the performance of QT from scratch with references [4], [5] and [2]. Reference [4] proposes the best accuracy score of all the presented solutions but is also the biggest model. Its F1-score is lower than the proposed solution by 2 points. When trained using biased training, the proposed approach achieves the same sensitivity than Zhao and when training data are augmented with SMOTE algorithm [9] it achieves the best SoA performances. Xu proposes the smallest model with excellent performances. However, binary classifier is designed in hardware which need "learned" threshold to work correctly. Its performances are equalled by the proposed approach when augmented data are used. Finally, [5], proposes a model in which weight are encoded as POT8 and stored on 4 bits which has the smallest size considering the same input features as [2]. However this quantization degrades the model performances which is out performed by 3 points by proposed QT without data augmentation.

III. PATIENT SPECIFIC QUANTIZED FINE-TUNING

Training from scratch is not always possible, as it is an energy-intensive process that can affect the battery life of the

Ref	Zhao [4]	Xu [2]	Gautier [5]	This work		
AI model	FNN $32 \times 16 \times 5$	Binary Classifier	MITBIHNet	MITBIHNet		
Model Size (Kb)	58.6	0.2	1.0	2.0		
Training Dataset Augmentation	No (Biased Training)	No	Yes (Data Duplication)	No	No (Biased Training)	Yes SMOTE
Quantized Training	No	N.A	No	INT8 POT8 WU gradients		
Weights precision	Fixed Point 16 bits	N.A	POT8	INT8		
Accuracy	99	98.5	95.69	98.11	96.6	99.33
F1-score	95	N.A	N.A	97.9	97.15	98.33
Sensitivity	86.22	98.5	N.A	69.23	85.99	99.22
Specificity	N.A	98.2	N.A	99.45	96.8	97.61

TABLE I
COMPARISON OF PERFORMANCES OF MODELS TRAINED FROM SCRATCH

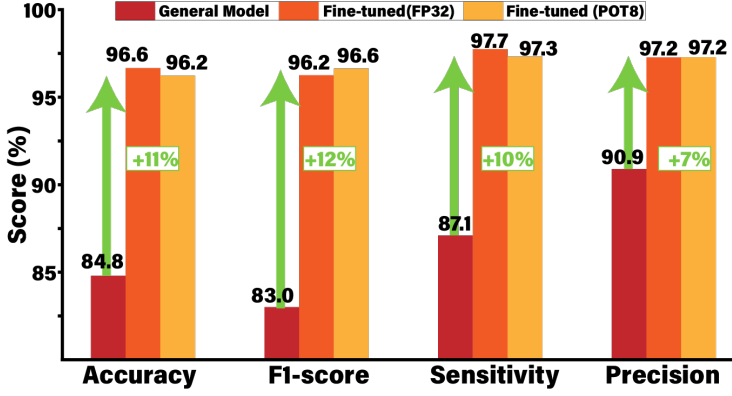


Fig. 1. Comparison of performances between general & fine-tuned models

integrated ECG sensor. In this section, it is therefore proposed to fine-tune a general CAC for a specific patient. To our knowledge, this is the first work proposing to edge fine-tune a NN model for the patient-specific classification of arrhythmias. During simulations, patients are randomly split in 2 lists. The public list, used to train the general model is composed of 80% of the patients. The private list, used for fine-tuning and test, is composed of 20% patients left. The general model train on the public dataset. As it is a classification problem Cross Entropy function is used as loss function. Adam optimizer is used as it is known to be more stable and converge faster than gradient descent. The general model is tested, patient specific, on the private patient list which data are divided in 2: the fine-tuning dataset composed of 70% of the private data and the test dataset composed the other 30%. The general model is fine-tuned on 1 epoch using the fine-tuning dataset and tested again. Fine-tuning is done using Gradient Descent algorithm.

Hence, the output layer is the only layer that is updated. Figure 1 compare the average performances between the general model and the fine-tuned model. Fine-tuning has been done with the quantized approach (in yellow) and without quantization (in orange). Over 10 simulations, fine-tuning has improved the performance of the NN model compared to its general version. Accuracy and F1-score have been increased by 11 and 12 points and sensitivity to arrhythmic heartbeats as well as the precision to detect them have been increased by respectively 10 and 7 points. In Table II, the average performances of the patient-specific quantized fine-tuned models are compared with references [6] and [10]. Both work use fine-tuning to enhance the performances of the initial

Ref	Aphale [10]	Bechinia [6]	This work
AI model	EfficientNet B7	MobileNet-V2	MITBIHNet
Training Dataset Augmentation	No	GAN Synthetic Data	No
On-Chip Fine-tuning	No	No	INT8 POT8 WU gradients
Precision	FP32	FP32	INT8
Accuracy	99.17	98.69	96.2
F1-score	97	90.8	96.6
Precision	99	95.8	97.2
Sensitivity	95	86.2	97.3

TABLE II
COMPARISON OF PERFORMANCES FOR FINE-TUNED MODEL

NNs. If [6] has a better accuracy than proposed approach by 2.5 points, mean precision is increased by 2 points and mean sensitivity is increased by 11 points. Accuracy from [10] is higher from 3 points but F1-score and Sensitivity are even or better in proposed QT. Finally It is important to note that proposed model has been trained using 8 bits resolution and it has 256 parameters against 66 millions for [10] and 3.4 millions for [6].

IV. CONCLUSION

In this paper, it has been proposed to apply a hardware-friendly QT for cardiac arrhythmia recognition. This approach would answer the data privacy issue and achieve SoT accuracy without data enhancement with a gain of 1 point in sensitivity. The model is also compatible with on-chip fine-tuning increasing performances by up to 11 points with a model $\times 25000$ smaller compared to other fine-tuned solutions of the SoA.

REFERENCES

- [1] A. Bui, "Home monitoring for heart failure management," *JACC*, 2012.
- [2] X. Xu, "A 2.66 μ w clinician-like cardiac arrhythmia watchdog based on p-qrs-t for wearable applications," *IEEE TBioCaS*, 2022.
- [3] J. Duforest, "Antidictionary-based cardiac arrhythmia classification for smart eeg sensors," in *IEEE ISCAS*, 2022.
- [4] Y. Zhao, "A 13.34 w event-driven patient-specific arrhythmia classifier for wearable eeg sensors," *IEEE TBioCaS*, 2020.
- [5] A. Gautier, "Evaluation of power-of-two quantization for multiplier-less near-memory and in-memory computing schemes for biomedical applications," in *2023 IEEE NorCAS*, 2023.
- [6] H. Bechinia, "Approach based lightweight custom convolutional neural network and fine-tuned mobilenet-v2 for eeg arrhythmia signals classification," *IEEE Access*, 2024.
- [7] A. Avetisyan, "Deep neural networks generalization and fine-tuning for 12-lead eeg classification," *Biomedical Signal Processing and Control*, 2024.
- [8] M. Chêne, "An in-cell weight update scheme using one-hot gradients for on-chip learning," in *IEEE ICECS*, 2024.
- [9] N. V. Chawla, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, 2002.
- [10] S. Aphale, "High accuracy arrhythmia classification using transfer learning with fine-tuning," in *IEEE UEMCON*, 2022.

