RICE UNIVERSITY

Hardware Security Primitives for Resource-Constrained Devices

By

Dai Li

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Doctor of Philosophy

APPROVED, THESIS COMMITTEE

Kaiyuan Yang
Assistant Professor of Electrical and Computer Engineering

Taiyun Chi
Assistant Professor of Electrical and Computer Engineering

Joseph Cavallaro
Professor of Electrical and Computer Engineering

ang chen
ang chen (Aug 12, 2021 16:12 CDT)
Ang Chen
Assistant Professor of Computer Science

HOUSTON, TEXAS

August 2021

RICE UNIVERSITY

Hardware Security Primitives for Resource-Constrained Devices

by

Dai Li

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

Approved, Thesis Committee:

Kaiyuan Yang, Chair
Assistant Professor of Electrical and
Computer Engineering

Joseph Cavallaro
Professor of Electrical and Computer
Engineering

Taiyun Chi
Assistant Professor of Electrical and
Computer Engineering

Ang Chen
Assistant Professor of Computer Science

Houston, Texas

December, 2020

ABSTRACT

Hardware Security Primitives for Resource-Constrained Devices

by

Dai Li

With the number of IoT devices surpassed global population, the shortage of energy, area and security of IoT presents a challenge to its application in wider scenarios. The issues of cyber security, information and privacy have been critical to the involvement of edge devices to industry, finance and personal life.

Modern edge devices experience various attacks from different dimensions. In the physical domain, reverse-engineering, micro-probing and optical-reading are widely used techniques to hack IC structure and data information. Trojan injection, side-channel analysis, web attacks are some popular ways to carry out non-invasive attack.

For resource-constrained devices, these attacks are especially efficient. Countermeasures of pure software, FPGA and ASIC solutions have been implemented in conventional electronic devices. There are mature solutions such as TPM and TEE deployed by Intel, ARM and other vendors to protect devices from certain attacks.

Traditional hardware security solutions mainly focus on cryptography related protection schemes. The required key usually comes from a random number generator (RNG). RNG-based key generation and storage scheme has shown drawbacks including high fabrication cost and insecure storage.Physically unclonable function (PUF) has emerged as an alternative. But its stability prevents its further usage.

Cryptography algorithms like RSA and ECC are utilized to keep the confidentiality

and integrity of the chip. However, they come with issues of high area and energy budget for resource-constrained devices when implemented in hardware to achieve reasonable throughput. Some algorithms such as RSA can be cracked by quantum computers easily. Post-quantum cryptography (PQC) has become a hot topic to provide security solutions in new era. But the few ASIC works have shown excessive area, energy and latency cost, which is the bottleneck for real-world applications.

Another security-related issue is network intrusion detection system (NIDS), which is normally executed in software and the workload is beyond the capability of most IoT devices. Attempts have been made to design custom hardware for cloud-based NIDS. However, existing solutions do not fit into energy-constraint devices.

It is an area of little exploration to develop hardware security solutions for resource-constraint devices. To provide better security for edge devices with affordable cost and robustness against prevalent attacks and quantum computers, we designed memory-centric hardware primitives to accelerate the root and chain of trust of miniaturized area and energy. Three major projects were implemented to realize this concept.

First, a 562-feature-square self-regulated physically unclonable function provided state-of-the-art native stability as well as energy and area for secure key generation.

Second, an 8-T CAM based network intrusion detection system performed signature-based intrusion detection for distributed IoT devices with 1.54-fJ/Byte/Search efficiency. The automata engine with system-level and circuit-level co-design presented the first silicon solution for IoT hardware firewall.

Third, a processing-in-memory accelerator for PQC was implemented to provide compact and low-power engine for PQC computation. A range-matching CAM-based cumulative-distribution-table (CDT) sampler was implemented to achieve 20.6pJ/sample energy efficiency and 85.9M sample/s throughput. A 6-T SRAM-based near-memory accelerator for number theoretical transformation was implemented for ultra-compact single-bank NTT operation for PQC. The use of in- and near-memory computation achieved area and energy efficiency compatible with low-power IoT applications.

Acknowledgements

First and foremost, I would like to express my greatest gratitude to my research advisor Professor Kaiyuan Yang. He helped me a lot not only on technical skills, but also on the concept and approach of critical thinking, problem handling and project execution. He also taught me the valuable knowledge of paper writing and presentation. The dissertation would not be possible without Kaiyuan's support, guidance and patience over my studies at Rice University.

I would also like to thank Professor Joseph Cavallaro, Professor Taiyun Chi and Professor Ang Chen for serving as my thesis committee member. I learned a lot from their courses in Rice. I am thankful to all the members in Secure and Intelligent Micro-Systems (SIMS) lab in Rice University. I am thankful to Yan He, Zhanghao Yu, Zhiyu Chen, Liwen Jiang and Akhil Pakala for their help and cooperation during my PhD study. I am also thankful to all members of Rice Integrated Systems and Circuits (RISC) lab including Professor Aydin Babakhani, Xuebei Yang, Peiyu Chen, Mahdi Assefzadeh, Yuxiang Sun, Babak Jamali, Hamed Rahmani, Yaswanth Cherivirala, Seyed Kazempour, Mehdi Forghani, Mostafa Mosseini, Sam Razazi and Yash Mehta. I would like to thank Martin Li and David Genzer during my intern at Biotronik and Chao Jiao and Jingyi Zhou during my intern at Cadence.

I am sincerely grateful for the professors and administrative staff of the Department of Electrical and Computer Engineering at Rice University for their help and support.

Finally I would like to thank my parents and wife for their love and patience.

Contents

Abstract ii

List of Illustrations viii

List of Tables xiii

1 Introduction 1

1.1 Cross-layer Security in Hardware and Software 6

1.1.1 Basics Building Blocks of Cryptography and Hardware Security 6

1.1.2 Trusted Platform Module (TPM) 8

1.1.3 Trusted Execution Environment (TEE) 10

1.2 Attack Threats and Challenges to Existing Hardware Security Design 11

1.2.1 Security against Physical Attacks 12

1.2.2 Drawbacks of Traditional ASIC 15

1.2.3 The Demand for Post-quantum Crypotgraphy 16

1.3 Thesis Statement and Proposed Memory-centric Hardware Security for Resource-constraint Devices 18

1.4 Outline and Contribution of the Thesis 19

2 Reconfigurable Physically Unclonable Function 23

2.1 Motivation and Related Work 24

2.2 Proposed PUF Cell Design 28

2.2.1 Subthreshold Inverter Based PUF Cell 29

2.2.2 Voltage Regulation using Native Transistors 30

2.2.3 System Implementation 33

2.3 Zero-Overhead Reconfiguration 34

2.3.1 Source of Instability 35
2.3.2 In-Cell Reconfiguration 36
2.3.3 Physical Implementation of Reconfiguration Scheme 39
2.3.4 Fast R-Map Searching via Body Bias Emulation 40
2.4 Measurement Results 41
2.4.1 Voltage Regulation 42
2.4.2 Native PUF Stability 42
2.4.3 PUF Stability Over Voltage and Temperature Variations . . . 43
2.4.4 Uniqueness and Randomness 46
2.4.5 Aging Effects 48
2.4.6 Throughput and Energy Efficiency 49
2.4.7 Related Works and Comparison Table 50
2.5 Summary 51

3 8-T Dual-Port CAM-Based Network Intrusion Detection System 53

3.1 Motivation and Related Work 53
3.2 Reconfigurable Three-Phase NIDS Architecture 56
3.3 Circuit Design And Implementation 60
3.4 Measurements 64
3.5 Summary 67

4 MeNTT: A Compact and Efficient In-Memory Number Theoretic Transform Accelerator 69

4.1 Motivation 69
4.2 Background and Related Work 72
4.2.1 Ring Learning with Errors (LWE) 72
4.2.2 Number Theoretic Transform (NTT) 73

4.2.3 Processing in Memory for Cryptographic Acceleration 75
4.3 Data Storage and Arithmetic Flow in MeNTT 76
4.4 Proposed Modular Addition/Subtraction in Memory 79
4.5 Proposed Bit-Serial Modular Multiplication in Memory 81
4.5.1 NTT snd INTT Dataflow 84
4.6 Evaluation and Discussion 87
4.6.1 Comparisons with software solutions 88
4.6.2 Comparison with FPGA solutions 89
4.6.3 Comparison with ASIC solutions 91
4.6.4 Comparisons with PIM solutions 92
4.7 Summary 92

5 MePLER: A 20.6-pJ Side-Channel-Aware In-Memory CDT Sampler 96

5.1 Motivation and Related Work 97
5.2 Range-Matching-CAM 101
5.3 Segmented Search 105
5.4 Configurable Search Range and Precision 108
5.5 Row-Masking for Robustness against Side-Channel Attacks 110
5.6 Measurements 111
5.7 Summary 112

6 Conclusion 117

Bibliography 120

Illustrations

1.1 Estimated market of IoT device in recent future. 2
1.2 Number breakdown of devices affected by Mirai [1] 3
1.3 Software cost for popular cryptography algorithms. 3
1.4 FPGA performance and cost for popular cryptography algorithms. . . 4
1.5 ASIC performance and cost for popular cryptography algorithms. . . 4
1.6 Recent trends in security implementations. 5
1.7 Block diagram of a TPM and hardware part of a TEE. 9
1.8 Software and hardware stack of a TEE [2]. 10
1.9 Examples of commonly seen physical attacks [3] [4] [5] [6] [7] 12
1.10 Comparisons of hardware accelerators for cryptography with software approach. 16
1.11 Impact of quantum computer on prevalent cryptography. 17
1.12 Scope of this work. 20

2.1 Typical PUF diagram from entropy extraction to key generation. . . 26
2.2 (a) PUF cell topology, comparison of 2-transistor amplifier-based cell and proposed subthreshold inverter-based cell, (b) simulated histogram of voltages at each stage, (c) voltage gain of 2-transistor amplifier and subthreshold inverter in different technology nodes. . . 30
2.3 Comparison of (a) gain and (b) voltage histogram between 2-transistor amplifier and subthreshold inverter. 31

2.4 (a) Cell-wise voltage regulation model using a native transistor, (b) column-wise voltage regulation model using a native transistor. 32
2.5 PUF array (a) block diagram, (b) readout circuits schematic, (c) readout timing. 34
2.6 (a) Histogram of switching voltage (V_M) difference from 10000 Monte-Carlos simulations, (b) from 96 unstable cases. 36
2.7 (a) Process of in-cell reconfiguration of an unstable cell, (b) the switching voltages of an originally unstable cell before and after reconfiguration. 37
2.8 Reconfigurable PUF cell in (a) layout, (b) schematic views. 38
2.9 Search and enrollment process of R-MAP configuration. 40
2.10 (a) PUF chip micrograph, (b) cell layout. 42
2.11 VV_{DD} curve versus (a) supply voltage, (b) temperature under different bias voltages. 43
2.12 (a) Bit error rate, (b) percentage of unstable bits versus number of evaluations under nominal condition. 44
2.13 (a) Native BER, (b) unstable bits histogram of 10 chips. 45
2.14 (a) Bit error rate versus temperature variation under different stabilization methods, (b) BER and flipping bits versus supply voltage. 46
2.15 (a) Detection rate, (b) BER improvement versus body bias sweep. . 47
2.16 Normalized Hamming Distance before and after reconfiguration. . . 48
2.17 Autocorrelation of 40960 bits for up to 4000 lags. 49
2.18 Aging effects on BER and percentage of flipping bits. 49
2.19 Throughput and energy efficiency versus VVDD in nominal condition. 50

3.1 (a) A Snort rule example, (b) Performance comparison of proposed work with prior arts. 54
3.2 Proposed 3-Phase pattern matching workflow with an example. . . . 55

3.3 Block diagram of proposed network intrusion detection system (NIDS) and processing element. 57
3.4 (a) An example of Aho-Corasick algorithm Trie, (b) ClamAV and Snort selected sub-ruleset transition and state depth statistics. 58
3.5 (a) State and transition of conventional and pipelined AC algorithm on an example, (b) Simulated energy efficiency versus number of rules using different approach. 60
3.6 Proposed pipelined range-matching architecture. 61
3.7 (a) Proposed 8-T CAM Cell, (b) example search illustration, search "101" at port A, column 0 stores "101", column 1 stores "011". . . . 62
3.8 (a) Encoding from ASCII code to fixed-1s code, (b) single side search example, (c) masking example under single side search. 63
3.9 2-level range decoder and hierarchical clock gating. 64
3.10 (a) Chip micrograph, (b) Power breakdown at full workload. 65
3.11 (a) System power and search energy versus supply voltage(b) Shmoo plot of the system. 65
3.12 (a) Normalized search energy versus ratio of malicious patterns (hit rate) with and without clock gating, (b) Normalized search energy versus number of stages in Phase-1 under different attack rate. 66

4.1 The basic scheme of lattice-based cryptography. 72
4.2 The overall flow of Ring-LWE and polynomial multiplication in Ring-LWE. 73
4.3 Basic bit-wise logic operation in 6T SRAM by accessing two rows simultaneously. 76
4.4 Diagrams of MeNTT and custom circuitry for the key peripherals. . . 78
4.5 Data arrangement and operation sequence in an NTT round. 78
4.6 Bit-serial comparator operation in column peripheral. 79

4.7 Illustration of in-memory two-bit modular addition. 80
4.8 Illustration of in-memory 2-bit modular multiplication (q is 2'b10, Tag is 1 for two cycles). 83
4.9 Comparison of traditional modular multiplication methods and proposed method. 84
4.10 Proposed MeNTT dataflow enabled by a new mapping strategy. . . . 86
4.11 NTT energy versus different polynomial order at bitwidth of 14. . . . 88
4.12 NTT cycle count versus different polynomial order at bitwidth of 14. 89
4.13 NTT energy versus different bitwidth at polynomial order of 1024. . . 90
4.14 NTT cycle count versus different bitwidth at polynomial order of 1024. 90

5.1 Applications of random sampling in cryptography and machine learning. 97
5.2 Energy break down in the execution of NewHope. 97
5.3 Comparisons of sampling methods. 98
5.4 Illustration of a rejection sampler. 99
5.5 Illustration of a CDT sampler. 101
5.6 Block diagram of proposed range-matching-cam array and schematic of CAM cell. 103
5.7 Range-matching process in the proposed MePLER. 105
5.8 Proposed segmented range-search MePLER. 106
5.9 2x2 layout of the differential MePLER cell. 107
5.10 Optimization of power consumption with respect to bitwidth selection in segments. 108
5.11 Row-wise and column-wise power-gating for configurable search range. 109
5.12 Random masking of rows for defence against side-channel attack. . . 110
5.13 Flow of differential power analysis. 112
5.14 Micrograph of the designed MePLER chip. 113

5.15 Sampler distribution from a Gaussian distribution versus ideal distribution. 113
5.16 Sampler distribution from a Gamma distribution versus ideal distribution. 114
5.17 System shmoo plot of MePLER. 114
5.18 DPA accuracy and energy versus number of available rows for masking. 115

Tables

2.1 PROBABILISTIC MODEL FOR RECONFIGURABLE PUF CELL . 35
2.2 COMPARISON OF STABILIZATION METHODS for PUF 39
2.3 NIST PUB 800-90B RESULTS 50
2.4 NIST PUB 800-22 RESULTS 51
2.5 COMPARISON OF SELF-REGULATED PUF WITH PRIOR PUF ARTS 52

3.1 COMPARISON TABLE OF THE OUR CAM-BASED NIDS WITH PRIOR NIDS AND CAM WORKS 68

4.1 COMPARISON TABLE OF MENTT WITH PREVIOUS SOFTWARE, FPGA, ASIC AND PIM IMPLEMENTATIONS OF NTT ACCELERATOR. 95

5.1 COMPARISON OF MEPLER WITH PREVIOUS HARDWARE SAMPLER WORKS. 116

Chapter 1

Introduction

As the technology and cost of semiconductor device scale down, Internet of Things has seen wider use in an increasing market [8]. It is envisioned that 22 billion IP devices with ubiquitous usage will be connected in 2025 (Figure. 1.1) [9]. The application of ubiquitous devices include smart home, industry monitoring, medical implant and so on. They are typically used to sense, collect, process and transmit certain domain data. These devices are featured with large numbers, compact size, limited battery life,lightweight processor and small memory.

Due to the constraint of resources, the way these devices manage the storage and transmission of privacy data lacks sufficient security. And this is posing new challenges and requirements for the cybersecurity and privacy to the academia and industry. Many organizations have realized the necessity for emphasis of security in IoT devices. For example, NIST has claimed three high-level considerations that may affect the management of cybersecurity and privacy risks for IoT devices compared to conventional information technology (IT) devices [10].