JL2 Vertical GAA Transistor Based Standard Cell Library Design Flow

Sara Mannaa, Cédric Marchand, Damien Deleruyelle, Alberto Bosio, Ian O'Connor
Ecole Centrale de Lyon, INSA Lyon, CNRS, Université Claude Bernard Lyon 1, CPE Lyon, INL,
UMR5270, 69130 Ecully, France

Abstract— Based on a vertical nanowire technology with a 2-gate stack approach, we describe our flow toward the generation of standard cell libraries: from the circuit and physical design of logic cells to benchmarking. Compared to a single-gate approach, our library shows better performance (delay and power values) with smaller footprint.

I. INTRODUCTION

The extensive implementation of Artificial Intelligence (AI) is continuously pushing the performance requirements of hardware processors to ever-higher levels. This comes with the limitations imposed by the physical properties of traditional CMOS itself with scaling. Thus, new technological breakthroughs are needed to maintain the desired performance and computational requirements along with the associated constraints on footprint, delay and energy consumption. Vertical Nanowire Field Effect Transistor (VNWFET) is an emerging technology that promises to improve the sustainability of future transistor scaling beyond the limitations of conventional lateral devices. With its 3D Gate-All-Around (GAA) architecture, this technology enables designs with improved energy efficiency and a smaller footprint. It also supports the gate stacking approach to benefit designs with multiple transistors in series. In this work, we adopt a VNWFET device [1] with a junctionless (JL) architecture, and we aim to investigate the advantages of implementing gate stacking approach as compared to single gate variant.

II. JL2 STANDARD CELL LIBRARY CHARACTERIZATION

In this section, we describe our design flow toward the generation of standard cell library based on VNWFET technology. Throughout this work, we adopt an accurate executable compact model [2] implemented in Verilog-A

with a parameter set fitted to measurements of an experimental fabricated VNWFET device.

A. Logic Cell Design and Characterization

Performing electrical simulation on logic cells is considered to be one of the most important steps in such a flow. It allows the evaluation of the necessary performance characterization such as cells' output behavior, delay and power consumption. It also acts as the main building block toward the generation of the cells' physical design, parasitic extraction (PEX) and thus accurate performance metric measurements. To this end, we adapted our single-gate (JL1) logic cell designs [3] to benefit from the 2-gate stacking (JL2) architecture. JL2 is advantageous for circuit structures with two series transistors, such as the pull-up network in a 2-input NOR gate, the pull-down network in a 2-input NAND gate or both networks in a 2-input XOR gate. We then carried out electrical simulations with HSpice™ to characterize JL2 logic cells under different drive strengths. Table I presents the selected drive strengths of the cells present in our libraries such that: OPnXk_Style indicates the Boolean operation OP, the number of inputs n, the drive strength k, the logic design style (i.e. complementary static logic). Subsequently, we conducted both static and dynamic analysis of the cells, considering all possible input transitions that result in output transitions.

B. Logic Cell Physical Design and Verification

We then expanded our physical design flow [3] and design rules for JL1 cells to incorporate an additional gate layer, accommodating the JL2 architecture. This allowed us to generate a preliminary physical layout for the cells in the standard cell library (in GDSII file format) and thus the Library Exchange File (LEF). JL2 designs exhibited a substantial improvement in cell area efficiency, with an average decrease of 39.35% relative to JL1 cells. In order to verify

that the generated layout guarantees the functional logic behavior of the standard cells, we completed a verification test using Global TCAD Solutions (GTS tools. Fig. 1 shows an example 3D view of an XOR2 gate, exemplifying one of the more complex cells. After completion of the PEX, a transient analysis was carried out using the generated netlist which allowed an accurate measurements of timing and power consumption. The obtained values illustrate the improvements in capacitive and resistive parasitic values extracted from JL2 designs as compared to that of JL1. For JL2, average delay and energy/transition increased by 11% and 8.4% respectively as compared to ideal netlist whereas for JL1 these values increased by 18.6% and 14.2% respectively.

C. Library Evaluation

In order to verify the validity of the generated library on one side and to illustrate the advantages of JL2 implementation on the other side, we chose to synthesize a Full Adder (FA) circuit. FA is considered to be an important benchmark in technology evaluation as it has a simple structure and widely used in processors. Fig.2 shows the comparison of performance metrics obtained after the synthesis of the FA by JL2 library to that obtained by JL1 library.

TABLE I. VNWFET LIBRARY STANDARD CELLS WHERE K=1 CORRESPONDS TO 4 NWS.

Logic Gate	Drive Strength (k)	Variant
INVXk_CStatic	1,6,11,16	JL1
BUFxk_CStatic	1,6,11,16	JL1
NAND2Xk_CStatic	1,6,11	JL1,JL2
NOR2Xk_CStatic	1,6,11	JL1,JL2
XOR2Xk_CStatic	1,6,11	JL1,JL2
Asynch_DFFXk_Cstatic	1	JL1,JL2

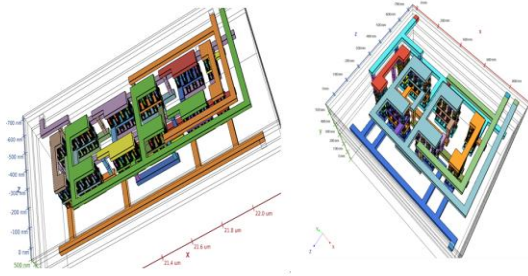


Figure 1. 3D view of XOR2 gate with JL1 variant (left) and JL2 variant (right).

III. CONCLUSION AND PERSPECTIVE

In this work, we emphasized on the advantages of JL2-VNWFET over JL1. However, this flow can be extended to 3-gate stacking as it is also supported by this technology thus enhancing designs with 3 series transistors (e.g. 3-input XOR gate). Indeed, it will guarantee having more compact cells with smaller parasitic values which will allow the synthesis of designs (e.g. FA) with even smaller footprint and better performance metrics.

ACKNOWLEDGMENT

This work has been funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 101016776 (FVLLMONTI).

REFERENCES

- [1] A. Kumar, J. Muller, S. Pelloquin, A. Lecestre, and G. Larrieu, "Logic gates based on 3d vertical junctionless gate-all-around transistors with reliable multilevel contact engineering," *Nano Letters*, 2024.
- [2] C. Mukherjee, A. Poittevin, I. O'Connor, G. Larrieu, and C. Maneux, "Compact modeling of 3d vertical junction-less gate-all-around silicon nanowire transistors towards 3d logic design," *Solid-State Electronics*, vol. 183, p. 108125, 2021.
- [3] S. Manna, C. Marchand, D. Deleruyelle, B. Deveautour, A. Bosio, C. Lenz, O. Baumgartner, and I. O'Connor, "3d vnwfet-based standard cell library design flow: from circuit and physical design to logic synthesis," in *2024 IFIP/IEEE 32nd International Conference on Very Large Scale Integration (VLSI-SoC)*. IEEE, 2024, pp. 1–4.

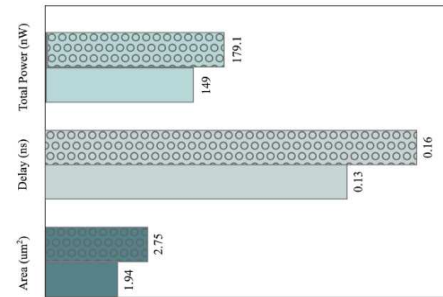


Figure 2. Comparison of obtained metrics of synthesized FA between VNWFET-JL2 library (solid bars) and VNWFET-JL1 (hatched bars) showed the following gains: area (1.4X), delay (1.2X) and power (1.75X and 1.2 for leakage and dynamic power respectively).

Non-volatile Ferroelectric-AND (FeAND) memory cell design

Basile Darne, Alberto Bosio, Miqueas Filsinger, Damien Deleruyelle, Ian O'Connor, Bertrand Vilquin, Cédric Marchand
Centrale Lyon, INSA Lyon, CNRS,
Université Claude Bernard Lyon 1,
CPE Lyon,
INL, UMR5270, 69130 Ecully, France
basile.darne@ec-lyon.fr

I. INTRODUCTION

With the growing demand of IoT devices, new electronic circuits are required in order to work under unstable power. This aspect is crucial for memories, because volatile memories lose their state when the power is off and conventional non-volatile memories are slow and energy-consuming. Ferroelectric materials offer a promising alternative for non-volatile memories when it comes to latency and energy consumption. We present a novel memory device made of a ferroelectric capacitor and a CMOS inverter, that will be designed as a non-volatile backup for a volatile memory cell.

II. PROPOSED CIRCUIT

Whereas most of the work conducted around ferroelectric memory devices in the literature consists in reading the stored information as a current, we propose a new circuit which allows to read the stored data directly as a voltage.

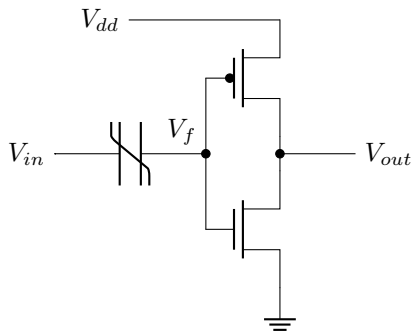


Figure 1: Proposed FeAND memory cell

The device contains a ferroelectric capacitor connected in series with the input of CMOS inverter. The voltage V_{in} is the data input, applying a voltage on this node allows to write data in the memory. The voltage V_{dd} serves as a read input, the data is read when V_{dd} is a logical '1'. The output is V_{out} , if the value stored is '1' then V_{out} will be '1' during read, otherwise V_{out} is '0'.

This device is based on two operating schemes, read and write:

- Write: V_{dd} is set to '0' and a programming voltage is applied at V_{in} . In order to store a '0', V_{in} must be negative (-5V), in order to store a '1', V_{in} must be positive (+4V).
- Read: V_{in} is '0' grounded and a read pulse is applied at V_{dd} . Depending on the memory state, the output V_{out} will be either '0' or '1'.

This device stores information in a non-volatile way using the polarity of the ferroelectric capacitor. One memory state corresponds to a positive polarity being stored in the ferroelectric capacitor and the other one corresponds to the storage of a negative polarity. We choose the following convention: if the polarity is positive then $mem = 1$, if the polarity is negative then $mem = 0$, where mem indicates the logical value of the memory state. In order to put the ferroelectric capacitor in a positive polarity, a positive voltage $V_{in} = 4V$ must be applied and in order to write a negative polarity the input voltage must be $V_{in} = -5V$.

The polarity state stored in the ferroelectric capacitor will affect the charge distribution inside the ferroelectric capacitor, and in turn inside the floating node V_f . If the polarity is positive, V_f will be in a low value V_{fp} . If the polarity is negative, V_f will be in a high value V_{fm} .

Therefore, the information in the device is stored as a polarity value, and in turn as a value of V_f . The device is designed in such a way that if $mem = 1$ then $V_f = V_{fp}$ is low enough to cause the CMOS inverter to output $V_{out} = 1$. Similarly, when $mem = 0$ then $V_f = V_{fm}$ must be high enough to trigger an output of $V_{out} = 0$ from the inverter. On the data is written in the memory, the value of V_f is set and will be retained. Applying a reading pulse on V_{dd} will then allow to power on the inverter and read the corresponding output. If $V_{out} = 1$ then we know that $mem = 1$, if $V_{out} = 0$ then $mem = 0$.

One important parameter of the design is the threshold voltage of each transistor of the inverter. We define $V_{TN} = V_{th,n}$ the threshold voltage of the N-type transistor, and $V_{TP} = V_{dd} + V_{th,p}$ where V_{dd} is the supply voltage of the inverter and $V_{th,p}$ the threshold voltage of the P-type transistor ($V_{th,p} < 0$). For the N type transistor, we know that if $V_f < V_{TN}$ the transistor is blocked, and if $V_f > V_{TN}$

the transistor is passing, where V_f is the voltage applied on the gate of the inverter. For the P type transistor we know that if $V_{gs} < V_{th,p}$ the transistor is passing (for the P type transistor, $V_{gs} = V_f \vee V_{dd}$, so this condition is equivalent to $V_f < V_{dd} + V_{th,p} = V_{TP}$). In a similar way, we get that if $V_f > V_{TP}$ then the P transistor is blocked. To sum up (see figure 2), we know that if $V_f < V_{TN}$ and $V_f < V_{TP}$ then $V_{out} = 1$, if $V_f > V_{TN}$ and $V_f > V_{TP}$ then $V_{out} = 0$.

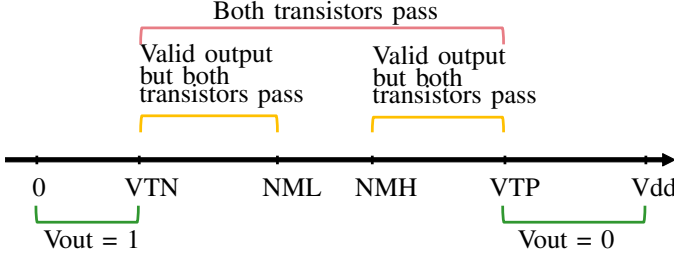


Figure 2: Behavior of the device depending on V_f

We know that with a CMOS inverter, if the input voltage is lower than the low noise margin (NML) the output will be a valid logical '1' and if the input is higher than the high noise margin (NMH) the output will be a valid logical '0'. However in these two cases, even if the output is valid the power consumption will be high. This is not desired for our device.