• **Many IoT devices interact with the physical world in ways conventional IT devices usually do not.**

• **Many IoT devices cannot be accessed, managed, or monitored in the same ways conventional IT devices can.**

• **The availability, efficiency, and effectiveness of cybersecurity and privacy capabilities are often different for IoT devices than conventional IT**

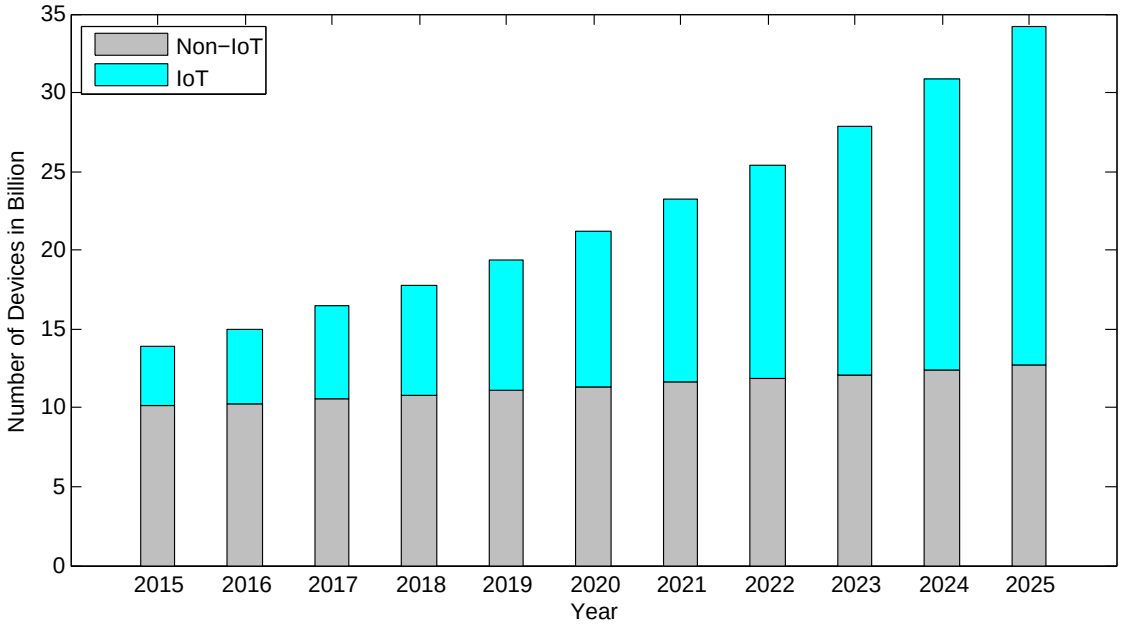


Figure 1.1 : Estimated market of IoT device in recent future.

.

devices.

Take a botnet called Mirai for example (Fig. 1.2), it affected over 2 million devices and took down many important networks using DDoS attack, such as the internet of the entire Liberia, and the German internet giant Duetch telekom, causing huge amount of losses. An analysis show that the major victims of it are IoT devices like routers and cameras. It is mainly because they do not protect the authentication well and lack enough anti-attack measures like firewalls.

Cybersecurity has been a hot topic for decades. Many organizations and companies have proposed solutions to guarantee the security of IoT device or embedded device. NIST published a guideline on IoT security for the essential capabilities [11]. It covers both hardware and software domain. And many desired capabilities are highly related to identification, authentication and cryptography.

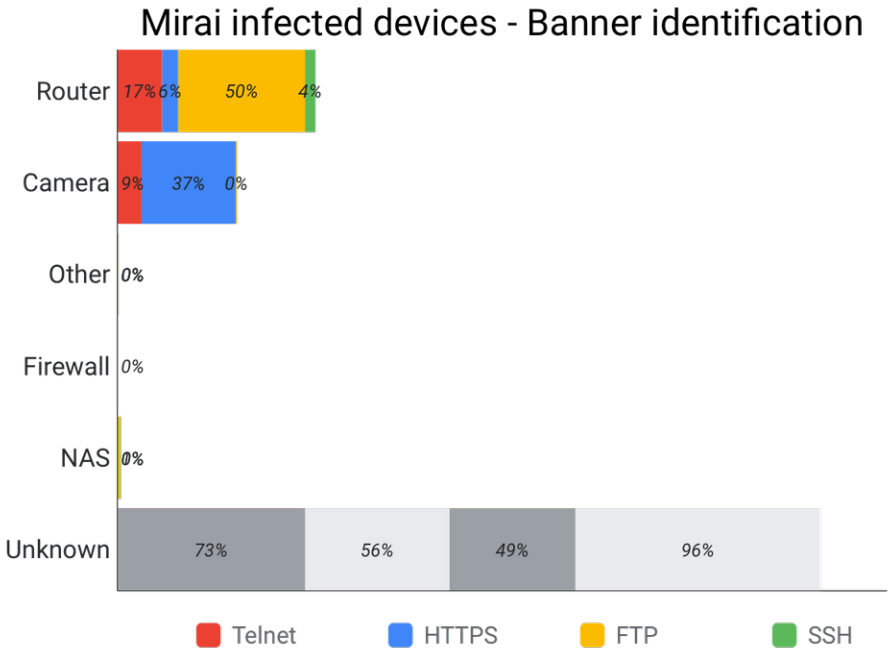


Figure 1.2 : Number breakdown of devices affected by Mirai [1]

Algorithm	Number of Cycles	Energy	Processor
AES-128 [1]	6358	64.2nJ	ARM-M0+
ECC (ECSM 256bit) [2]	13.6×10^8	8640uJ	RISC-V
Ring-LWE (NewHope-1024 Enc) [3]	1.9×10^6	1207uJ	ARM-M4

Figure 1.3 : Software cost for popular cryptography algorithms.

Traditional solutions focus on the software level and have developed mature solutions and applications ranging from firewall, anti-virus software, authentication and cryptography. However, the software latency for some algorithms and firewalls are too high for resource-constraint devices, as can be shown in Fig. 1.3. The main reasons for this include large bitwidth for data in cryptography, complicated arithmetic operation and limitations of computing and memory capability in most embedded devices.

Accelerator	Platform	Area/Slices	Area/BRAMS	Throughput/Mbps	Baseline Software Throughput/Mbps
AES	XC2VP20-7	9446	0	21540	0.97
SHA	XCV2P30	764	1	785	5.09
ECC	Virtex-5	3140	0	0.11	3e-5

Figure 1.4 : FPGA performance and cost for popular cryptography algorithms.

Accelerator	Technology/nm	Throughput/Mbps	Baseline Throughput/Mbps	Energy	Baseline Energy
AES	45	53000	0.97	0.38nJ	64.2nJ
SHA	14	174592	5.09	15.8nJ	8640uJ
ECC	65	0.02	3e-5	6.34uJ	1207uJ

Figure 1.5 : ASIC performance and cost for popular cryptography algorithms.

One way to improve this is to use hardware accelerators. FPGA is a popular choice to improve the throughput. Many works have been done to design FPGA accelerators for cryptography algorithms. From the table shown here, one can observe the significant elevation in throughput. But FPGA is normally power-hungry and bulky (Fig. 1.4), and not the best choice for edge devices. That leads to the ASIC accelerator integrated in many solutions (Fig. 1.5). We listed the throughput and energy of 3 ASIC implementations and compare them with software baseline. The improvement in throughput is at least 70 times. And Improvement in energy efficiency is at least 200 times. Recent industry giants proposed various solutions to address the security problem of IT devices (Figure. 1.6). Both innovative hardware and software approaches are deployed to create platforms for trusted computing. And hardware security is drawing more and more attention because it provides higher level of efficiency, privacy and safety than software-only approaches. The solutions usually contains hardware primitives including secure key, cryptography engine and other modules serving as root of trust and chain of trust. But as pointed out by NIST, IoT

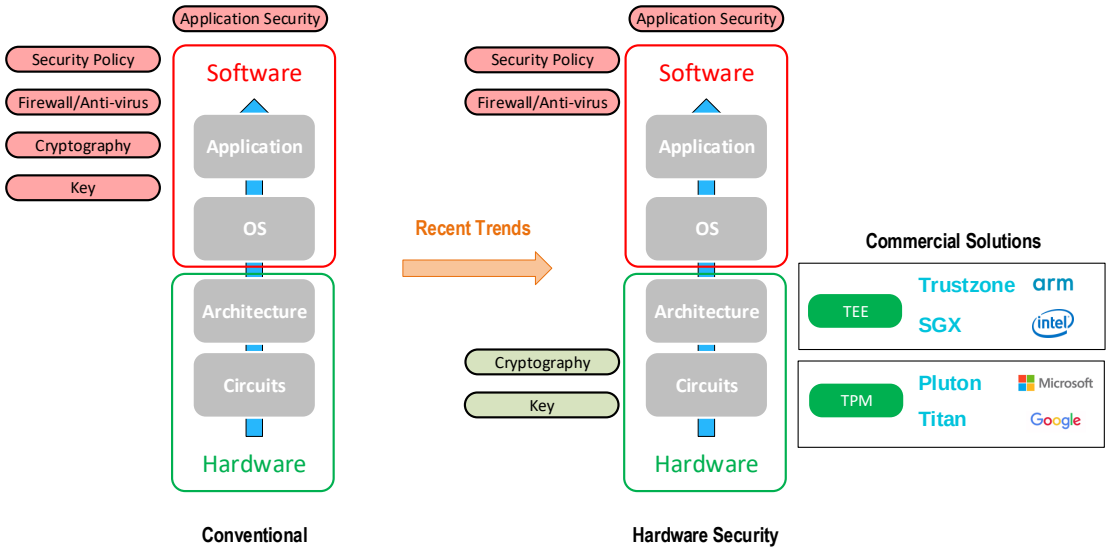


Figure 1.6 : Recent trends in security implementations.

device, especially resource-constrained devices, experience severer security threats. Traditional approach is not necessarily the best option for resource-constrained devices, or is beyond the capability of devices. And this is a major obstacle for the deeper involvement of such devices into daily business and personal life.

There are various attacks that have been proven efficient against those devices. The hardware security modules in traditional solutions is vulnerable as more and more devices are physically accessible to the public. Area efficiency is also a potential problem for ASIC-based accelerators. There are also challenges from recent development of quantum computer that may be able to break prevalent cryptography in minutes.

In this thesis, we propose to address issues of hardware security for ubiquitous devices by innovative circuits and primitives for secure keys, cryptography engines and firewalls.

1.1 Cross-layer Security in Hardware and Software

There are many commercial solutions to provide trusted computing for the sake of security. They usually combine hardware and software approaches. Among them, the 2 most famous ones are TPM and TEE. TPM is abstract for trusted platform module. Nowadays It is mostly a fully hardware solution on circuit level. There are products from Microsoft called Pluton and from Google called Titan. TEE is abstract for trusted execution environment, it is a hybrid approach of circuit, architecture and software. There are commercial products from ARM called trustzone and from intel called SGX. They are designed for general purpose security, and IoT devices are also using them to provide security.

1.1.1 Basics Building Blocks of Cryptography and Hardware Security

Cryptography is the core idea and scheme for most prevalent security solutions . The idea of cryptography originated from ancient times. Its main application ranges from secure communication, authentication, e-commerce to crypto-currencies. The common approach is to use a known function to convert plain-text into cypher-text which is unreadable to eavesdroppers. This function is straightforward to compute from input to output, but difficult to infer the input even if one already has the output and function itself. The function is generally categorized as 'trapdoor' function. A key is usually required to encrypt and decrypt the text. .

Key Generation and Storage

In security, a secret key is needed in scenarios of identification, public and private key encryption and other applications. Depending on the usage, the keys can be generated temporarily by a random number generator (RNG), or programmed by

server and burnt in the Fuse at manufacturing time. One important requirements for the secret key is that the device should keep them while out of power which needs non-volatile memory. In IoT devices, NVM is relatively expensive. The conventional storage media for the secret key can be Fuse, EEPROM and Flash memory.

RNG can be categorized into true random number generator (TRNG) and pseudo random number generator (PRNG). TRNG works by collecting environmental noise as entropy source and amplify it into digital signals. PRNG make use of certain arithmetic functions to generate numbers that satisfy randomness standards. Typically TRNG is preferred in scenarios for high security since it generate 'real' random numbers which contains more information or entropy. There are also applications where PRNG is more suitable because of its higher throughput than TRNG.

Public and Private Key Cryptography

Cryptography can be traditionally divided into public key and private key schemes. People can also refer them to asymmetric key encryption and symmetric key encryption.

In public key cryptography, the user generate a pair of keys. He sent one key out to the public as the public key and keep another one as the private key. In case where other users want to send a message to him, they can encrypt the message using the public key. The encrypted message can only be decrypted using the private key by the original user. This scheme ensures the safety of keys and is a fundamental scheme in modern cryptosystems. Many high-level protocols such as TLS and SSH make use public key cryptography. Some famous and widely deployed public key cryptography includes Rivest-Shamir-Adleman (RSA), Elliptic Curve Cryptography (ECC) and Diffie-Hellman (DH).

On the other side, private key cryptography, or symmetric key cryptography, relies on a shared secret key for data transfer. The encryption and decryption use the same key. The most famous and commonly used algorithm is Advanced Encryption Standard (AES). AES is a block cypher used in enormous applications such as BLE, WiFi, VPN and application-level Apps such as WhatsApp. Symmetric key cryptography generally provide higher throughput than asymmetric key schemes. It is preferred to encrypt large amount of data. But usually the sharing process of keys relies of public key cryptography.

Hashing

Hashing is another important aspect in cryptography. It is used to generate digital signature to provide proof of trust for hardware or software. Popular hashing functions include MD5, SHA1, SHA2 and SHA3. Symmetric key and asymmetric key algorithms can also be used to generate a hash.

1.1.2 Trusted Platform Module (TPM)

TPM is an international standard defined by Trusted Computing Group (TCG). TCG is an alliance formed by major IT companies like Intel, AMD, IBM, HP, Microsoft and Cisco. Its goal is to develop Trusted Computing. TPM is defined as a platform providing a variety of functions including but not limited to:

- A random number generator
- Secure cryptographic key generator
- A hashing of hardware and software configuration for remote attestation
- Binding and sealing of encryption data and TPM bind key and state

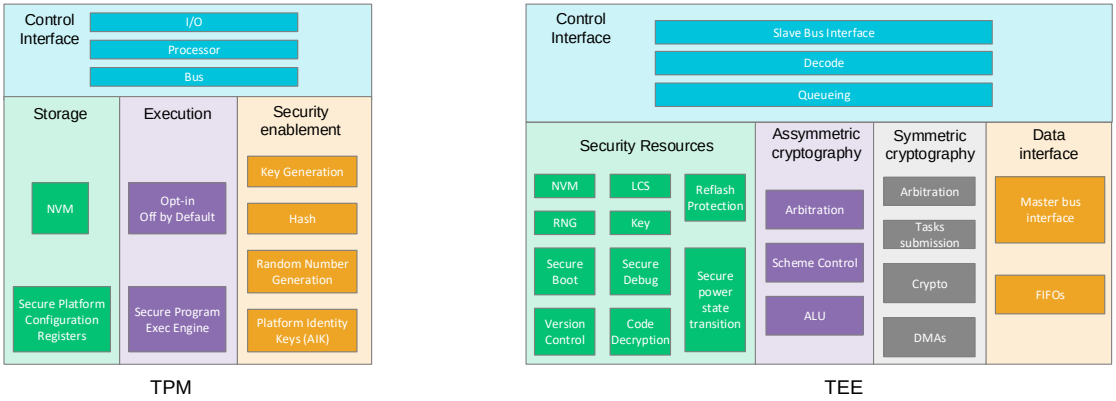


Figure 1.7 : Block diagram of a TPM and hardware part of a TEE.

The main target of TPM is to provide platform integrity, disk encryption, password protection and other security-related functions. By definition, TPM can have different kinds of implementations, including discrete TPM, integrated TPM, firmware TPM, hypervisor TPM and software TPM. In general, TPM is mostly implemented as a dedicated chip that performs the above functions. It is isolated from the main processor and memory, thus providing highly secure platform integrity and encryption.

TPM is successfully used in many devices ranging from cloud, desktop to mobile devices. Microsoft has required all Win10 computers to have a TPM. Meanwhile Microsoft has also developed a security engine called Pluton. In its recent Azure IoT platform, Pluton is implemented to serve as hardware root of trust and provide IoT security. Google also has its own TPM called Titan series. The main application is to enable integrity verification of hardware and firmware and tamper-evident logging in Google's purpose built servers. The main modules in Titan series are memories including Fuse, ROM SRAM and Flash, IO peripherals and crypto peripherals like AES, SHA, HMAC, TRNG and key manager. Nevertheless, Intel's Trusted Execution

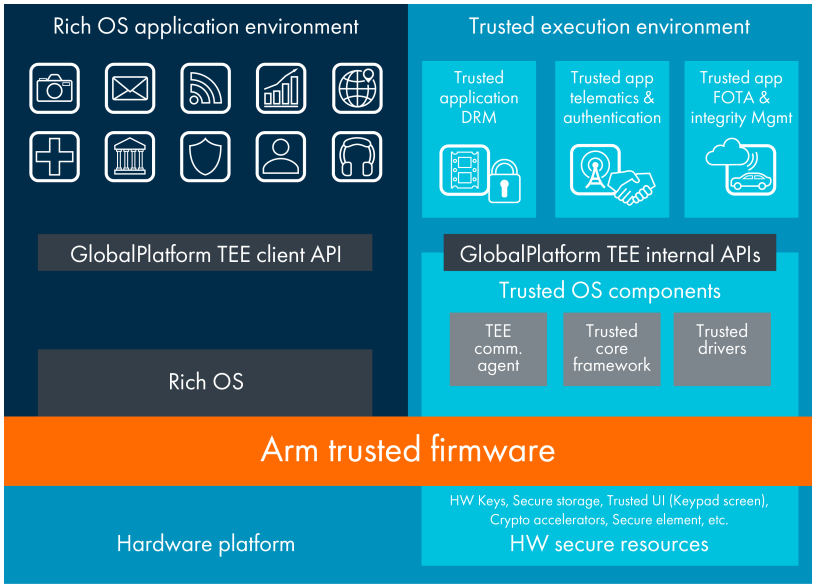


Figure 1.8 : Software and hardware stack of a TEE [2].

Technology also make use a TPM to defend against software attacks and provide a trusted computing environment. .

1.1.3 Trusted Execution Environment (TEE)

TEE is an isolated execution environment in the main processor and memory that provides a secure area for isolated execution, integrity of applications and confidentiality of application assets (Fig. 1.8).

Compared with TPM, TEE is a higher-level concept that provides a secure environment to run arbitrary applications. The target is to run applications in parallel with the normal operating system with higher security and access only from the trusted environment. To achieve this goal, a combination of custom hardware and software components are required to work closely. The aim of TEE is to guard the system from both hardware and software attacks. It usually runs inside a larger chip or SoC and is more flexible. Compared with TPM, TEE is part of the main processor

and requires less hardware cost.

ARM's TruseZone is an example of TEE. It has been integrated in many of ARM's processors. It utilizes a hybrid implementation of hardware and software to protect data. Intel SGX is another commercial TEE that is widely adopted.

Over time, TPM and TEE have been successfully accepted by industry as standard solutions for trusted computing. While concept and implementation may be different, TPM and TEE share some core modules in hardware, which are the key generation, storage, encryption and hashing modules. These hardware modules are cryptography modules which manages authentication, encryption and digital signature of various hardware and software applications.

1.2 Attack Threats and Challenges to Existing Hardware Security Design

Despite the fact that TPM, TEE and other custom secure elements provide decent guard against attacks for traditional devices, there are challenges that conventional approaches do not handle well on resource-constrained devices. Compared with traditional devices, IoT devices are more vulnerable to physical attacks because they are not protected well physically. And because of the large numbers, a single weakpoint in the network being attacked can cause loss to many users in the network. Traditional embedded device store key and other information in Flash, and flash can be attacked using optical approach. The EM emission information and power trace are easy to collect for IoT devices, which is vulnerable to side-channel attack. Discrete chip of TPM also suffers from probing attack because its connection to main processor is not protected. One can even replace the TPM with another broken module to attack the

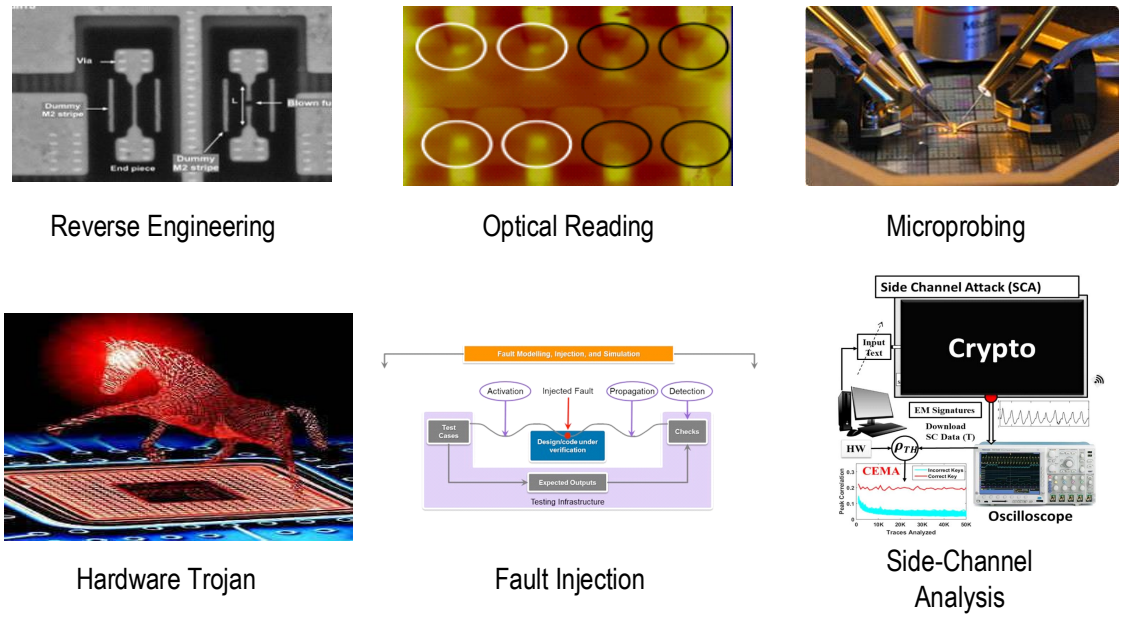


Figure 1.9 : Examples of commonly seen physical attacks [3] [4] [5] [6] [7]

main device. The emerging challenges drive research forward towards innovations for hardware primitives with higher security and affordable cost.

1.2.1 Security against Physical Attacks

Although software attacks are the most common type of attacks, there have been numerous physical attacks for IoT devices that are even more direct and efficient. For example, AES is a widely-used symmetric-key cryptography algorithm. The study for side-channel attacks on AES has been a popular topic for years. An attack method demonstrated by FoxIT made use of the EM emission during the execution of AES algorithm implemented in FPGA, and cracked the secret key in 5 minutes. Although the device setup for the crack cost only $ 200, the data encrypted by AES can worth much more than that. Software implementation of AES can also be broken by cache timing attack as demonstrated by Daniel Bernsitein from UIC.

Physical attacks take direct approaches to attack the device. Since most IoT devices are relatively more accessible than conventional IT devices, physical attacks are more feasible and dangerous in the era of IoT. Some known physical attacks (Figure 1.9) include:

- **Depackaging and Delayering**

By removing the package and etching the layout layer by layer, it is possible to detect and reconstruct the layout information from a chip and reverse-engineer the design. The layout information can be extracted from the metal layers, which include bus bitwidth, location of processor and memory and so on. Analog circuits are especially vulnerable to this kind of reverse engineering. It is widely used in many IC design companies to analyze products from competing vendors.

- **Optical Reading**

In non-volatile memories, data are stored as charges in the floating gate. This can be used by attackers to extract important information. Photon Emission Analysis was proposed to recover keys. In 2010, optical fault injection was proposed to attack Flash memory and get the stored data during the verification process. Unfortunately, flash memory is one of the most commonly used memory in resource-constraint devices. This make them very vulnerable to certain optical attacks.

- **Microprobing**

Microprobing is another invasive attack to directly fetch information from the device. Attackers can directly observe, manipulate and interfere with the circuits through the probe. It is usually combined with depackaging and delayering.

- **Side-Channel Attack**

Side-channel attack make use of side-channel signals such as current, EM emission and timing information to analyze the transient power and circuit activities, which

are usually correlated to the key information. A typical example is the side-channel attack on crypto modules whose execution in hardware has fixed patterns. Since the algorithm behind cryptography is well understood, attackers can analyze physical characteristics closely related to the data being processed. It is relatively easy and low-cost, but can take longer time to gather enough data for analysis.

- **Fault Injection**

Fault Injection Attack mostly refers to glitch attack. In this attack scenario, the supply voltage, clock signal or external electrical field are manipulated and change rapidly. The aim of the glitch attack is to manually force circuits to malfunction and corrupt data. Registers will likely to enter meta-stability and get a random data when influenced by the attack. It will change the state of important finite-state machines and cause the chip to work incorrectly, which may give attackers access to key information. Some research claim it as one of the most effective approach against smart cards.

- **Hardware Trojan**

The idea of hardware trojan is similar to that of a software one. It is injected at manufacturing or other procedures of the IC supply chain, embedded in the system and acts like a normal hardware. Without external trigger, it stays inactive and can not be detected. Some hardware trojans can even pass LVS check. But once it is triggered by certain actions or environment, it will steal information or launch attacks on other hosts. This is a powerful attack as it directly reside in the device and can hardly be detected before the attack.

Due to the direct interaction with the physical world with less protection, physical attacks on resource-constrained devices has become a major concern. Especially as the number of devices goes up dramatically, any weak point in a certain device can

cause influential damage to the owner.

In most embedded devices with TPM or TEE, choices of key storage is limited to unprotected Flash, EEPROM and Fuse. Usually these NVMs can be accessed and attacked without destroying the functionality of the chip or device. By reading from the NVMs, keys and other data can be directly fetched and cause huge loss.

The exposure of IoT devices make it easy for side-channel attackers to gather physical characteristics such as power trace, EM emission and timing information while executing certain programs. The problem is worse in TEE than TPM because TEE executes on the main processor which may not be designed specifically secure against side-channel attacks for a certain algorithm or procedure.

TPM is usually a standalone chip connected to the main processor through SPI or LPC bus. This poses two major risks to it.

First, the bus connection on the PCB is completely unprotected and can be probed and manipulated easily by external sources. This gives control of accurate and detailed transient signals to the attackers, which contains rich information. The attacker can also do fault injection easily by over-driving the bus signals.

Second, the binding of TPM to the main processor can be broken and altered with a cracked module. Since the connection between two chips completely lies on the PCB, it is possible to alter the original TPM with a new one which fakes the original one and aids attackers' operations.

1.2.2 Drawbacks of Traditional ASIC

The high cost of software implementation for cryptography leads to the exploration of hardware accelerators. The crypto engines implemented in hardware in TPM and TEE greatly boosted the performance of cryptography algorithms with lower energy

Accelerator	Technology/nm	Area/mm²	Throughput/Mbps	Baseline Software Throughput/Mbps
AES	45	0.15	53000	0.97
SHA	14	0.007	174592	5.09
ECC	65	0.13	0.02	3e-5

Figure 1.10 : Comparisons of hardware accelerators for cryptography with software approach.

cost. Several state-of-the-art crypto hardware works achieved more than 100 times higher throughput and higher energy efficiency. However, they come with a price of larger area.

As can be observed from fig. 1.10, custom crypto accelerators gains high energy efficiency and throughput, but brings about substantial area overhead. As a comparison, a baseline ARM Cortex-M0+ processor occupies less than $0.2mm^2$ in 180nm technology. In resource-constrained devices, the manufacturing cost will increase linearly with chip area as more and more crypto engines are integrated to ensure the security as well as throughput and efficiency. Many applications also require the device dimension to be as small as possible. There will be a trade-off between cost, functionality and security, which will eventually harm the usage of secure ubiquitous devices.

1.2.3 The Demand for Post-quantum Crypotgraphy

Recent studies in quantum computer have been inspiring. Despite the huge gap between the prototype and a real usable quantum computer, researchers have gained significant advances in getting more Qubits in to the computer. In 2019, Google published their quantum computer with 53 Qubits. In 2020, UTSC published a

Category	Algorithm	Effect
Symmetric key	AES	Double key length
Hash	SHA2, SHA3	Double length
Public key	RSA, ECC	Broken
Digital signature	DSA	Broken

Figure 1.11 : Impact of quantum computer on prevalent cryptography.

quantum computer prototype for Bose sampling problem with peak Qubits of 76. It may be promising that in 20 years, quantum computers can be used to solve real world problems, among which cryptography is the one of the most impacted. Shor's algorithm is designed for quantum computers. It is capable of solving problems of big integer factorization and discrete logarithm. Unfortunately, RSA, DSA and ECC are the victims of Shor's algorithm (Fig. 1.11). They have been the standard algorithms used in many public key encryption and digital signature. Many TPM and TEE use them as part of the root of trust. To handle this upcoming challenge, NIST has hosted a competition for post-quantum cryptography. It went through 3 rounds and 15 candidates remained. Among them, lattice-based and code-based cryptography are the most popular candidates. Similar to conventional cryptography, they are computation-intensive and above the capability of embedded devices. In order to cope with potential threats brought by quantum computers, efficient hardware modules for post-quantum cryptography need to be studied to protect resource-constraint devices in the new era.

1.3 Thesis Statement and Proposed Memory-centric Hardware Security for Resource-constraint Devices

To summarize the problems of current resource-constraint hardware in security area, we are facing the following challenges:

•Conventional protection approaches could not fit into edge devices due to excessive cost. These approaches ranges from key generation and storage, cryptography schemes and network intrusion detection, etc.

•Traditional software and hardware architecture needs further innovation to adapt to emerging computation tasks.

•Conventional protection approaches is vulnerable against various attacks including invasive and non-invasive methods.

•Many popular security schemes are broken by quantum computer attacks.

In this thesis, new solutions are proposed to address the above issues. In order to provide area and energy efficient security hardware primitives for resource constrained devices, we look into the areas of secret key generation and storage, hardware firewall and post-quantum cryptography (Figure 1.12). This thesis presents several projects that handles existing problems with innovative circuits, architectures and algorithms. First one is for key generation and storage. It is a subthreshold voltage inverter based PUF. Second one is for network intrusion detection, it is a memory-centric hardware firewall. And the third aspect is the memory-centric crypto engine. It includes in-memory accelerator for NTT and CDT sampler, mainly targeting at lattice-based schemes for post -quantum cryptography. The 3 aspects are targeted to address the problems above about current approach. And they serve as the root of trust and part of the chain of trust for edge devices with resistance against physical attacks, security

against quantum computers, and high area and energy efficiency.

There are some common methodologies and techniques in my solutions. They all use crossbar arrays composed of custom memory or analog cells. They use in-memory and near-memory computations for high bandwidth, energy efficiency and low area. The most computation intensive parts are accelerated using memory-centric approach. Other controller and secondary modules are designed in Verilog. And then the whole engine is designed in the VLSI flow treating the custom circuits as an IP. Certain flexibility and configurability is designed to fit into different applications.

1.4 Outline and Contribution of the Thesis

The organization of this thesis is as follows: in Chapter 2, we present a $562F^2$ physically unclonable function (PUF) with a zero-overhead stabilization scheme to provide robust key generation and non-volatile storage. The PUF features a subthreshold inverter-based PUF cell to provide high gain and low power consumption. Column-wise native transistor-based voltage regulation suppresses influence of supply voltage fluctuation on PUF stability. A in-cell reconfiguration scheme combined with body-bias sweep further stabilize the PUF with no area overhead. SRAM-like array and peripheral enables high-speed readout for the secret keys.

In Chapter 3, we present a 8-T dual-port CAM-based network intrusion detection engine serving as a hardware firewall. The engine make use of 3-phase architecture to handle signature-based pattern matching for deep packet inspection. Aho-Corasick algorithm was deployed to construct pipelined NFA in the engine, which was implemented with processing element (PE) array. The PE array can be flexibly reconfigured for different depth and stage to fit into different scenarios and rule sets. Range-enable CAM helped eliminate excessive energy overhead induced by activating

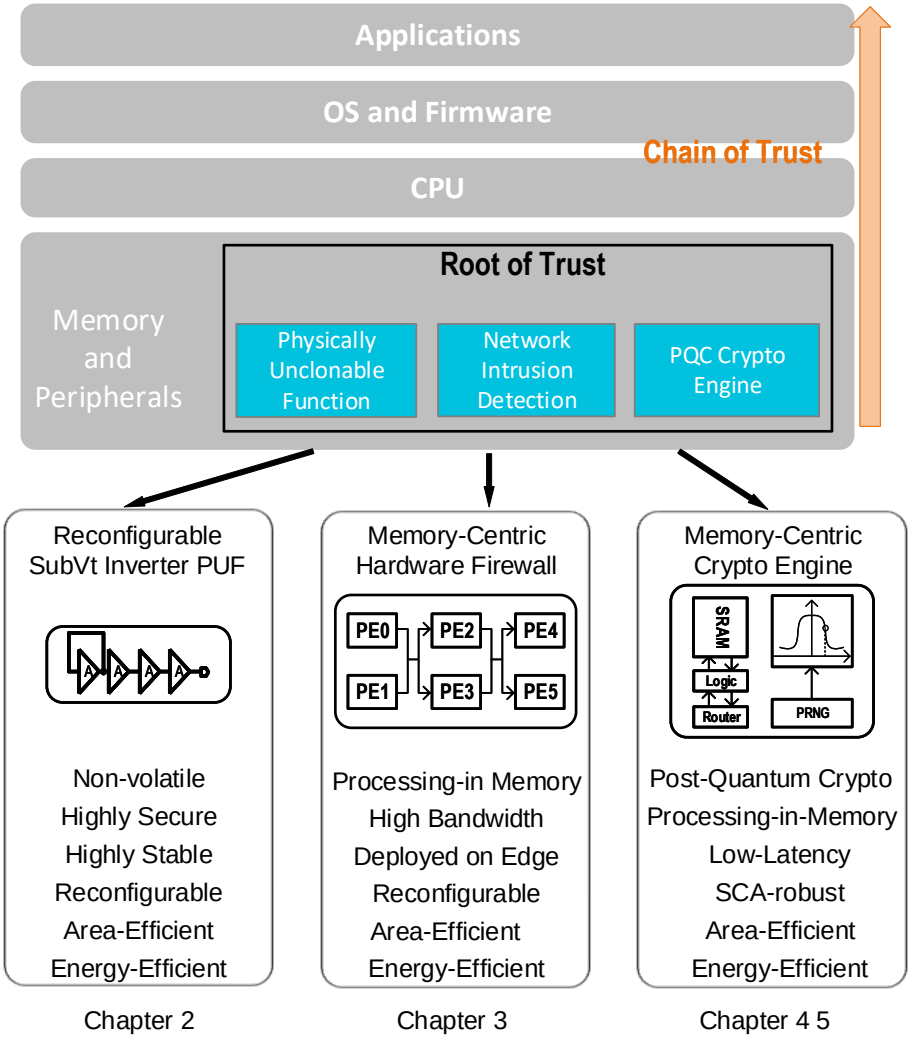


Figure 1.12 : Scope of this work.

the whole CAM array, which is made possible by a custom range decoder based on switches. We designed innovative dual-port 8-T CAM cell for doubled throughput and high density. A fixed-1s' encoding was proposed to work closely with the 8-T cell. A variant 7-T cell was capable of supporting masking as in 16-T TCAM cells.