III. DESIGN METHODOLOGY

Based on the previous theoretical study, this is the methodology we will follow:

- 1) Optimization of CMOS inverter: choice of transistor type, transistor sizing, Goal = balance NMH/NML
- 2) Define (arbitrarily) read/write pulses
- 3) Optimization of fecap size, Goal = find fecap size for which V_{fp} and V_{fm} satisfy the design conditions
- 4) Optimization of read/write pulses, Goal = reduce V_f , reduce energy consumption

IV. RESULTS

This work is conducted with the GlobalFoundries 28SLP transistor technology. ForThe study is limited only to thin oxide regular threshold voltage transistors.

The input and V_{dd} pulses are set as follows:

- $V_{progm} = -5V$
- $V_{progp} = +4V$
- $v_{dd} \text{ max} = 1V$
- T write
- T read

Once the writing and reading pulses are set, the following sizes of components give a functional memory cell (as highlighted by figure 3):

- $L_n = 30nm$, $W_n = 80nm$
- $L_p = 75nm$, $W_p = 200nm$
- $A_{fecap} = 0.5 \cdot A_{tot} = 8700nm^2$

We can see on the results presented in figure 3 that when $mem = 1$, we get $V_{out} = 1$ when $V_{dd} = 1$. In a similar way,

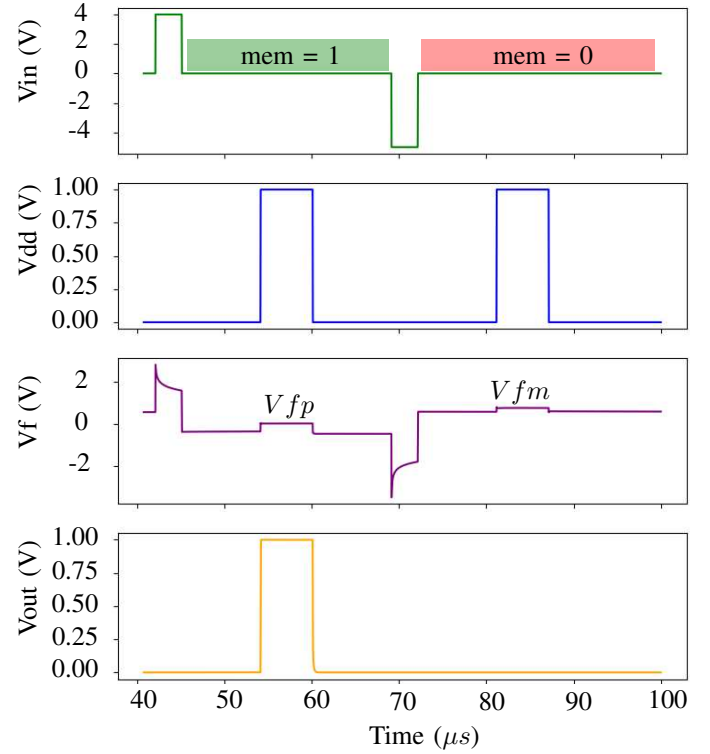


Figure 3: FeAND simulation results

when $mem = 0$ we get $V_{out} = 0$ when $V_{dd} = 1$. This device thus displays the behavior of an AND logic gate having mem and V_{dd} as its inputs:

mem	V_{dd}	V_{out}
0	0	0
0	1	0
1	0	0
1	1	1

V. CONCLUSION

We have presented a functional memory cell where the read mechanism is performed by applying a voltage on V_{dd} and reading V_{out} . The write mechanism is performed through a positive or negative voltage applied at V_{in} .

This circuit offers some advantages:

- no need to use a sense amplifier for a single cell, because the output is a voltage
- reduced power consumption because the device is designed in such a way that always one of the transistors is blocked, limiting power draw from V_{dd} .

However, the circuit has also some drawbacks, because no multilevel storage is possible, and the value of V_f must be kept low enough not to exceed the maximum voltage the gates of the transistors can tolerate.

Développement de nouveaux algorithmes d'IA pour le traitement de données radar 4D

Adrien Hernandez
OPmobility
Levallois-Perret, France
adrien.hernandez@opmobility.com

Alain Pegatoquet
LEAT, Université Côte d'Azur
Sophia Antipolis, France
alain.pegatoquet@univ-cotedazur.fr

Jean-Yves Dauvignac
LEAT, Université Côte d'Azur
Sophia Antipolis, France
jean-yves.dauvignac@univ-cotedazur.fr

Hadrien Passet
OPmobility
Levallois-Perret, France
hadrien.passet@opmobility.com

Abstract—Cette thèse vise à développer de nouveaux algorithmes d'intelligence artificielle pour le traitement des données issues de radars 4D dans le contexte des systèmes avancés d'aide à la conduite (ADAS). L'objectif est de proposer des solutions basées sur l'apprentissage automatique profond afin d'améliorer la détection, la classification, et le suivi des objets dans l'environnement de véhicules autonomes. Les contraintes liées à l'embarquabilité des différentes solutions proposées seront considérées dès le début de la thèse afin de dimensionner les algorithmes de traitement mis en œuvre. Cette thèse Cifre s'effectue en collaboration entre le Laboratoire d'Electronique, Antennes et Télécommunications (LEAT) de l'université Côte d'Azur et l'équipementier automobile OPmobility (ex Plastic Omnium).

Index Terms—Radar 4D, Intelligence Artificielle, Machine Learning, Deep Learning, Traitement du Signal, Systèmes embarqués, ADAS

I. CONTEXTE ET INTRODUCTION

La sécurité des conducteurs, des passagers et de tous les autres usagers de la route est devenue un enjeu majeur au cours des dernières décennies. Dans ce but, les capteurs radar ont été considérés comme des moyens essentiels pour détecter les autres véhicules, les piétons ou les cyclistes, ainsi que l'environnement routier. Le radar est insensible aux mauvaises conditions de luminosité et aux intempéries, et peut mesurer directement la distance, la vitesse radiale, et, avec un système d'antennes approprié, également l'angle des objets éloignés. Initialement, l'attention était portée sur l'avertissement de distance et l'évitement de collision, mais avec l'augmentation de la maturité et de la complexité des systèmes, les fonctionnalités inclus le régulateur de vitesse adaptatif (ACC – Adaptive Cruise Control), le freinage d'urgence automatique (AEB – Automatic Emergency Braking), la détection d'angle mort (BSD – Blind Spot Detection) ou l'assistance au changement de voie (LCA – Lane Change Assist). De nos jours, les fonctions de sécurité protégeant les passagers et les usagers vulnérables de la route jouent un rôle primordial.