In Chapter 4, we present a 6-T SRAM-based NTT accelerator for lattice-based cryptography. In-memory and near-memory logic were designed to handle modular arithmetic in Number Theoretic Transform (NTT) operation, which is the main

process of Ring Learning with Errors (RLWE) and Module Learning with Errors (MLWE). The modular arithmetic were performed in bit-serial approach, which enabled parallel processing of all NTT items. The structure scales well with post-quantum cryptography algorithms of high order polynomial and low bit-width. A novel mapping scheme was proposed to reduce the area overhead for inter-NTT-stage data movements. The SRAM bank is reused from stage to stage to provide compactness and high energy efficiency.

In Chapter 5, we present a Range-Matching-CAM-based CDT sampler for post-quantum cryptography. The sampler make use of novel CAM cell which is capable of making range matching with differential match lines. Segmented CAM banks ensure early termination of most searches to achieve high energy efficiency. Row-wise and column-wise power gating make possible configurable sampling range and precision, further lowering power consumption. To elevate robustness against side-channel attacks, the sampler is capable of activating rows randomly without influencing the sampling output, masking the energy cost.

My contribution in the thesis can be summarized as follows:

1. Proposed memory-centric hardware accelerators for security issues in key generation, network intrusion detection and post-quantum cryptography. The architecture and circuit level innovation produced significant improvement in terms of area, energy and latency. No prior solutions have reached this area with similar approach and silicon results.

2. Proposed unique cell and stabilization scheme for self-regulated PUF [12] [13] [14], which presented state-of-the-art native stability, best-in-class area-efficiency and energy-efficiency for silicon PUFs, making it much more reliable for private key storage in real devices.

3. Presented a software-hardware co-optimized NIDS hardware using innovative CAM design [15]. It is the first silicon solution for hardware NIDS in edge devices. The system can also be utilized to solve many other automata problem like gene sequencing and data mining.

4. Proposed a computation-in-6-T-SRAM accelerator for NTT and INTT, which deployed optimized bit-serial operation for generic modular arithmetic, and can be applied to many other cryptography applications.

5. Designed MePLER [16], a CDT sampler which solved the throughput and side-channel robustness issue in existing hardware samplers.It is world's most energy-efficient and side-channel secure sampler as of now.

Chapter 2

Reconfigurable Physically Unclonable Function

This section presents a reconfigurable physically unclonable function (PUF) design fabricated in 65nm CMOS technology. Subthreshold-inverter-based static PUF cell achieves 0.3% native bit error rate (BER) at 0.062 fJ/bit core energy efficiency. A flexible, native transistor-based voltage regulation scheme achieves low-overhead supply regulation with 6 mV/V line sensitivity, making the PUF resistant against voltage variations. Additionally, the PUF cell is designed to be reconfigurable with no area overhead, which enables stabilization without redundancy on chip. Thanks to the highly-stable and self-regulated PUF cell and the zero-overhead stabilization scheme, a 0.00182% native BER is obtained after reconfiguration. The proposed design shows 0.12%/10 °C and 0.057%/0.1 V bit error across military-grade temperature range from -55 °C to 125 °C and supply voltage variation from 0.7 V to 1.4 V. The total energy/bit is 15.3 fJ. Furthermore, the unstable bits can be detected by sweeping body bias instead of temperature during enrollment, significantly reducing the testing costs. Last but not least, the prototype exhibits almost ideal uniqueness and randomness, with a mean inter-die hamming distance (HD) of 0.4998 and a 1020× inter/intra-die HD separation. It also passes both NIST 800-22 and 800-90B randomness tests.

2.1 Motivation and Related Work

Physically unclonable functions (PUFs) are increasingly studied and developed for secure electronic devices, especially for resource-constrained systems like Internet of Things (IoT) [17] [18] [19]. PUFs harvest intrinsic process variations of integrated circuits to generate keys and IDs unique to each device. It not only provides a low-cost solution to secure key generation and storage, but also offers attractive features such as bonding with specific hardware and tamper evidence, which makes them viable solutions for many emerging hardware security issues on supply chain tracking, counterfeit protection, system attestation, etc.

The generation and storage of secure keys are critical for entity authentication, secure communication, and serving as roots of trusts for computing systems. Ubiquitous IoT devices face additional challenges because of varying environmental conditions and physical access by attackers. An ideal solution should exhibit the following three properties. First, non-volatile storage of the keys under voltage and temperature variations is necessary. Second, low cost and energy consumption are essential for IoT devices with constrained resource and battery lifetime. Third, robustness and alertness against physical tampering attacks is highly desired.

Non-volatile memories (NVMs) are the conventional solutions for secret key storage. NVMs include one-time programmable read-only memories (ROMs), fuses, and programmable flash memories. While NVMs provide excellent reliability and long-term data storage, NVM-based key storage solutions suffer from the following drawbacks: (1) Most NVMs require extra fabrication steps, adding to higher fabrication costs; (2) Conventional ROM, fuses and flash memories are all vulnerable to physical attacks, such as optical imaging techniques and direct probing; (3) Software vulnerabilities can also be exploited to gain access to keys stored in NVMs with standard

I/O interface.

Physically unclonable function (PUF) is the most promising alternative to conventional NVM and is expected to meet all the desired properties for secure key storage. Firstly, PUFs use intrinsic process variations to generate and store the keys. The keys are stored in device characteristics, rather than direct digital data storage, making it more difficult to directly read out stored keys with tampering attacks. Moreover, PUFs exploit small device variations, which are believed to be sensitive to physical tampering and make PUFs tamper-evident. Secondly, the keys are unique to each chip due to the random and chip-specific nature of process variations during chip manufacturing. Therefore, no key programming is required and cloning a known device is almost impossible, making PUF literally "unclonable". Thirdly, silicon PUFs are low cost and easy to integrate with modern system-on-chip (SoC), because they only require standard CMOS devices and are easily portable across process nodes. Lastly, PUFs can be designed to be highly area and energy efficient, making them suitable for systems ranging from IoT devices to high-performance SoCs.

Generally, PUFs are categorized into strong and weak ones based on their capabilities. Strong PUFs [20] [21] [22] [23] [24] provide a large number of challenge-response pairs (CRP) for direct authentication, but existing designs have limited stability and randomness, leading to vulnerabilities against machine learning attacks [25]. Recent works explored circuit nonlinearity to improve resiliency against machine learning attacks [22] [23]. Weak PUFs have smaller capacity, but achieve higher stability against PVT variations and close-to-ideal uniqueness to provide reliable IDs and keys. This work focuses on weak PUFs and we do not differentiate strong and weak PUFs in the rest of the chapter. The following metrics are widely adopted to evaluate PUFs: reliability, randomness, area, energy efficiency, and throughput, out of which the stability

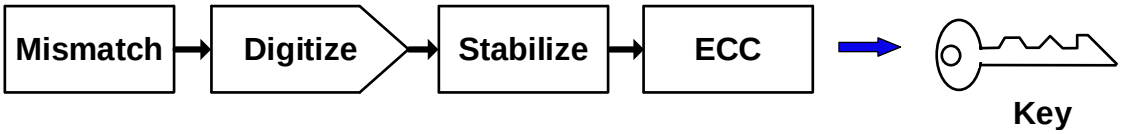


Figure 2.1 : Typical PUF diagram from entropy extraction to key generation.

over PVT variations is the major challenge to PUF designs.

As shown in Fig. 2.1, a PUF usually comprises of four stages. The random mismatch of transistors or other components, acting as the entropy source, is extracted and digitized into a binary data. The raw PUF responses then go through stabilization process and error-correcting code (ECC) to remove and correct all unstable bits, providing 100% reliable keys for security applications. In theory, all bit errors can be corrected by reserving enough redundant bits for stabilization and ECC under a given bit error rate (BER). However, the redundancy and complexity increase super-linearly with BER. According to [26], the energy requirement for ECC to correct one-bit error in a 256-bit key equals to the energy of accessing over 200 PUF cells. Therefore, achieving higher stability at early stages of the whole PUF processing flow is the key to improve the overall PUF performance, demanding better PUF cell design and stabilization method.

PUF design involves entropy extraction and digitization. Various circuit topologies have been proposed for entropy extraction, including metastable cross-coupled inverters [27], power-up state of SRAM cells [28], delay lines [19], oscillators [21], current mirrors [29], PTAT references [30], and leakage-based transistor pairs [31]. Comparing these PUF designs, circuits with static operations avoid random noise associated with dynamic transients, generally providing more reliable responses. In terms of digitization, comparators shared by one column or the whole array were com-

monly deployed for lower area overhead, but the comparator must be designed with high gain and accurate offset cancellation, which incurs high power and complexity. Even microvolt offsets lead to biased PUF responses and reduced uniqueness. Moreover, even with optimal offset cancellation, shared comparators suffer from long wires and coupling effects. Comparatively, if the digitizer is integrated into every cell locally [32], comparator offset becomes part of the entropy source and will not affect PUF uniqueness. Complementary current mirror-based monostable PUF design is one of the first implementations to combine static operation and local digitization [29] [33]. This fully static design provides strong resiliency to environmental variations and random noise because of the absence of dynamic switching events, at the expense of higher standby power and larger PUF cell footprint. Another local amplification scheme is proposed in [32], where a NAND gate chain amplifies threshold voltage differences between neighboring stages. This design obtained lowest BER among all reported CMOS PUFs and a compact footprint, but it consumes high short circuit power. In [34], a chain of sub-threshold 2-transistor amplifiers further improves the NAND chain design and achieves best-in-class metrics.

Numerous stabilization techniques have been proposed to correct or discard error bits, which usually comes with area and time overheads. Temporal majority voting (TMV) and spatial majority voting (SMV) filter out random noise by taking multiple bits in time or space domain for a single bit output. While the complexity for TMV and SMV is low, the enhancement of stability is limited. Burn-in through intentional aging [27] [35] improves stability with little area overhead but incurs high testing cost. Another widely used stabilizing technique is to find and filter out unstable PUF cells during enrollment. However, long testing time with temperature sweep are necessary to find all unstable cells. It also requires redundant cells in the PUF array on chip

for replacement and on-chip or off-chip storage of masking map. Lastly, ECC such as BCH code [27] [32] is capable of 100% stability at the expense of high computing complexity, high redundancy and long latency.

To further improve PUF stability without extra area, power, and throughput overhead and sacrifice of technology portability, this chapter presents a self-regulated and reconfigurable PUF design with zero-overhead stabilization scheme [12]. It features three major advantages:

•Subthreshold inverter-based static and local digitization structure exhibits ultra-low power consumption, state-of-the-art native stability, and compact footprint.

•Native transistor-based regulation provides subthreshold supply voltage for PUF cells with low overhead and further improves PUF's resistance to voltage variations.

•The proposed in-cell reconfiguration scheme enables 100 times BER reduction with no area overhead on chip.

2.2 Proposed PUF Cell Design

In order to achieve high stability with low area and power overhead, the proposed PUF cell employs a 4-stage subthreshold inverter chain for entropy extraction and amplification, inspired by the static and local digitization designs in [32] and [34]. As shown in Fig. 2.2, the input and output of first stage is shorted, setting its voltage to a switching point with high gain. Mismatch between switching voltages of successive stages is amplified to full rail after a few stages. This PUF structure eliminates the impacts of non-ideal comparators and noise during dynamic transitions, leading to higher stability. However, the use of NAND gates at nominal VDDs incurs high power consumption and 2-transistor amplifiers are found to be less effective at more advanced technology nodes than 180nm. The following subsections will discuss the

use of subthreshold inverters, low-overhead supply regulation of the PUF cell, and system implementation of the PUF module.

2.2.1 Subthreshold Inverter Based PUF Cell

The stability of PUFs based on a chain of amplifiers depends on the distribution of switching voltage mismatch and the amplification gain of each stage. Firstly, higher gain improves overall PUF stability over environmental variations because less stages are involved in deciding the final responses. Secondly, larger mismatch variations apparently reduce the probability of having an unstable cell. Thirdly, if the gain around the switching voltage is asymmetrical, the PUF response will be biased towards one value and the overall randomness is affected. It is found that the 2-transistor amplifier in [35] shows reduced voltage gain at more advanced process nodes because of decreased output impedance. Because an off-transistor is used to bias the amplifier, the switching voltage is close to the supply voltage, leading to unbalanced gain and headroom.

As shown in Fig. 2.2, we propose the use of thick-oxide inverters under subthreshold supply voltage, which provides higher gain and balanced output while consuming low static power. It is shown in Fig. 2.2(c) and Fig. 2.3 that as technology nodes scales, subthreshold inverter consistently yields higher voltage gain than 2-transistor amplifier, and shows symmetrical gain and headroom. Therefore, the proposed PUF cell achieves higher native stability, better randomness, and better portability to different process over the 2T-amplifier design in [35].

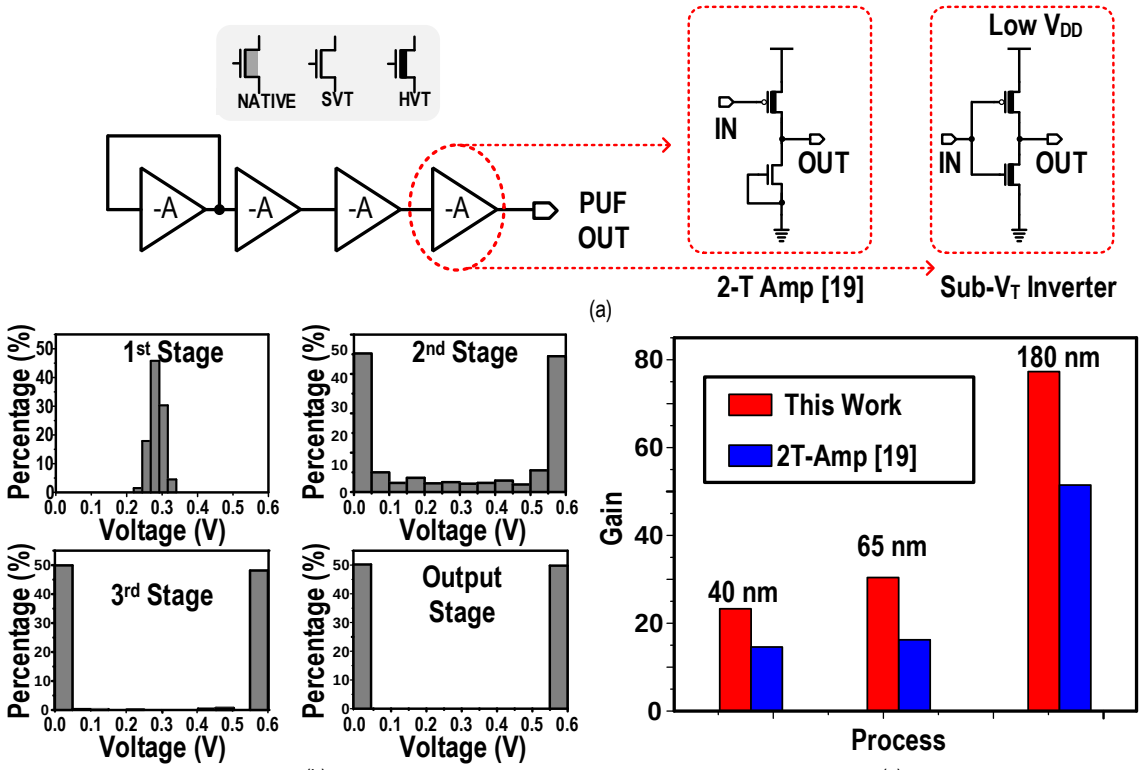


Figure 2.2 : (a) PUF cell topology, comparison of 2-transistor amplifier-based cell and proposed subthreshold inverter-based cell, (b) simulated histogram of voltages at each stage, (c) voltage gain of 2-transistor amplifier and subthreshold inverter in different technology nodes.

2.2.2 Voltage Regulation using Native Transistors

Stable and efficient low voltage supply is required for the subthreshold inverter-based PUF cell for stability and low energy consumption. One of the main sources of instability for PUF comes from voltage variation, especially in subthreshold PUF designs [24]. IR drop, EM interference and other incidents can cause fluctuation of supply voltage in PUF cells.

A conventional approach to provide a regulated low supply is to use a low drop-out regulator (LDO). LDOs have quiescent currents, leading to low efficiency and large area overhead when the load is light. PUFs can be offered in either a large array

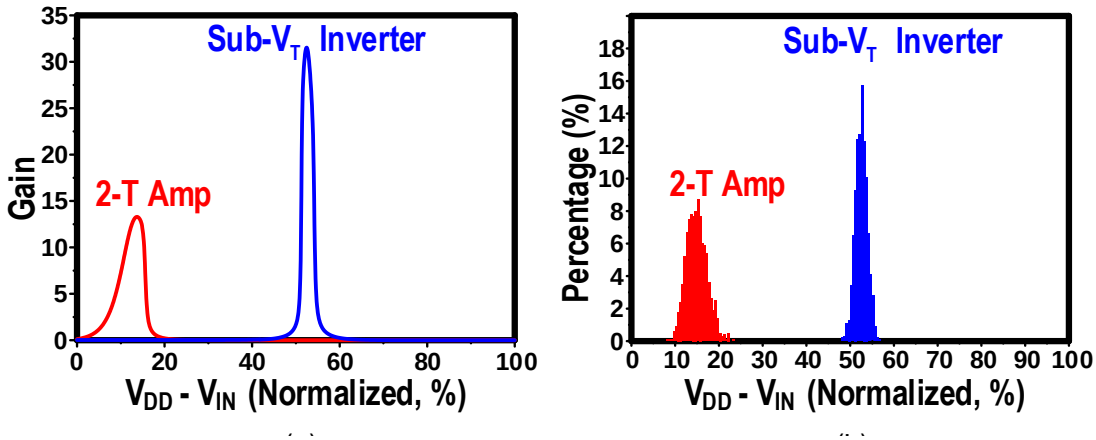


Figure 2.3 : Comparison of (a) gain and (b) voltage histogram between 2-transistor amplifier and subthreshold inverter.

or as small embedded key registers. Thus, a more scalable and low-overhead supply regulation is desired for our proposed PUF circuits.

Inspired by a threshold-based voltage reference [36], this work utilizes a native transistor to self-regulate supply voltage with low area and power overhead. The voltage regulation works on the basis that the cell is biased in subthreshold region. The first stage consumes most of the current and the rest stages are almost in cutoff region. Therefore only the first stage is considered in the following calculations. The equivalent model is shown in Fig. 2.4. The subthreshold region currents of M0 and M2 are derived in (1) and (2), according to [36], where Isub denotes the cell current flowing through M0, M1 and M2 and should be equal in (1) and (2). V_M denotes the switching voltage of stage 1. V_{VDD} denotes the regulated virtual supply voltage of the inverters.

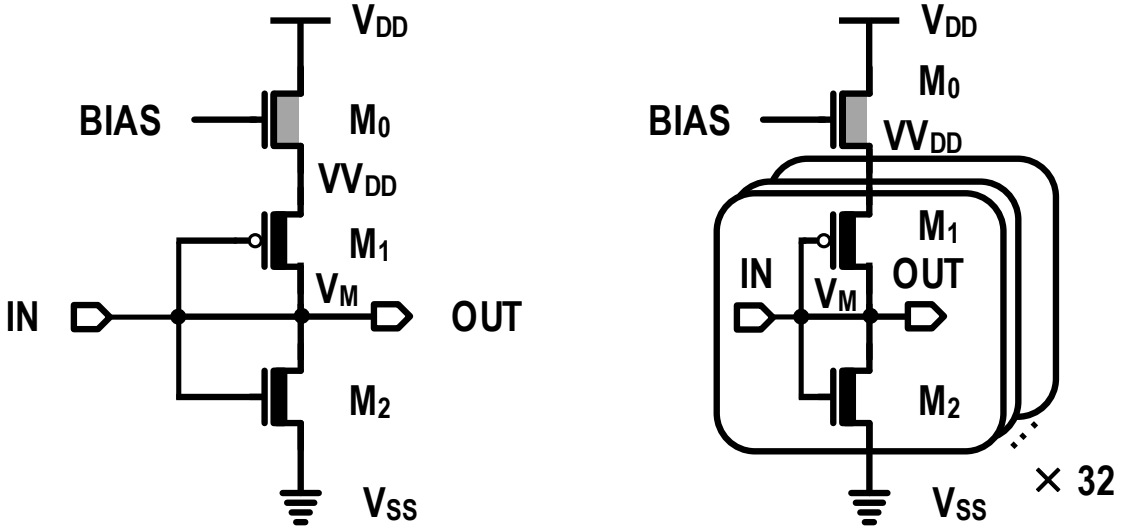


Figure 2.4 : (a) Cell-wise voltage regulation model using a native transistor, (b) column-wise voltage regulation model using a native transistor.

$$I_{sub} = \mu_0 C_{OX0} W_0/L_0 (m_0 - 1) V_T^2 exp(V_{BIAS} - V_{DD} - V_{th0})/(m_0 V_T))(1 - exp(-V_{ds0}/V_T)) \tag{2.1}$$

$$I_{sub} = \mu_2 C_{OX2} W_2/L_2 (m_2 - 1) V_T^2 exp(((V_M - V_{th2})/(m_2 V_T))(1 - exp(-V_{ds2}/V_T))) \tag{2.2}$$

V_{DS} in (2.1) and (2.2) are by magnitude larger than VT thus the last terms can be neglected. To simplify the model, it is assumed that V_M is half of V_{VDD}. In real design, V_M does not necessarily need to be exactly half of V_{VDD}.

$$V_M = 1/2 V_{VDD} \tag{2.3}$$

The equation for V_{VDD} can be derived in (4), showing independence of supply voltage. Moreover, since thermal voltage is proportional to temperature and threshold voltage

is complementary to temperature. The temperature effects of the transistors on V_{VDD} can be cancelled by carefully sizing the native transistor.

$$V_{VDD} = (2m_0m_2)/(m_0 + 2m_2)V_T ln(\mu_0 C_{OX0} W_0 L_2(m_0 - 1))/(\mu_2 C_{OX2} W_2 L_0(m_2 - 1)))$$

$$+(2m_0)/(m_0 + 2m_2)V_{th2} + (2m_2)/(m_0 + 2m_2)(V_{BIAS} - V_{th0}) \quad (2.4)$$

Further area saving is achieved by column-wise sharing of the native regulation transistor in this design. The flexible configuration of the native regulation transistor can adapt to cell-wise, column-wise and array-wise regulation. In the proposed design, each column of 32 cells share a native regulating transistor. The V_{VDD} equation for this case is modified as in (5).

$$V_{VDD} = (2m_0m_2)/(m_0+2m_2)V_T ln(\mu_0 C_{OX0} W_0 L_2(m_0-1))/(32\mu_2 C_{OX2} W_2 L_0(m_2-1)))$$

$$+(2m_0)/(m_0 + 2m_2)V_{th2} + (2m_2)/(m_0 + 2m_2)(V_{BIAS} - V_{th0}) \quad (2.5)$$

2.2.3 System Implementation

The block diagram of this design is depicted in Fig. 2.5(a). A 32 by 128 array is implemented. SRAM-style peripherals are integrated to allow for parallel and high-speed readout. Each column shares the same virtual supply voltage regulated by a native transistor. Readout process involves BL pre-charging, WL enabling, voltage sensing through a single-ended sense amplifier (SA), and latching of the result. A waveform for reading operations is included in Fig. 5(c).

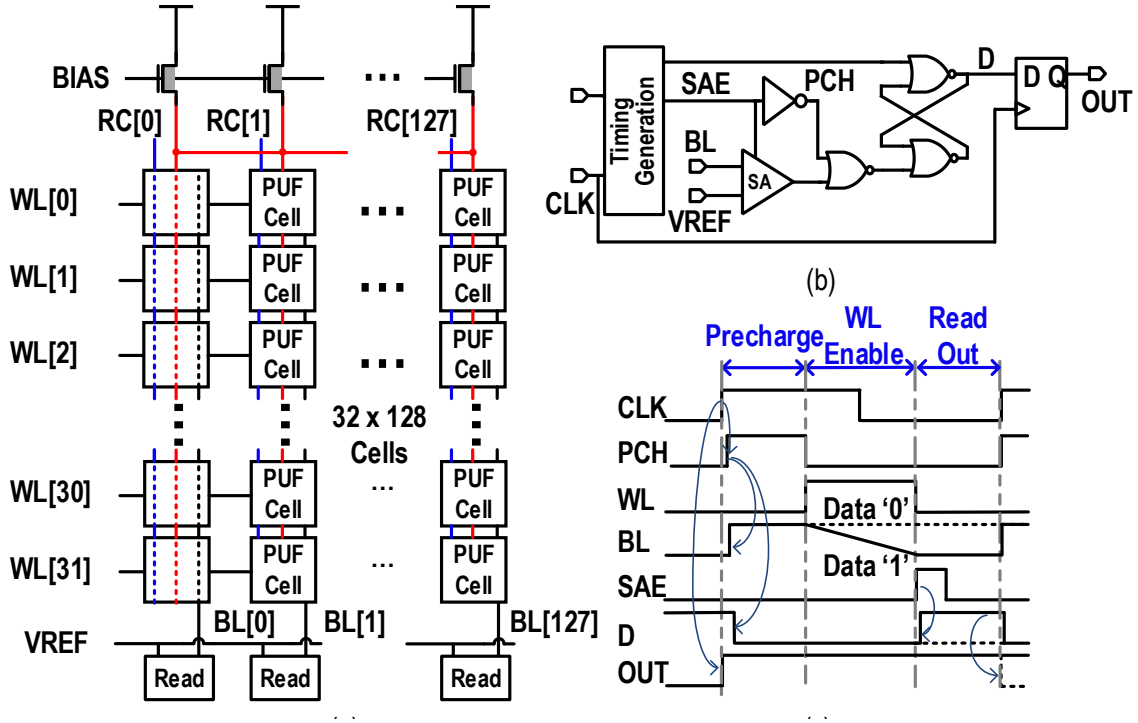


Figure 2.5 : PUF array (a) block diagram, (b) readout circuits schematic, (c) readout timing.

2.3 Zero-Overhead Reconfiguration

Conventional stabilization methods, as discussed before, could not achieve good balance between overhead and stabilization effect. A zero-overhead reconfiguration scheme is proposed to stabilize the PUF cell. The scheme is based on the specific structure of the proposed PUF cell, and can be enrolled with low testing cost. In this work, the V_M difference between the first two stages is the entropy source. Cell-level reconfiguration involves transistor merging and converting the original cell into a new cell with independent performance.

Cell Mismatch	Probability	Action	Output
$V_{M1} > V_{M2}$	$0.5*(1-P_O)$	Stay	'0'
$V_{M1} < V_{M2}$	$0.5*(1-P_O)$	Stay	'1'
$V_{M1} \approx V_{M2} > V_{M3}$	$0.5*P_O*(1-P_R)$	Reconfigure	'1'
$V_{M1} \approx V_{M2} < V_{M3}$	$0.5*P_O*(1-P_R)$	Reconfigure	'0'
$V_{M1} \approx V_{M2} \approx V_{M3}$	P_O*P_R	Reconfigure	Unstable

Table 2.1 : PROBABILISTIC MODEL FOR RECONFIGURABLE PUF CELL

2.3.1 Source of Instability

The entropy source of PUFs is the local mismatch due to process uncertainty. The randomness in dopant distribution, lithography, gate thickness causes variations of transistors. Ideally the switching voltages of two inverters within the same PUF cell should be fixed, leading to a stable '1' or '0' output. However, the actual mismatch can be altered or even flipped under voltage and temperature variations, especially when the original mismatch is small. This is caused by the fact that different transistors exhibit varying environmental sensitivities.

Monte-Carlo simulations of 10,000 proposed PUF cells are performed to investigate the cause of instability. More focus is put on temperature variations because the proposed native voltage regulation suppresses the impacts of voltage variation. Simulation results show that 96 out of 10,000 cells have unstable output under -55 °C to 125 °C temperature sweep. The histogram of switching voltage differences of stage1 and stage2 illustrate that all the 96 unstable cases have switching voltage differences smaller than 3 mV, as shown in Fig. 2.6. Conclusion can be drawn from above discussions that small mismatch is a necessary condition for an unstable PUF cell.

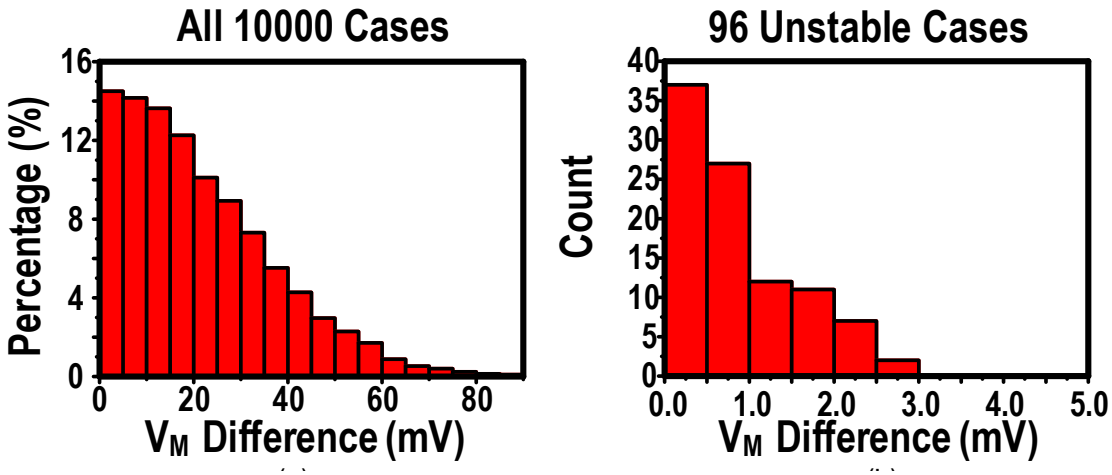


Figure 2.6 : (a) Histogram of switching voltage (V_M) difference from 10000 Monte-Carlos simulations, (b) from 96 unstable cases.

2.3.2 In-Cell Reconfiguration

The reconfiguration process is shown in Fig. 2.7 to further illustrate the source of instability and explain the proposed stabilization scheme.

An originally unstable cell has small mismatch between the first and second stage. The switching voltages of the first three stages of this cell are denoted by V_{M1}, V_{M2} and V_{M3} respectively. V_{M1} and V_{M2} are close to each other in nominal condition and are likely to flip their relative difference under V/T variations. Based on the observation that local mismatch is purely random, the reconfiguration scheme is proposed by combining the first two stages as a new first stage. The output of the reconfigured cell depends on the difference of the combined new stage and the third stage. The subthreshold inverter-based cell has a low bit error rate. Thus, the probability of V_{M1} and V_{M2} being close is low. For the reconfigured cell, the necessary condition of it being unstable over V/T variations is that V_{M1}, V_{M2} and V_{M3} are all close to each other. The probability is extremely low in the proposed design.

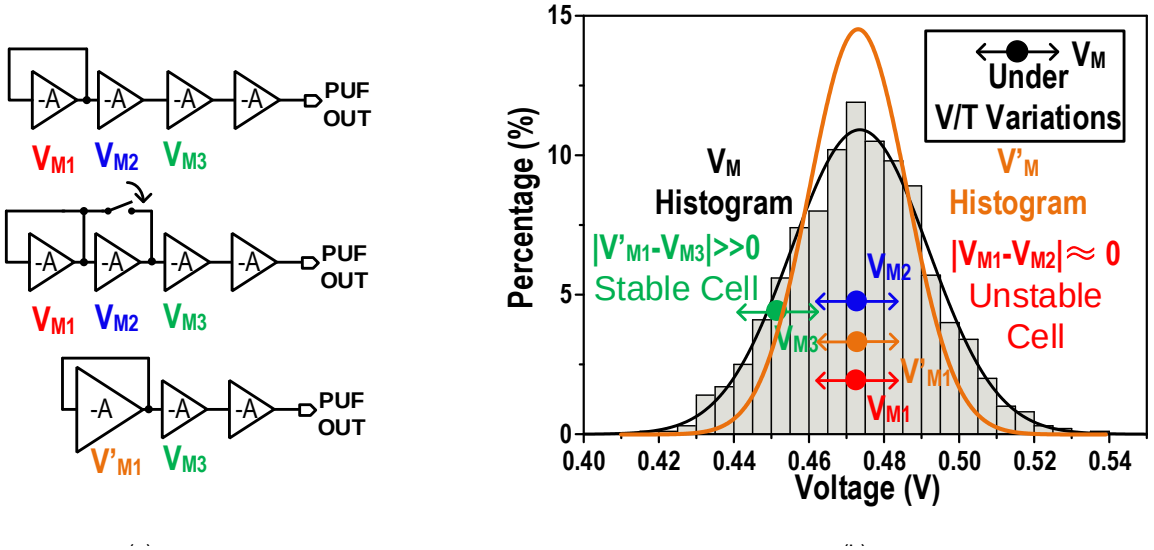


Figure 2.7 : (a) Process of in-cell reconfiguration of an unstable cell, (b) the switching voltages of an originally unstable cell before and after reconfiguration.

Table I describes the probabilistic model of the reconfiguration scheme. PO and AVO denotes the probability of unstable bits and gain for original cell. P_R and A_{VR} denotes the probability of unstable bits and gain for reconfigured cell. V_M and σ is the mean and standard deviation of a subthreshold inverter switching voltage The bit error rate and voltage gain have a relationship of $f(A_V)$. The probabilistic model of the process in shown in Table I. The switching voltages of the original cell and reconfigured cell exhibit the following distribution. It is assumed that transistor threshold voltage V_M is independent and identically distributed.

$$V_{M1}, V_{M2}, V_{M3} \sim N(V_M, \sigma) \tag{2.6}$$

According to [22], the standard deviation of threshold voltage is inversely proportional to transistor area, which leads to equation (7).