II. RADAR 4D AUTOMOBILE

Dans le contexte des ADAS dans l'automobile, les radars les plus utilisés sont les radars à onde continue modulée

en fréquence (*Frequency Modulated Continuous Wave* ou FMCW). Cette technique permet d'estimer la distance (par analyse du décalage de fréquence), la vitesse radiale (via l'effet Doppler), ainsi que l'angle d'arrivée (grâce à des techniques avancées telles que le MIMO) [1].

Cependant, ces radars ne fournissent qu'une information spatiale restreinte à un seul plan, ce qui peut engendrer des ambiguïtés dans certains scénarios de conduite autonome. Par exemple, un radar classique ne permet pas de distinguer un pont d'un camion arrêté en travers de la chaussée. Le radar détecte une cible mais ne dispose pas de suffisamment d'informations pour permettre une prise de décision sans ambiguïté.

Pour lever ces ambiguïtés, il est nécessaire d'accéder à une dimension spatiale supplémentaire, l'élévation. Les radars 4D répondent précisément à ce besoin en mesurant quatre dimensions : la distance, la vitesse, l'azimut et l'élévation. Cependant, l'exploitation de leurs données requiert la mise en œuvre de plusieurs méthodes de traitement du signal afin d'extraire les informations pertinentes pour les fonctions ADAS. La donnée radar brute est ainsi tout d'abord convertie en un nuage de points (détection). A partir de ce nuage de points, les traitements incluent le regroupement (clustering), la classification et le suivi dans le temps (tracking) des cibles. Afin d'améliorer les performances des fonctionnalités ADAS, il est essentiel d'accroître la qualité de l'information extraite de ces fonctions.

III. OBJECTIFS DE LA THÈSE

Au cours de cette thèse, nous souhaitons apporter des améliorations aux fonctionnalités ADAS afin de relever le niveau d'autonomie des véhicules par une meilleure prise de décision. Pour cela, nous proposons d'explorer les techniques d'intelligence artificielle appliquées aux données de radar 4D. Nous nous attachons également à conserver une dimension d'intelligence embarquable et donc frugale, adaptée aux contraintes des systèmes embarqués automobiles.

La mise en œuvre de méthodes d'apprentissage automatique nécessite l'accès à un jeu de données adapté. Trois

approches principales permettent d'obtenir un jeu de données : l'acquisition expérimentale, la génération synthétique, ou l'utilisation de jeux de données open-source. Dans notre cas, la collecte de données impliquerait l'installation d'un radar sur un véhicule et la réalisation de campagnes de mesure (i.e. roulage). Toutefois, cette approche étant particulièrement chronophage, nous ne l'avons pas retenue. La génération synthétique de données radar, que nous avons abordé au début de cette thèse, requiert malheureusement des temps de calcul très importants. Nous avons donc choisi de privilégier l'utilisation de jeux de données open-source. Plusieurs d'entre eux sont disponibles [2], [3], selon les tâches visées. En effet, pour la détection par exemple, il est nécessaire d'accéder à la donnée radar brute ainsi qu'au nuage de points associé. Les datasets RaDelft [4] et K-Radar [5] sont pertinents à cet égard, car ils contiennent à la fois des données brutes issues de radars 4D, de lidars et de caméras. Pour le clustering, la classification et le tracking, un nuage de points prétraité ainsi qu'une labellisation des objets sont requis. Les jeux de données View-of-Delft [6] et K-Radar répondent également à ces besoins.

Différentes approches de traitement des données radar 4D par apprentissage automatique ont été proposées récemment. Avec le dataset RaDelft [4], les auteurs proposent un réseau de neurones pour générer des nuages de points à partir des données radar 4D. Ce modèle neuronal est entraîné en utilisant les données issues du lidar comme vérité terrain. Les résultats montrent que les performances obtenues avec ce réseau sont supérieures aux méthodes de détection classiques. Sa modularité constitue également un atout majeur, l'architecture ayant été conçue pour s'adapter aux contraintes spécifiques de chaque application. Les auteurs du dataset View-of-Delft [6] proposent une méthode de clusterisation et de classification de nuages de points basée sur une approche par pointPillars [7]. Cette méthode convertit le nuage de points en une pseudo-image pour ensuite extraire des caractéristiques par convolutions, puis en utilisant une tête de détection pour produire les boîtes englobantes (*bounding box*). On peut également

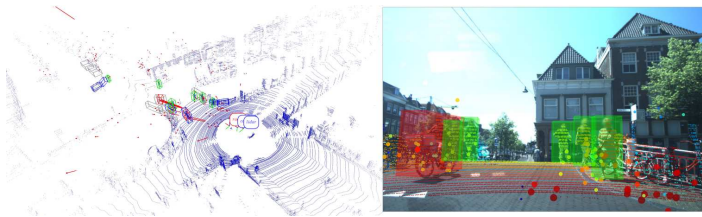


Fig. 1. Extrait du dataset View-of-Delft, pointillés fin issus du lidar, points plus gros issus du radar 4D, photo issue de la caméra avec clustering et classification des points par bounding box.

citer RadarPillars, une autre méthode de génération de boîtes englobantes à partir d'un radar 4D. Cette approche est aussi basée sur les pointPillars mais inclut également un mécanisme d'attention pour la gestion des pillars. Cette méthode a aussi été évaluée sur le dataset View-of-Delft.

IV. TRAVAUX EN COURS

Nous sommes actuellement en train de prendre en main le code et le dataset RaDelft [4]. Nous avons relancé des entraînements du modèle pour reproduire les résultats présentés dans l'article. Malheureusement, malgré l'aide de l'auteur, nous n'arrivons pas à obtenir les mêmes résultats. Dans un second temps, nous allons faire du profiling sur le réseau RaDelft afin d'identifier les parties gourmandes en mémoire et ressources computationnelles. L'objectif est ensuite d'optimiser ces modèles par des techniques de compression (quantification, distillation de connaissances,...) dans le but de déployer cette solution sur une architecture embarquée. Nous envisageons également d'investiguer les réseaux de Kolmogorov-Arnold (KAN) [8] pour leur apport en interprétabilité.

V. CONCLUSION






Ce projet de recherche vise à proposer des solutions innovantes pour améliorer la détection, la classification et le suivi des objets dans les systèmes ADAS, en s'appuyant sur les capacités des radars 4D et la puissance des algorithmes d'intelligence artificielle. Grâce à une meilleure exploitation de l'information spatiale en quatre dimensions (distance, vitesse, azimuth, élévation), les radars 4D permettent de lever certaines ambiguïtés encore présentes avec les capteurs conventionnels.

Notre objectif est d'explorer des méthodes d'apprentissage automatique profond capables de s'adapter aux contraintes croissantes de précision, de robustesse et de temps réel. Pour cela, nous nous appuyons sur des jeux de données open-source variés et réalistes, intégrant des annotations multi-capteurs (caméra, lidar, radar).

REFERENCES

- [1] C. Waldschmidt, J. Hasch, and W. Menzel, "Automotive radar — from first efforts to future systems," *IEEE Journal of Microwaves*, vol. 1, no. 1, pp. 135–148, 2021.
- [2] Y. Zhou, L. Liu, H. Zhao, M. López-Benítez, L. Yu, and Y. Yue, "Towards deep radar perception for autonomous driving: Datasets, methods, and challenges," *Sensors*, vol. 22, no. 11, 2022. [Online]. Available: <https://www.mdpi.com/1424-8220/22/11/4208>
- [3] X. Peng, M. Tang, H. Sun, L. Servadei, and R. Wille, "4d mmwave radar in adverse environments for autonomous driving: A survey," 2025. [Online]. Available: <https://arxiv.org/abs/2503.24091>
- [4] I. Roldan, A. Palffy, J. F. P. Kooij, D. M. Gavrila, F. Fioranelli, and A. Yarovoy, "A deep automotive radar detector using the radelft dataset," *IEEE Transactions on Radar Systems*, vol. 2, pp. 1062–1075, 2024.
- [5] D.-H. Paek, S.-H. KONG, and K. T. Wijaya, "K-radar: 4d radar object detection for autonomous driving in various weather conditions," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 3819–3829. [Online]. Available: <https://arxiv.org/pdf/2206.08171v4>
- [6] A. Palffy, E. Pool, S. Baratam, J. F. P. Kooij, and D. M. Gavrila, "Multi-class road user detection with 3+1d radar in the view-of-delft dataset," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4961–4968, 2022.
- [7] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," 2019. [Online]. Available: <https://arxiv.org/abs/1812.05784>
- [8] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, "Kan: Kolmogorov-arnold networks," 2025. [Online]. Available: <https://arxiv.org/abs/2404.19756>

FPGA-based Deep Learning Object Detection for Event Cameras in Drones

Arthur Gaudard , Ismail Amessegher , Matthieu Arzel , Matthieu Léonardon , Hugo Le Blevec 
IMT Atlantique, Lab-STICC, UMR CNRS 6285, F-29238, Brest, France, *first-name.last-name@imt-atlantique.fr*

Abstract—This paper explores the integration of Deep Learning object detection models with event cameras on FPGA hardware for drone applications. Event cameras, known for their ultra-low latency and low power consumption, are ideal for real-time tasks in embedded systems. We focus on adapting the YOLOv3 and tiny-YOLOv3 models, originally designed for RGB data, to process event data. Using the DSEC-MOD dataset, we demonstrate the feasibility of this approach. First, we show that models developed for RGB data can be adapted to event data, and then we propose an FPGA implementation of tiny-YOLOv3 using the FINN framework.

Index Terms—Event cameras, Computer vision, Object detection, FPGA, Drones

I. INTRODUCTION

The integration of event cameras in embedded systems, particularly drones, represents a significant advancement in sensing technology. Event cameras, inspired by the human retina, capture changes in light intensity asynchronously, offering several advantages over traditional frame-based cameras [1]. These advantages include ultra-low latency, high dynamic range, and low power consumption, making them ideal for real-time applications in drones where rapid response times and energy efficiency are crucial.

Leveraging the unusual type of signal output by event cameras – asynchronous ternary data – requires adequate processing techniques. Deep Learning computer vision models, built up to now for RGB signals such as classification and segmentation algorithms, have to be redesigned in order to efficiently extract meaningful information from the sparse and asynchronous event data. These models could enable drones to perform complex tasks such as object detection, tracking, and navigation in dynamic environments with high precision and reactivity. However, the implementation of these Deep Learning models on traditional hardware platforms like CPUs, microcontrollers, or GPUs presents challenges in terms of power consumption, processing speed, and flexibility. Field-Programmable Gate Arrays (FPGAs) emerge as a superior alternative for embedded systems due to their parallel processing capabilities, reconfigurability, and energy efficiency. FPGAs can be customized to optimally execute Deep Learning algorithms, providing the necessary computational power while maintaining low power consumption, which is essential for the constrained environments of drone operations.

This paper explores the integration of Deep Learning object detection models with event cameras on FPGA hardware. We focus on the example of YOLOv3 [2] and tiny-YOLOv3

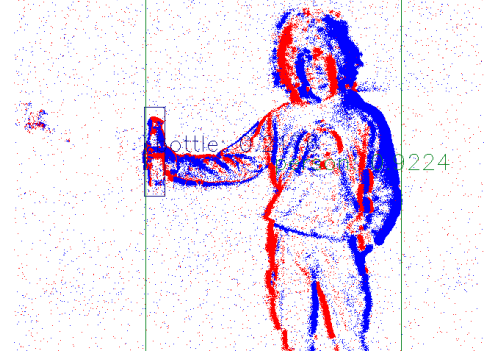


Fig. 1: Example of the DPU pipeline running on a Kria KV260 board

[3], and we use the event-based dataset DSEC-MOD [4] to train and evaluate the model, showcasing the feasibility and advantages of this approach.

II. DPU IMPLEMENTATION

PYNQ is a Python framework that facilitates communication between the CPU and FPGA, providing access to the Xilinx Deep Learning Processing Unit (DPU) [5]. The DPU acts as a hardware accelerator for Deep Learning inference, capable of running different models without design changes. Its principle is described in figure 2. We used a pre-compiled YOLOv3 model trained on the VOC12 RGB dataset [6] to test our pipeline on a Kria KV260 board. The model successfully detected objects in event data (see figure 1), demonstrating the adaptability of RGB-trained models to event data with minimal fine-tuning.

III. DSEC-MOD DATASET

In this work, we use the DSEC-MOD dataset, a curated subset of the DSEC (Dynamic Stereo Event Camera) [7] dataset, specifically annotated for moving object detection tasks. The dataset comprises sequences captured in diverse driving scenarios, with annotations focusing on a single class

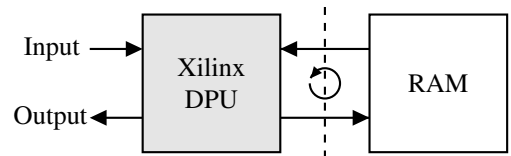


Fig. 2: Xilinx DPU overview

TABLE I: Compared Performance and FPGA Resource Utilization of Quantized Models

Quantization	mAP@50		FPGA Resource Utilization					
	Event	RGB	LUT	SRL	FF	BRAM	URAM	DSP
FP32	0.52	0.58	-	-	-	-	-	-
INT8	0.51	0.58	124193	18451	140538	518	5	643
INT4	0.51	0.58	83277	10994	90193	172	5	463
INT3	0.49	0.57	108227	7030	93082	148	0	1633
INT2	0.46	0.56	94752	5265	76755	97	4	1333

labeled as “moving objects.” This class encompasses various dynamic objects belonging to a total of 8 classes, reflecting the critical entities encountered in autonomous driving environments. DSEC-MOD provides synchronized RGB frames and event streams, enabling comprehensive analysis of multimodal detection approaches. The dataset contains 13,314 frames of 640×480 resolution, with 10,495 frames allocated for training and 2,819 frames for testing.