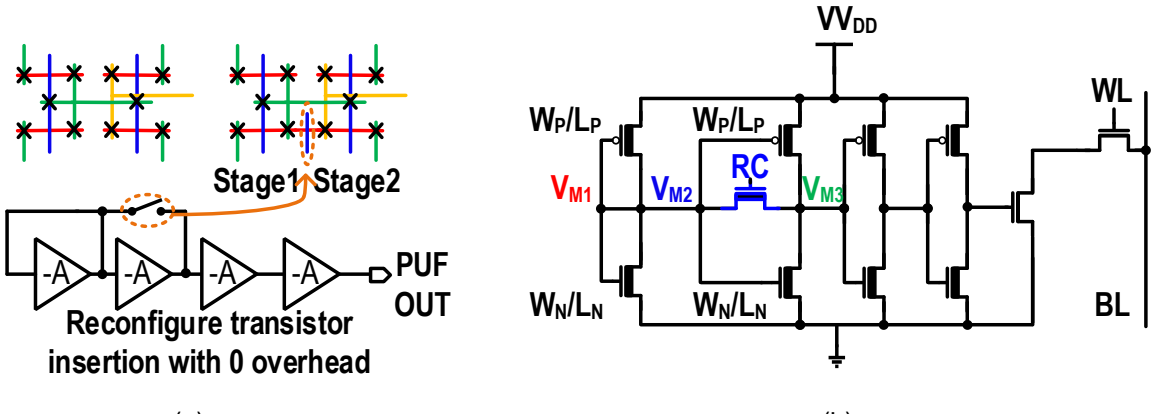


Figure 2.8 : Reconfigurable PUF cell in (a) layout, (b) schematic views.

$$V_{M1}^{'} \sim N(V_M^*, 0.707\sigma) \tag{2.7}$$

V'$_{M1}$ denotes the switching voltage of stage1 in the reconfigured cell. V$_M$* denotes the mean of a combined subthreshold inverter's switching voltage. The difference of switching threshold voltage between the first two stages can be derived in (8) and (9) for the original and reconfigured cell.

$$V_{M1} - V_{M2} \sim N(0, 1.414 * \sigma) \tag{2.8}$$

$$V_{M1}^{'} - V_{M3} \sim N(V_M^* - V_M, 1.224 * \sigma) \tag{2.9}$$

The probability of unstable bits is inversely proportional to the standard deviation of first two stages' switching threshold voltage difference.

$$P_O \propto f(A_{vO})/[Var(V_{M1} - V_{M2})]^{1/2} \tag{2.10}$$

$$P_R \propto f(A_{vR})/[Var(V_{M1}^{'} - V_{M3})]^{1/2} \tag{2.11}$$

Stabilization Method	Testing Cost	Redun-dancy	Runtime Latency	Efficacy
Temporal Majority Voting	Low	Low	Medium	Low
Burn-in	High	Low	Low	Medium
Mask (Filter)	High	Medium	Low	High
Error-Correcting Code	Low	High	High	Highest
Proposed Reconfiguration	Medium	Low	Low	High

Table 2.2 : COMPARISON OF STABILIZATION METHODS for PUF

From (10) to (11), P_R is estimated to be 1.15*f(A_{VR})/ f(A_{VO})*P_O. In measurements, the portion of cells requiring reconfiguration is 4% under -55 °C to 125 °C temperature variation and 0.7 V to 1.4 V supply voltage variation. The reconfigured cell has slightly higher BER than the original cell due to decreased mismatch induced by doubled size of the first stage [37]. Reduced gain due to reduced number of stages also induces higher instability. However, the overall probability for a cell to be unstable is the product of the two probability in theory, which is 0.44% in measurement.

2.3.3 Physical Implementation of Reconfiguration Scheme

The benefit of the proposed reconfiguration scheme is its low cost in this specific design. To combine the first and the second stage, an NMOS transistor is inserted between the drains of the two NMOS. Fig. 2.8 depicts the physical insertion of the reconfiguration transistor. In 65nm CMOS process, no area overhead is added under the design rules. In most technologies, the area overhead is also expected to be small. The reconfigure transistor is driven by full swing signal to connect two stages without voltage bias induced by V_{TH} drop. As shown in Fig. 2.5, reconfiguration control

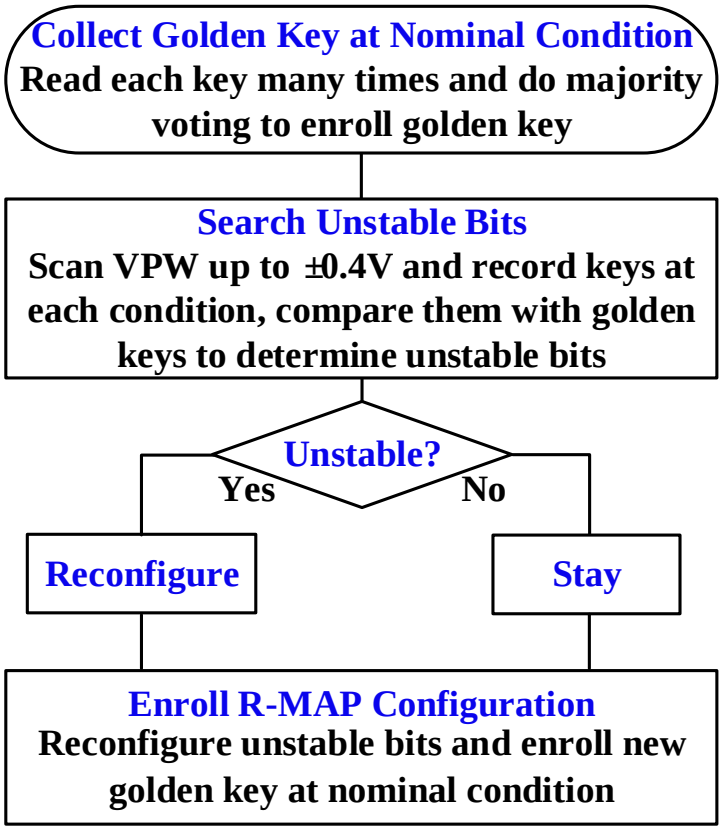


Figure 2.9 : Search and enrollment process of R-MAP configuration.

signals are applied vertically. For each PUF array, a reconfiguration map (R-MAP) is maintained in the server. And upon fetching a specific key, its corresponding reconfiguration setting is applied to the array. No on-chip redundancy and low runtime latency is added in this scheme, in contrast to masking and ECC approach.

2.3.4 Fast R-Map Searching via Body Bias Emulation

Ideally the R-MAP should be obtained by sweeping temperature and voltage in the desired operation range during enrollment. However, this approach consumes considerable testing cost and is expensive for massive production.

A key observation from (12) and (13) for the proposed PUF is that its entropy

source, threshold voltage, can be modulated by adjusting body bias to emulate temperature sweep. By properly sweeping body bias, the effects of temperature variation can be emulated and PUF cells with small mismatch may flip their outputs. By denoting such cells for reconfiguration, the necessary condition of them being unstable is eliminated, increasing the overall stability for PUF key generation. The R-Map search and enrollment process is shown in Fig. 2.9. In this design, deep n-well is used to isolate PUF array from external noise and enable PMOS body bias sweep. The amortized area overhead is negligible for each PUF cell. Emulating temperature variation using body bias (EVB) is fast and works at nominal condition, eliminating the need for expensive and time-consuming temperature sweep. The comparison of the proposed reconfiguration scheme and conventional methods is listed in Table II. It is highly effective with no on-chip area overhead, while adding low runtime latency and moderate test cost with the help of body bias.

2.4 Measurement Results

The chip is fabricated in 65nm CMOS process. The 32 by 128 cells array occupies 0.018 mm^2. The die micrograph and the layout of the PUF cell are shown in Fig. 2.10. Each PUF cell measures 0.96 µm by 2.475 µm, or 562 F^2. Clock generator is integrated to provide high-speed clocking for key access. The nominal condition for the PUF chip is 27 °C and 1.2 V supply voltage. Golden keys are collected at nominal condition by averaging our random noise with many samples. BER and unstable bit percentage results are measured by comparing separately collected samples under nominal and V/T variations with the golden key.

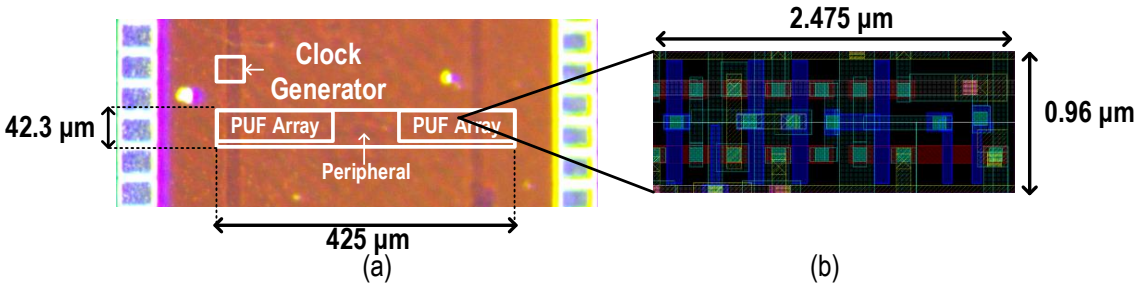


Figure 2.10 : (a) PUF chip micrograph, (b) cell layout.

2.4.1 Voltage Regulation

The reliability of the proposed native regulation is essential to the quality of the PUF. V_{VDD} is measured across supply voltage and temperature sweep to evaluate the efficacy of native regulation. The measured line sensitivity of V_{VDD} is less than 6mV/V over supply voltage range of 0.7 V to 1.4 V. Additionally, V_{VDD} shows less than 10mV variation across -55 to 125 °C. For testing purpose, bias voltage is provided through off-chip source. For system-level applications, it can be generated by a 2-transistor voltage reference [36] which is also robust against voltage and temperature variations. No extra current is consumed for voltage regulation with the use of native transistor regulation.

2.4.2 Native PUF Stability

The bit error rate (BER) and the proportion of unstable bits of 5 chips, measured at nominal condition before and after stabilization, are depicted in Fig. 2.12. BER denotes rate of bit flips over evaluations compared with golden value and unstable bit denotes percentage of ever-flipped bits. The definitions are adopted from [31] [32] [33] [34] for fair comparison. It should be noted that the proportion of unstable bits increases as the number of evaluations increases re shown for fair. The native BER

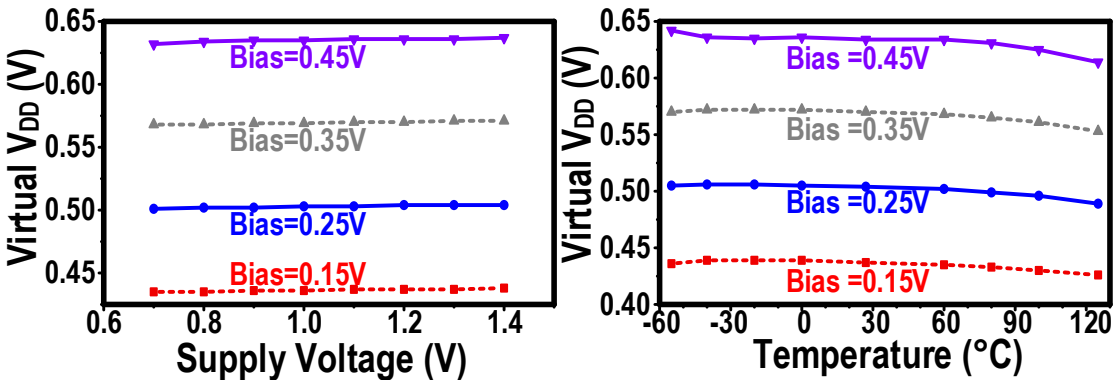


Figure 2.11 : VV_{DD} curve versus (a) supply voltage, (b) temperature under different bias voltages.

reaches steady state quickly after number of evaluations increases. The raw native BER is 0.30% without any stabilization methods. Temporal majority voting (TMV) reduced BER by 48.3% with 11 samples for 1 output. The reconfiguration methods followed by TMV can further reduce the BER to 0.00182%. The percentage of unstable bits is 2.95% after 2000 evaluations before stabilization. TMV11 alone reduces the percentage by 60.3% percent. Reconfiguration followed by TMV11 reduce the native unstable bits to 0.024%. More than 100 times BER improvement is achieved with low-overhead TMV11 and in-cell reconfiguration schemes.

2.4.3 PUF Stability Over Voltage and Temperature Variations

The stability of the proposed PUF is evaluated under the military-grade temperature range from -55 °C to 125 °C. The bit error rate (BER) over temperature sweep is 4.26% before stabilization. TMV and reconfiguration are applied to improve stability. Fig. 2.14(a) presents BER across temperature variation when different stabilization methods are applied. By detecting unstable bits under room temperature, a portion of bits that will be unstable over temperature variation is filtered out. However, this

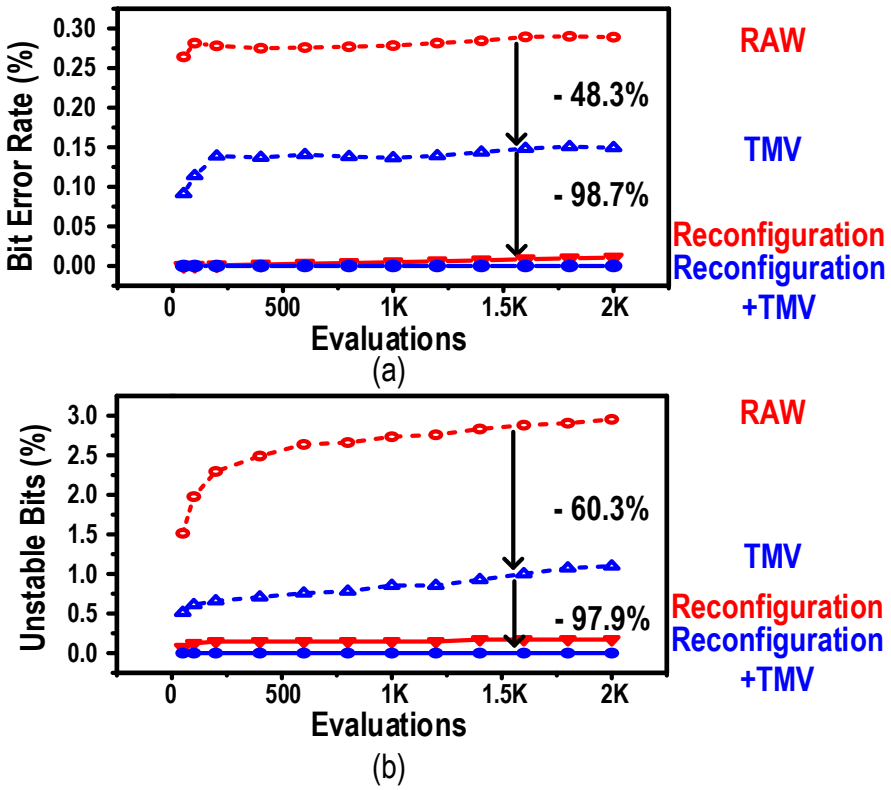


Figure 2.12 : (a) Bit error rate, (b) percentage of unstable bits versus number of evaluations under nominal condition.

only achieves 25% BER reduction. Since unstable cells under temperature variation are not necessarily natively unstable cells. Filtering out all unstable cells by sweeping temperature during enrollment achieves the highest stability, reducing BER to 0.44%. The remaining errors are caused by random noise and cannot be stabilized, the stability is still among best in class. For stability-first applications, higher stability can be obtained at the cost of higher testing cost. The proposed body-bias sweeping approach, named EVB, optimizes the tradeoff between stabilization effect and cost. BER is reduced to 2.27% by sweeping p-well biasing voltages (VPW) from -0.4 V to 0.4 V searching for unstable bits for reconfiguration. The process does not involve temperature sweeping thus has much lower testing cost than searching the

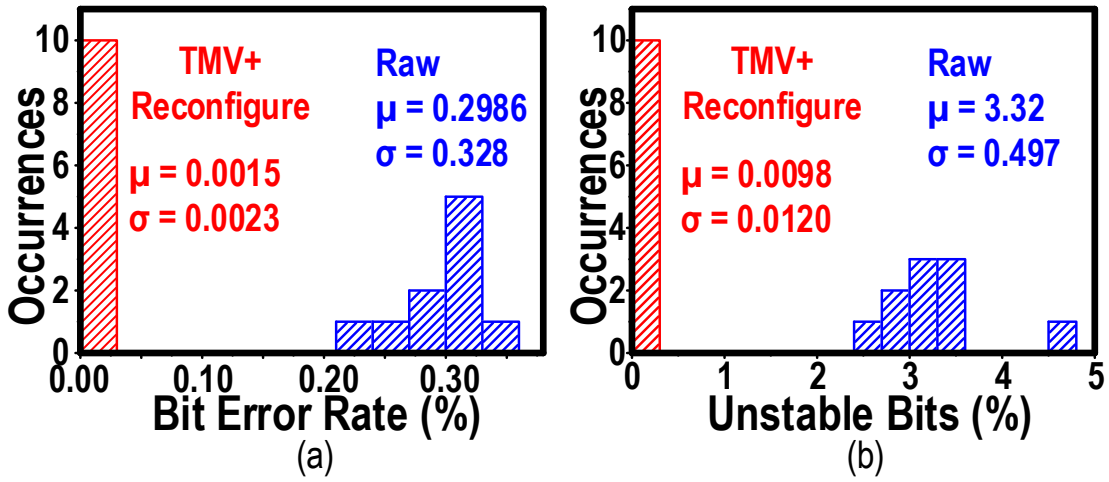


Figure 2.13 : (a) Native BER, (b) unstable bits histogram of 10 chips.

full temperature range.