IV. FINN IMPLEMENTATION

FINN [8] is an open-source framework developed for deploying quantized neural networks (QNNs) on FPGA. FINN provides an end-to-end flow for exploring and implementing QNN inference solutions, generating dataflow-style architectures customized for each network, as shown on figure 3. This pipelined architecture allows for better performance than the DPU in terms of latency and throughput, at the cost of a higher resource utilization and no versatility.

In this study, we use Tiny-YOLOv3, with a redesigned backbone to enhance memory efficiency and reduce computational cost. Initially, we train a floating-point (FP32) model as a baseline reference. Subsequently, we quantize the model to 8-bit, 4-bit, and 2-bit precision levels, evaluating each version’s performance. The training and evaluation are conducted using both the DSEC-MOD Event and DSEC-MOD RGB datasets. The accuracy is measured using the mean Average Precision at an Intersection over Union (IoU) threshold of 0.5, commonly referred to as mAP@50. The results of these experiments are summarized in table I.

The quantized versions of our custom Tiny-YOLOv3 model are exported to the ONNX format, facilitating compatibility with the FINN framework. Within FINN, these models are transformed into dataflow-style architectures optimized for FPGA deployment. Subsequently, a bitstream is synthesized and deployed onto the TySOM-3A-ZU19EG prototyping board. Operating at 100 MHz with a batch size of 10, we evaluated each quantized model ranging from 8-bit to 2-bit precision and targeting a real-time throughput of 60 frames

per second (FPS), this analysis focused on the utilization of FPGA resources, results are detailed in table I.

V. CONCLUSION

In this work, we propose several tiny-YOLOv3 FPGA implementations specialized for event data, with different quantization levels. It appears that the best compromise between performance and resource utilization – and therefore inference time – is 4-bit-quantization for this model. Indeed, the loss in mAP@50 is negligible and FINN can still optimize the DSP utilization, resulting in relatively low FPGA footprint.

REFERENCES

- [1] G. Gallego *et al.*, “Event-based vision: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 154–180, Jan. 2022, doi: 10.1109/TPAMI.2020.3008413.
- [2] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” Apr. 2018, Available: <https://arxiv.org/abs/1804.02767v1>
- [3] P. Adarsh, P. Rathi, and M. Kumar, “YOLO v3-tiny: Object detection and recognition using one stage improved model,” *2020 6th International Conference on Advanced Computing and Communication Systems, ICACCS 2020*, pp. 687–694, Mar. 2020, doi: 10.1109/ICACCS48705.2020.9074315.
- [4] Z. Zhou, Z. Wu, R. Bouteau, F. Yang, C. Demonceaux, and D. Ginjac, “RGB-event fusion for moving object detection in autonomous driving,” *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7808–7815, Mar. 2023, Available: <https://github.com/ZZY-Zhou/RENet>.
- [5] “DPU for convolutional neural network.” Available: <https://www.amd.com/en/products/adaptive-socs-and-fpgas/intellectual-property/dpu.html>
- [6] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (VOC) challenge,” *International Journal of Computer Vision*, vol. 88, pp. 303–338, Jun. 2010, doi: 10.1007/S11263-009-0275-4/METRCS.
- [7] M. Gehrig, W. Aarents, D. Gehrig, and D. Scaramuzza, “DSEC: A stereo event camera dataset for driving scenarios,” *IEEE Robotics and Automation Letters*, vol. 6, pp. 4947–4954, Jul. 2021, doi: 10.1109/LRA.2021.3068942.
- [8] Y. Umuroglu *et al.*, “FINN: A framework for fast, scalable binarized neural network inference,” Dec. 2016, doi: 10.1145/3020078.3021744.

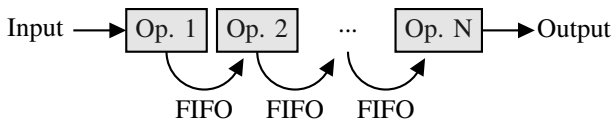


Fig. 3: FINN architecture overview

Design Exploration of RISC-V Soft-Cores through Speculative High-Level Synthesis

Thomas Feuilletin¹, Dylan Leothaud¹, Jean-Michel Gorius¹, Simon Rokicki¹ and Steven Derrien²

¹Univ Rennes, Inria, CNRS, IRISA

²Université de Bretagne Occidentale

Abstract

The RISC-V ecosystem is quickly growing and has gained a lot of traction in the FPGA community, as it permits free customization of both ISA and micro-architectural features. However, the design of the corresponding micro-architecture is costly and error-prone. We address this issue by providing a flow capable of automatically synthesizing pipelined micro-architectures directly from an Instruction Set Simulator in C/C++. Our flow is based on HLS technology and bridges part of the gap between Instruction Set Processor design flows and High-Level Synthesis tools by taking advantage of speculative loop pipelining. Our results show that our flow is general enough to support a variety of ISA and micro-architectural extensions, and is capable of producing circuits that are competitive with manually designed cores.

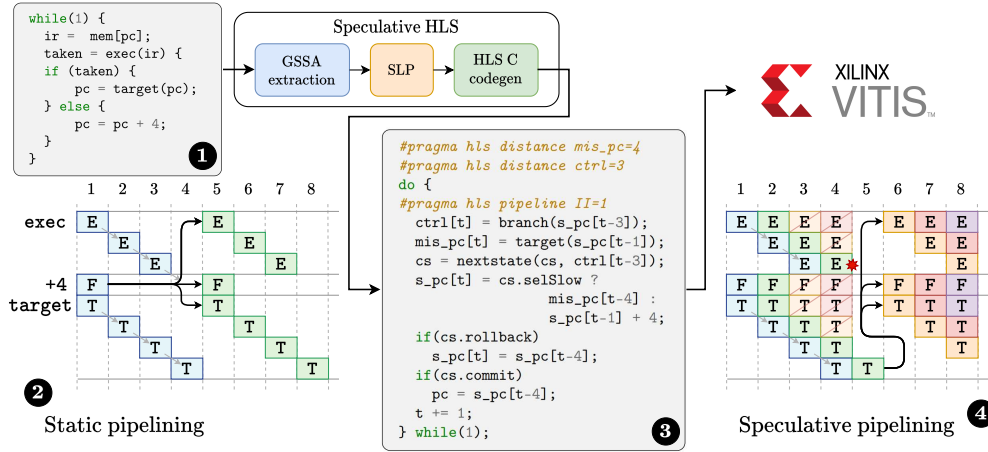


Figure 1: Our SLP source-to-source transformation flow. The toolchain takes C code ① as an input and produces transformed C code ③. ② and ④ show the respective schedules of the input and output code.