Fig. 2.15 shows the detection rate and BER improvement of EVB at different VPWs. The detection rate is defined as the ratio of number of truly unstable bits across temperature variation filtered by EVB, over the number of total filtered bits by EVB at one VPW. Higher detection rate is observed for low VPW. This is expected because low VPW filters out cells with small mismatch, which is more likely to flip when temperature changes. Although detection rate lowers as body bias voltage rises, the overall stability still improves as shown in Fig. 2.15(b). High VPW filters out additional cells with larger mismatch, which will still flip under temperature variation, but with a lower probability. The BER improvement curve shows that positive VPW is more effective for high temperature while negative VPW is more effective for low temperature. This proves the assumption that the main source of entropy is mismatch of threshold voltage. Since positive VPW and high temperature both decrease threshold voltage, and negative VPW and low temperature both increase threshold voltage.

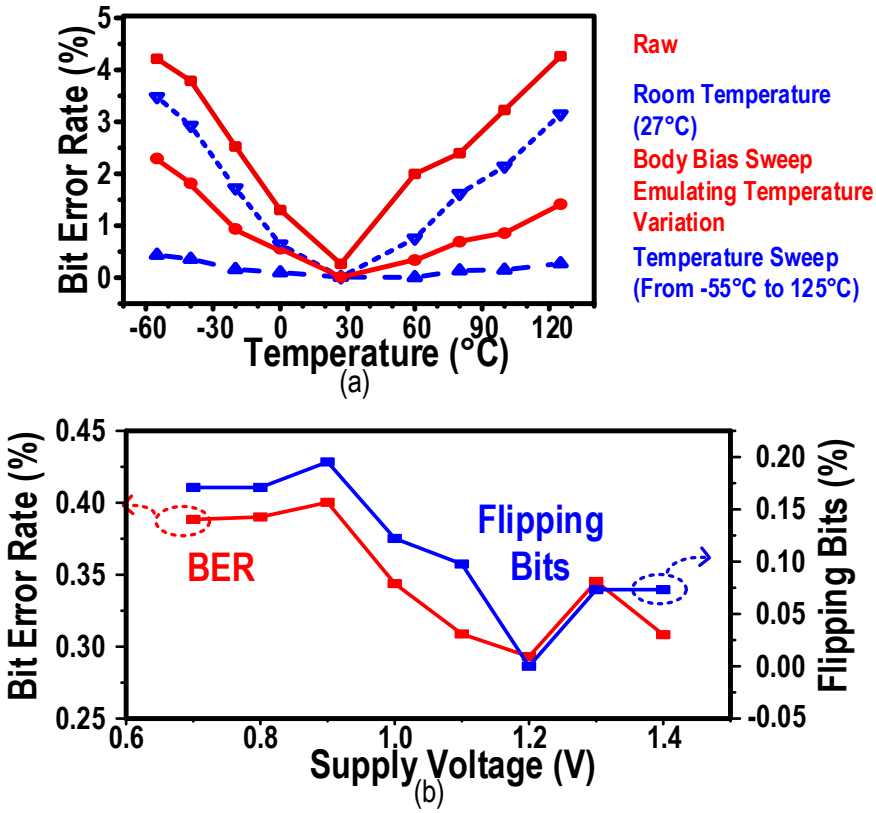


Figure 2.14 : (a) Bit error rate versus temperature variation under different stabilization methods, (b) BER and flipping bits versus supply voltage.

In addition to temperature variation, we also evaluate the PUF's resistance to supply voltage variation from 0.7 V to 1.4 V. Only 0.4% BER is observed across the voltage range, thanks to the supply regulation based on native transistors.

2.4.4 Uniqueness and Randomness

The inter-die and intra-die hamming distances are depicted in Fig. 2.16. The inter-die hamming distance measured over 10 chips has mean value of 0.4998 which is near-ideal. The mean intra-die hamming distance before stabilization is 0.0047, achieving 106 times separation between inter- and intra-die hamming distances. After stabiliza-

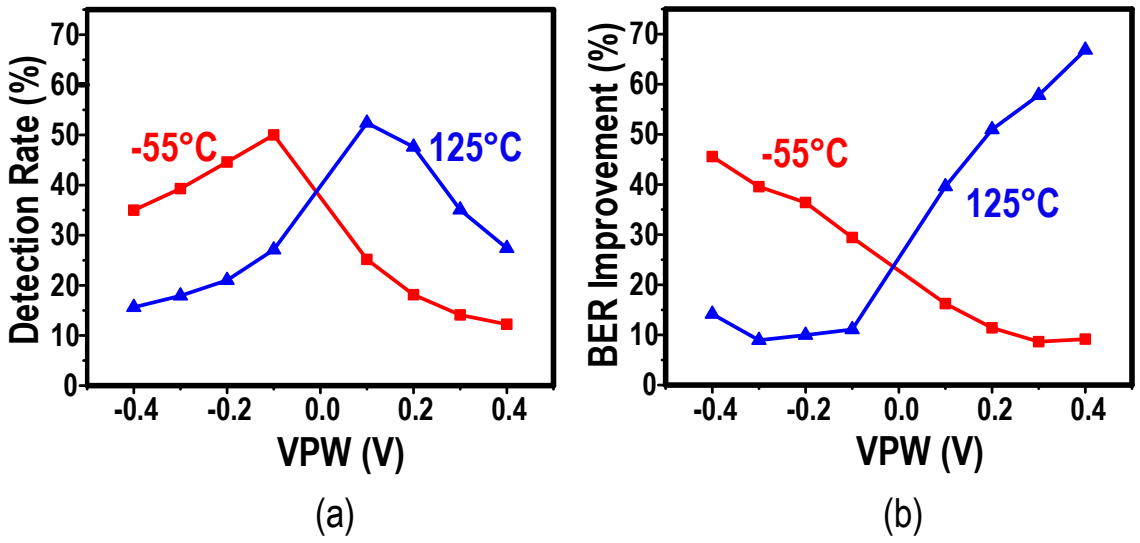


Figure 2.15 : (a) Detection rate, (b) BER improvement versus body bias sweep.

tion (reconfiguration followed by TMV), the intra-die hamming distance is 0.00049, showing state-of-the-art identifiability. The autocorrelation of 40960 PUF bits with the 95% white noise confidence level at 0.01385 is shown in Fig. 2.17. The near-ideal hamming distance and autocorrelation results validate the uniqueness of the proposed PUF.

To further evaluate the randomness of PUF responses, NIST 800-90B [38] and 800-22 [39] randomness test suites are performed on 40,960 bits collected from 10 chips. With the limited number of bits, 10 out of 15 subtests in 800-22 are available. NIST recommended settings were used to run the tests. More detailed definition and explanations of the testing parameters can be found in [38] [39]. The PUF bits passed all available sub-tests in the two suites, showing high-quality randomness as shown in Table III and Table IV.

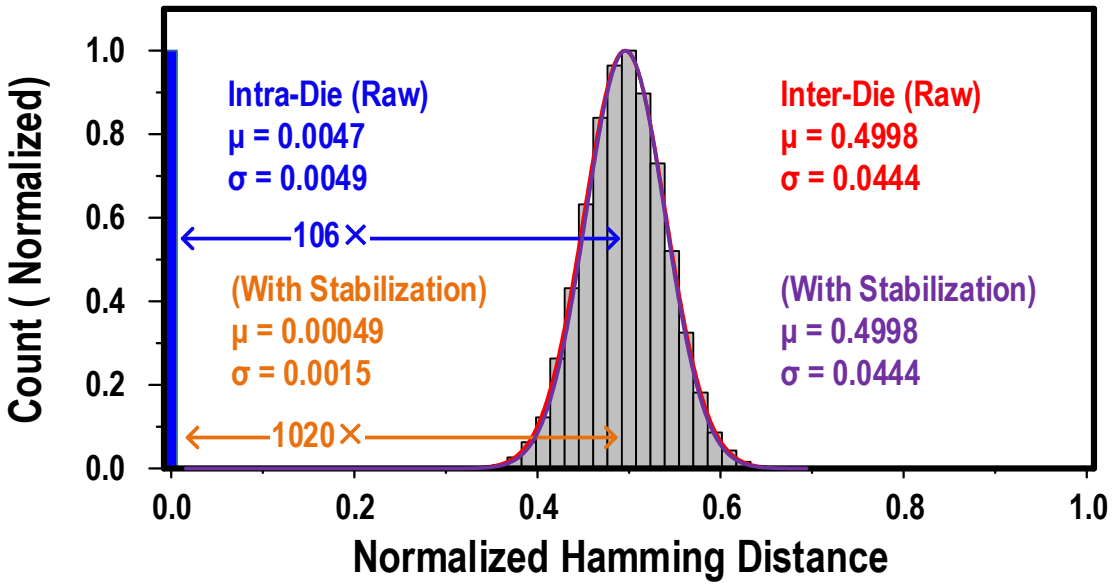


Figure 2.16 : Normalized Hamming Distance before and after reconfiguration.

2.4.5 Aging Effects

Another factor to consider for PUF stability is aging. Aging degrades PUF stability or completely flip bits. The main sources of aging effects in PUF are NBTI and HCI [30] [33] [21] [40]. In order to evaluate aging impacts, accelerated aging is applied by stressing the PUF at 150 °C and 1.4 V supply voltage. Every 12 hours, measurement was taken at nominal condition. A stressing of 108 hours in total was applied, resulting in equivalent effects as several years' aging under nominal condition. Since this design works in subthreshold region, it is expected to suffer minor aging effects. The measured aging-induced instability shown in Fig. 2.18 is similar to previously best reported results [30] [33].

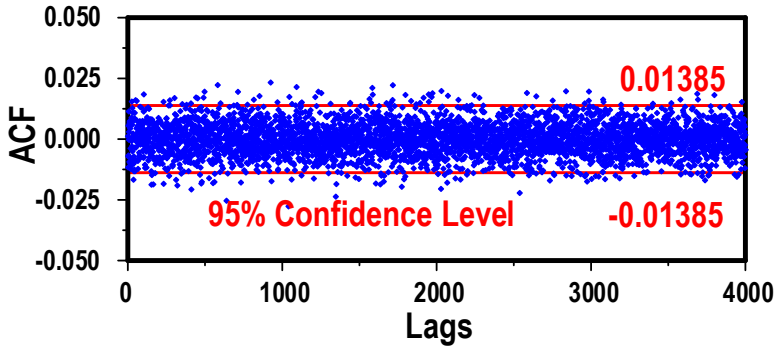


Figure 2.17 : Autocorrelation of 40960 bits for up to 4000 lags.

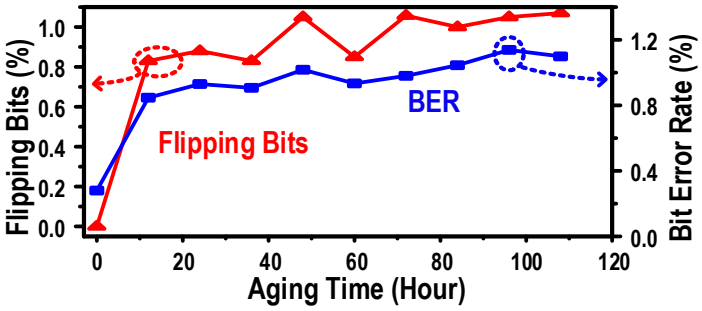


Figure 2.18 : Aging effects on BER and percentage of flipping bits.

2.4.6 Throughput and Energy Efficiency

The design reaches 8.592 Mb/s readout throughput in high performance (HP) mode and 1.459 Mb/s in low power (LP) mode. This is enabled by SRAM-style array and readout peripheral. The subthreshold operation consumes 0.076 fJ/bit core energy. The total energy including that of core and peripheral circuits is 25.6 fJ/bit in HP mode and 15.3 fJ/bit in LP mode. The throughput and energy efficiency curves versus V_{VDD} are depicted in Fig. 2.19. The throughput and energy efficiency were measured at nominal condition. Since the circuits work in subthreshold region, transistor current increases with temperature exponentially. Native transistor-based regulation suppresses supply voltage-induced influence on current.

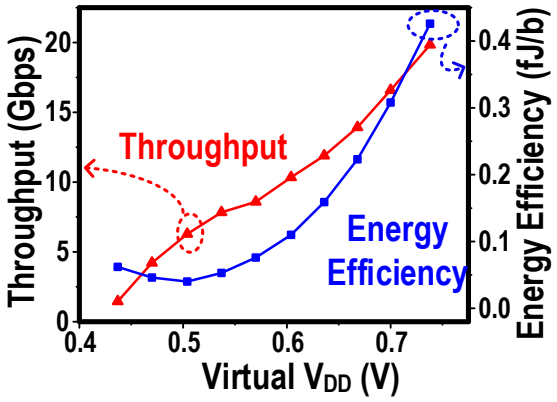


Figure 2.19 : Throughput and energy efficiency versus VVDD in nominal condition.

NIST Pub 800-90B (Draft 2) 2016)	Results of 10 chip × 4096 bits
IID Permutation	PASS
Chi-square Independence	PASS (score=1821.68, dof=2047)
Chi-square Goodness-of-fit	PASS (score=6.904, dof=9)
LRS Test	PASS (Pr=0.792)

Table 2.3 : NIST PUB 800-90B RESULTS

2.4.7 Related Works and Comparison Table

As shown in previous sections and Table V, the proposed PUF shows state-of-the-art native stability, area, and energy efficiency. The zero-overhead stabilization method further improves the stability of the PUF without the use of redundancy-based ideal masking and ECC. Emerging NVM-based PUFs using anti-fuses and RRAMs [41] [42] provides almost zero BER by randomly generating keys and storing them in memories. This class of PUF does not preserve the sensitivity to tampering as in CMOS PUFs and usually require extra fabrication steps and high testing costs. Thus, these designs are not included in the comparison table but they represent a new promising direction in PUF design.

NIST Pub 800-22 (rev. 1a, 2010)	X^2 of p-value	Average p-value	Pass Rate
Frequency	0.097	0.456	96.67%
Block Frequency	0.469	0.454	98.67%
Cumulative Sum-1	0.271	0.493	97.30%
Cumulative Sum-2	0.271	0.488	98.67%
Runs	0.350	0.499	98.67%
Longest Runs	0.686	0.509	100.00%
FFT	0.437	0.462	98.67%
Serial-1	0.630	0.497	96.00%
Serial-2	0.779	0.478	100.00%
Approximate Entropy	0.469	0.458	99.30%
Non Overlapping Template	PASS	PASS	PASS

Table 2.4 : NIST PUB 800-22 RESULTS

2.5 Summary

In conclusion, this chapter presents a self-regulated PUF based on a subthreshold inverter chain, achieving 0.3% native BER and 0.076 fJ/bit energy efficiency. Native regulation provides resistance to supply voltage variation and lead to 0.057 %/0.1 V BER sensitivity against voltage variations. The proposed in-cell reconfiguration scheme reduced native BER by two orders of magnitudes to 0.00182% with no area overhead. Moreover, a fast searching method of emulating temperature variation by sweeping body bias is applied to locate unstable bits with low testing cost. The PUF prototype in 65nm occupies only 562 F^2 per bit. Measured responses from 10 chips pass all applicable NIST 800-22 and 800-90B randomness tests and show 1020 times separation of intra/inter-die Hamming Distances after stabilization. The design achieves best-in-class metrics in all desired properties, making it suitable to provide low-cost, high-performance key generation and storage for a wide range of applications.

		THIS WORK (27°C 1.2V)	JSSC' 18 [17]	ISSCC' 18 [15]	JSSC' 17 [19]	ISSCC' 17 [18]	JSSC'16 [14]	ISSCC' 16 [16]	ISSCC'14 [11]
Technology		**65nm**	40nm	180nm	14nm	180nm	65nm	45nm	22nm
PUF Cell Area/Bit (F^2)		**562**	3643	890	9387	782	726	2613	9628
Native Unstable Bits (Evals)		**2.95% (2K)**	2.55% (500)	5.62% (1K)	~27%[f] (5K)	1.73% (2K)	6.54% (500)	-	30% (5K)
Native Bit Error Rates		**0.30%**	0.81%	0.69%	5.76%[f]	0.18%	-	0.1%[b]	8.3%
Stabilization Method		**TMV, EVB, Reconfigure**	Hysteresis, T Compensation	Remapping, TMV	Burn-in, TMV, SBD[c]	Masking, TMV	TMV, Masking	Valid Map, SMV[d], ECC	Burn-in, TMV, Masking
Unstable Bits After Stabilization[a]		**0.024%**	-	0.08%	-	0.69%	2%	-	4.6%
BER after Stabilization[a]		**0.00182%**	-	0.019%	1.46%[f]	0.08%	-	-	0.97%
Tested Conditions	**Temp (°C)**	**-55~125**	-40~125	0~80	25~110	-40~120	0~80	-25~85	25~50
	Supply (V)	**0.7~1.4**	0.8~1.0	1.2~1.8	0.55~0.75	0.8~1.8	0.6~1.2	-	0.7~0.9
Bit Errors per 10°C		**0.12%[e]**	0.32%	-	~0.17%	0.21%	0.44%	0.15%	-
Bit Errors per 0.1V		**0.057%[e]**	0.72%	-	-	0.29%	0.13%	-	0.49%
Fractional Inter-PUF HD		**0.4998**	0.4907	0.50001	0.486	0.499	0.5001	0.498	0.49
Entropy		**0.9998**	0.9972	-	0.99993	-	0.9998	0.999998	0.9997
Bit Rate (Mb/s)		**8592 / 1459[h]**	24000	0.018	-	4832	10.2	1.92	2000
Core Energy (fJ/bit)		**0.076 / 0.062**	1.02	-	4	13.5	-	-	13
Total Energy[g] (fJ/bit)		**25.6 / 15.3**	56.5	3600	-	91.1	548	-	-

a. At nominal condition, before using redundancy-based stabilization methods such as masking
b. Using glitch detection
c. Selective Bit Destabilization
d. Spatial Majority Voting
e. Results after EVB
f. Under V/T Variations
g. Including readout, timing and WL driver power
h. HP Virtual V_{DD} ~ 0.57V LP Virtual V_{DD} ~ 0.44V

Table 2.5 : COMPARISON OF SELF-REGULATED PUF WITH PRIOR PUF ARTS

Chapter 3

8-T Dual-Port CAM-Based Network Intrusion Detection System

This section presents an energy- and memory-efficient pattern-matching engine for a network intrusion detection System (NIDS) in the Internet of Things. Tightly coupled architecture and circuit co-designs are proposed to fully exploit the statistical behaviors of NIDS pattern matching. The proposed engine performs pattern matching in three phases, where the phase-1 prefix matching employs reconfigurable pipelined automata processing to minimize memory footprint without loss of throughput and efficiency. The processing elements utilize 8-T content-addressable memory (CAM) cells for dual-port search by leveraging proposed fixed-1s encoding. A 65-nm prototype demonstrates best-in-class 1.54-fJ energy per search per pattern byte and 0.9-byte memory usage per pattern byte.