Speculative HLS of RISC-V Cores

The RISC-V ecosystem is quickly growing and has gained a lot of traction in the FPGA community, as it permits free customization of both the ISA and the micro-architecture.

Retargeting a compiler to a new ISA is a widely studied problem, but automatically synthesizing the corresponding instruction set micro-architecture has received less attention. Existing tools and technique offer significant room for improvement: they either lack generality [1, 2] or operate from low-level structural models that are not fundamentally different from RTL specifications.

In the meantime, High-Level Synthesis (HLS) technology, which compiles C and C++ code directly to hardware circuits, has continuously improved. For example, several recent research results have shown how High-Level-Synthesis techniques could be extended to synthesize efficient speculative hardware structures [3,

4]. In particular, Speculative Loop Pipelining (SLP) appears as a promising approach as it can handle both control-flow and memory speculations within a classical HLS framework [5].

Our work bridges part of the gap between Instruction Set Processor design flows and High-Level Synthesis tools. We show how to take advantage of SLP to automatically synthesize in-order pipelined micro-architectures from Instruction Set Simulator (ISS) models written in C, focusing on the RISC-V ISA. Our contributions are the following:

- We show how SLP can serve as a foundation to perform fully automatic micro-architectural synthesis from a behavioral description of a processor, in the form of an ISS. We extend SLP to support the synthesis of in-order pipelined CPU micro-architectures and their hazard recovery logic.
- We evaluate our approach in terms of supported features (both from an ISA and micro-architectural perspective) and quality of results

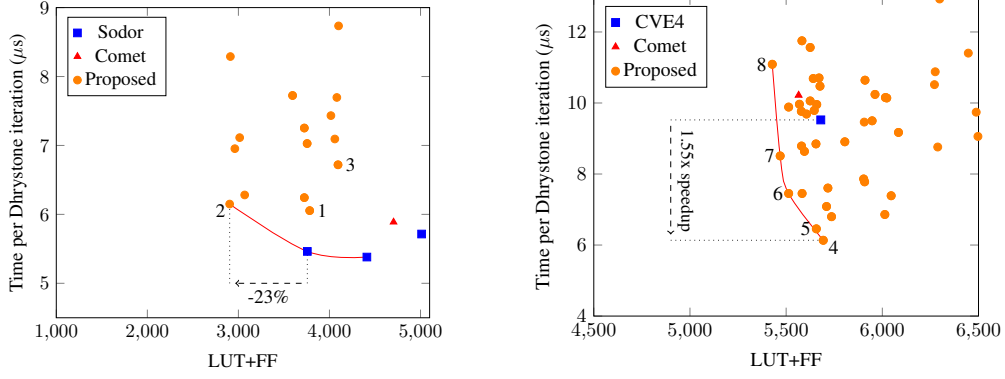


Figure 2: Area/performance result set for 21 variants of RV32I core (left) and 105 variants of RV32IM core (right) synthesized through our approach. Results for Sodor, Comet and CVE4 are reported on the figure as baselines.

(performance and area). Our results show that our flow can handle complex mechanisms like branch prediction and hardware Control-Flow Integrity while providing QoR similar to manual designs.

Our source-to-source transformation flow, depicted in Figure 1, accepts C code as input and produces speculatively pipelined C code targeting an HLS toolchain. The latter can then be compiled/synthesized to obtain an RTL-level description of the processor core.

The key idea in SLP resides in rescheduling critical operations to extend their dependence distance before calling the HLS tool. The HLS static pipelining pass will harness this additional schedule slack to produce more aggressive (i.e., deeper) pipelined schedules with higher clock speeds. The output C code shown in Part ③ of Figure 1 is produced from the input C code in Part ①. Its corresponding execution trace is provided in Part ④.

Results

To demonstrate that our proposed approach can generate competitive pipelined micro-architectures, we generate a large set of processors by exploring different speculation setups: no speculation on the register file, pipeline interlocking, or forwarding. We also modify the latency of the different operational blocks used in the ISS to explore several different pipeline depths. As baselines, we also synthesize the three Sodor pipelined cores (2-, 3-, and 5-stage pipelines) and the CV32E40P core [6]. We synthesize two configurations of the Comet processor [7], RV32I and RV32IM. As our generated cores do not implement RISC-V CSR registers, we remove the CSR unit from the Sodor and CVE4 cores.

Our experiments target an Artix7 XC7A200TISBG-1L and use Vitis HLS 2021.2 as the HLS backend. Performance results were obtained by executing the Dhrystone benchmark, compiled using `newlibc`.

Results of the automatic design space exploration are provided in Figure 2. The leftmost part represents the results obtained for the RV32I ISA, and the

rightmost part represents the results obtained with the RV32IM ISA. The generated micro-architectures are slower than the Sodor and Comet baselines for the RV32I ISA, while we are able to generate faster cores for the RV32IM target (55% faster than CVE4). Our generic approach generates extra control logic on the critical path of the RV32I cores, reducing their maximal frequency. On the other hand, the critical path of RV32IM cores is located in the multiplication/division unit. The extra logic that hinders the RV32I performance could be optimized during the SLP transformation, but this improvement is left for future work.

References

- [1] Gai Liu, Joseph Primmer, and Zhiru Zhang. “Rapid generation of high-quality RISC-V processors from functional instruction set specifications”. In: *2019 56th ACM/IEEE Design Automation Conference (DAC)*. IEEE. 2019, pp. 1–6.
- [2] Peter Yiannacouras, Jonathan Rose, and J Gregory Steffan. “The microarchitecture of FPGA-based soft processors”. In: *Proceedings of the 2005 international conference on Compilers, architectures and synthesis for embedded systems*. 2005, pp. 202–212.
- [3] Steven Derrien et al. “Toward Speculative Loop Pipelining for High-Level Synthesis”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39.11 (2020), pp. 4229–4239.
- [4] Lana Josipović, Andrea Guerrieri, and Paolo Ienne. “Speculative Dataflow Circuits”. In: *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. FPGA ’19. Seaside, CA, USA: Association for Computing Machinery, 2019, pp. 162–171. ISBN: 9781450361378. DOI: 10.1145/3289602.3293914.
- [5] Jean-Michel Gorius, Simon Rokicki, and Steven Derrien. “SpecHLS: Speculative Accelerator Design Using High-Level Synthesis”. In: *IEEE Micro* 42.5 (2022), pp. 99–107. DOI: 10.1109/MM.2022.3188136.
- [6] Andreas Traber et al. “PULPino: A small single-core RISC-V SoC”. In: *3rd RISC-V Workshop*. 2016.
- [7] Simon Rokicki et al. “What You Simulate Is What You Synthesize: Design of a RISC-V Core from C++ Specifications”. In: *RISC-V Workshop 2019*. 2019, pp. 1–2.