3.1 Motivation and Related Work

A network intrusion detection system (NIDS) is highly desired in the Internet of Things (IoT) and wireless sensor networks, which face security issues due to heterogeneous and diverse devices, networks, and applications. A signature-based NIDS, such as Snort [43] and ClamAV [44], examines every packet, both header and payload, against expert crafted rules. Deploying NIDS capabilities on edge devices provides better traffic coverage, scalable computing resource, and higher flexibility in rule deployments than a central NIDS at a hub. But the major barrier is the limited energy

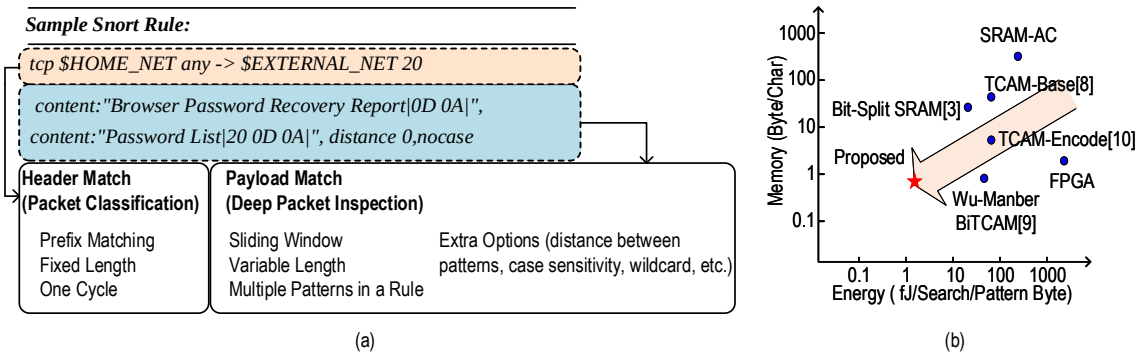


Figure 3.1 : (a) A Snort rule example, (b) Performance comparison of proposed work with prior arts.

budget for edge devices.

Matching the full payload of packets is more powerful and effective in finding malicious packets than header matching and filtering. This process is often referred to as deep packet inspection (DPI). DPI compares the streaming packet against variable-length patterns that may appear at any position. Many DPI rules have additional options including case sensitivity, wildcard matching, and specified distance between the occurrences of multiple patterns. The enormous number of patterns requires high parallelism and bandwidth to process pattern matching. The irregular memory access patterns of DPI make implementing it in von Neumann architectures inefficient.

Pattern matching is the core task in signature-based NIDS, which finds broad applications in security, database, and machine learning. Previous research has explored various hardware and algorithm optimizations, but all have faced trade-offs between energy and memory efficiency (Fig. 1b). Recently, memory centric architectures [45][46] have been proposed to handle the pattern matching problem using deterministic finite automaton (DFA)- and non-deterministic finite automaton (NFA)- based engines, for example, Micron's Automata Processor [47] is a silicon solution implementing State Transition Elements (STE) in DRAM aiming at line speed processing

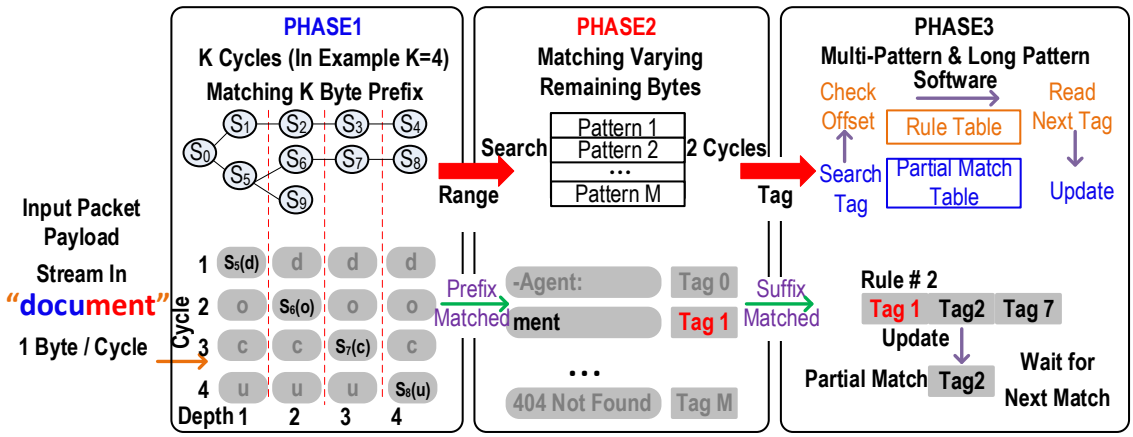


Figure 3.2 : Proposed 3-Phase pattern matching workflow with an example.

but comes with high power cost of tens of Watts. SRAM-based pattern matching engines [48][49] still suffer from area overhead due to the redundancy in most STEs. Content-addressable memory (CAM) is widely used in pattern matching for its parallel search capability. Previous studies have implemented brute-force [50], Wu-Manber [51] and Aho-Corasick (AC) pattern matching algorithms [52] in TCAM. They are high in throughput but suffer from large searching energy and a large (T)CAM footprint.

In order to improve the efficiency of edge NIDS in terms of energy, memory, and area, this chapter presents a CAM-based pattern matching engine with tightly coupled architecture and circuit innovations that exploits the statistical behaviors of pattern matching: 1) three-phase architecture; 2) memory efficient reconfigurable pipelined AC structure; 3) fine-grained CAM row enabling, power gating and clock gating; 4) dual-port 8-T CAM cell for Phase-1 processing elements (PE) and 5) 7-T CAM cell capable of single-port wildcard matching for Phase-2, enabled by a fixed-1s encoding scheme.

3.2 Reconfigurable Three-Phase NIDS Architecture

Matching NIDS rules in the proposed system is performed in three phases to leverage the statistical characteristics of pattern hit rate for power savings (Fig. 2). Phase-1 matches the prefix using an AC algorithm with a reconfigurable prefix length (depth). The AC algorithm is chosen because of its constant throughput and hardware optimization opportunities. Phase-2 matches the rest of the characters only after Phase-1 has matched the prefix. Phase-3 records the patterns matched in Phase-2 and checks for possible matches of multi-pattern and long pattern rules whenever a new match arrives. A Rule Table and a Partial Hit Table are used in Phase-3 to store transition rules for multi-pattern and long patterns and to record existing hits. It will report a "matched rule" to the user when the last pattern or "suffix" of a rule is matched. Since Phase-3 has probability lower than 2-150 of usage in most cases, it is implemented in software and off-chip for trade-offs among speed, energy and area. The 3-Phase architecture and the algorithm implemented on the microprocessor are presented in Fig. 3.

The AC algorithm is based on a Trie-style finite automata composed of states and transitions, and is constructed from the pattern library (Fig. 4a). A Trie or prefix tree, is a data structure that stores the prefixes of patterns as a dictionary in a tree for the streaming-in data to search for a match. The finite automata reads one input character per cycle and transits from the current state to the next state based on the current input. There are two types of transitions, forward and backward ones. Forward transitions handle regular prefix matching. The depth of a state is defined as its distance to the root state in forward transitions. Backward transitions occur only when no valid forward transitions exist. They are used to find other states with a shorter prefix matching the suffix of the currently matched string. Conventional

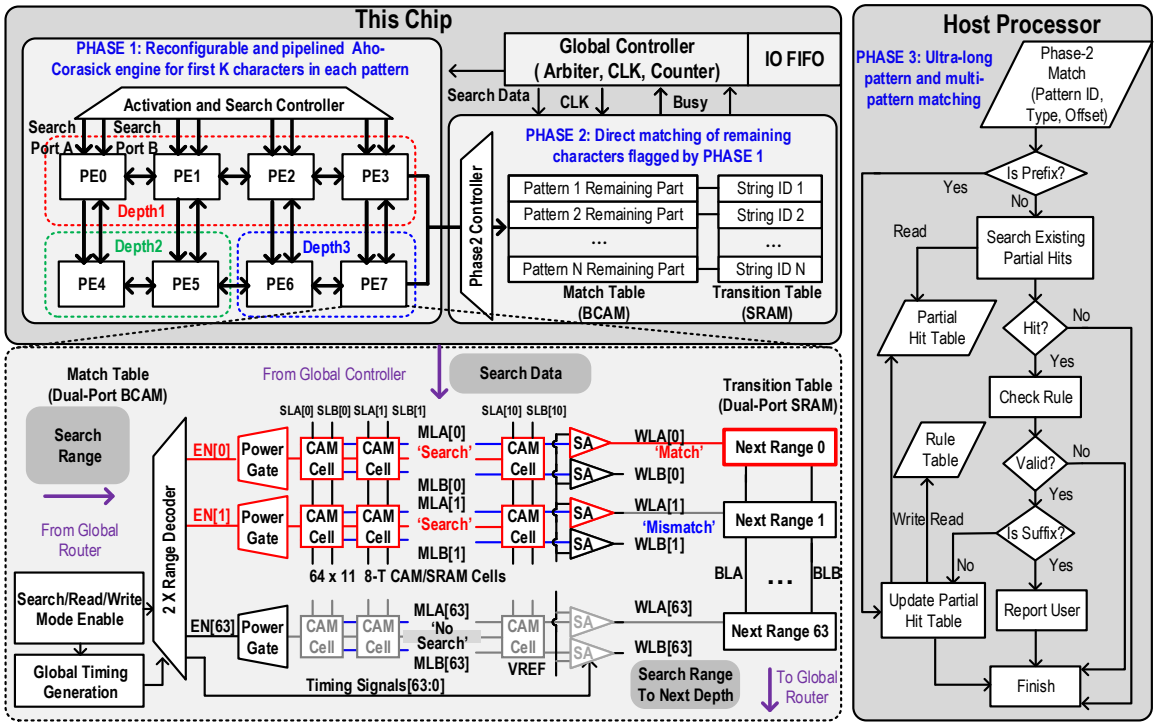
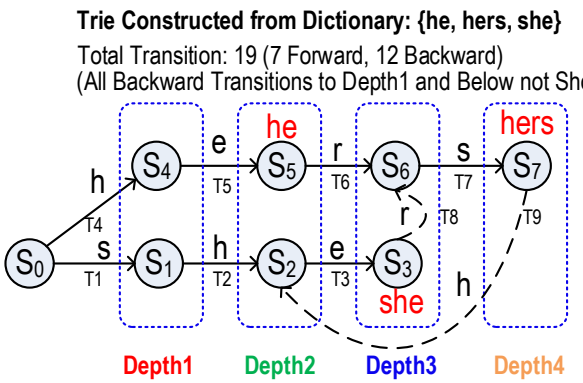


Figure 3.3 : Block diagram of proposed network intrusion detection system (NIDS) and processing element.

CAM-based AC designs [52] store current state, input character in CAM and next state in the corresponding SRAM entries. Packets streamed in are searched in the CAM for a valid transition and the index for the next state. In this approach memory usage is directly proportional to the number of transitions. Fig. 4b summarizes the transition statistics on a ClamAV ruleset [52] and an IoT-related subset of Snort. Typically, there are many more backward transitions than forward ones because the starting point of a pattern in the input is unknown. Two main drawbacks of conventional CAM-based AC design are: 1) high memory storage due to the number of transitions exponentially increasing with the number of rules, and 2) high power consumption due to a parallel CAM search and high memory volume caused by 1).

While various encoding schemes were proposed to reduce backward transitions, all



ClamAV Transition and State Depth Statistics [6]

Forward	Backward
7.6K	430K

State Depth	Count
1	56
2	89
3	100
4+	7.4K

Snort Rule Subset State Depth Statistics

State Depth	Count
1	75
2	166
3	191
4	194
5	175
6+	3.1K

(b)

Figure 3.4 : (a) An example of Aho-Corasick algorithm Trie, (b) ClamAV and Snort selected sub-ruleset transition and state depth statistics.

coding schemes required TCAMs and increase the width of state indices [52]. To completely eliminate backward transitions, we propose dividing the large match/transition table into PE clusters (Fig. 3) that store transitions at different depths in AC Trie and pipeline the execution of PE clusters. Each cluster maintains a current searching state and takes the same input byte in each cycle, as shown in Fig. 5a. As a result of pipelining, no backward transition is needed because there is always a shallower depth already at the destination state of the backward transition. While pipelining increases the total number of CAM searches by a few times, it reduces the number of transitions and the size of tables by orders of magnitude according to Fig. 4b. Furthermore, the clustering of PEs in Phase-1 is designed to be reconfigurable in terms of the number of pipeline stages and the number of PEs in each stage, which enables optimal efficiency under different rulesets and traffic conditions. The first stage in Phase-1 is implemented with an two-port SRAM, instead of the CAM-based PE, because no state matching is needed.

The searching energy is further improved by fine-grained row enabling in the

CAMs, based on the observation that only transitions to "children states" of the current state are valid candidates for which to search. Activating only the relevant rows in CAMs (Fig. 6) not only saves energy by up to 35% by spending less energy on match lines, but also reduces the width of CAM entries because the index of the current state is already contained in the range information. Thus, the CAM only stores input character, and the SRAM stores next range instead of next state. next range consists of a global PE ID to identify the PE for the next depth's search and a local UP/DN range pointing to the candidate rows for searching next transitions. Based on the global PE ID, a global router activates corresponding PEs and Phase-2 banks, and passes the detailed UP/DN range to them for the next search. Moreover, the row enabling mechanism prevents unnecessary CAM searches when there is no valid state at any depth.

Phase-2 is implemented with a single wide CAM-SRAM table without pipelining, because of the low activation rate of Phase-2. The CAM in Phase-2 supports wildcard matching for patterns with "don't care" bytes and of different lengths. CAM row enabling is still desired for Phase-2 CAM to save match line power. If a pattern is matched in Phase-2, the entire system is clock-gated to skip comparing the suffix of the matched pattern to save power (measured results in Fig. 12a).

We simulated the scaling of search energy over the ruleset size. Our proposed design with only architectural optimization (not including row enabling) and the complete co-designed system are compared with conventional methods in Fig. 5b, using the same CAM/SRAM energy models extracted from post-layout simulations. The energy of the proposed designs are by orders of magnitude lower than that of the baselines, and show much better scalability over the rule size. This is mainly because each search is confined within a small range of patterns, which is insensitive to the

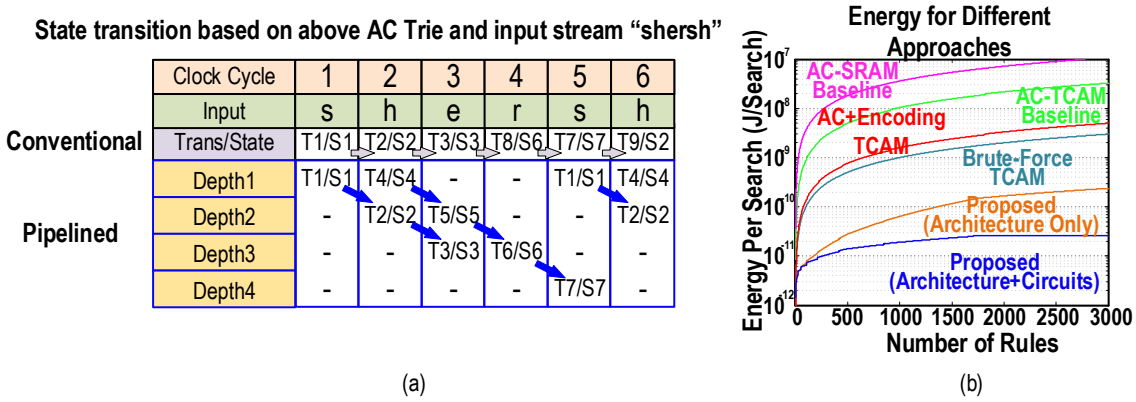


Figure 3.5 : (a) State and transition of conventional and pipelined AC algorithm on an example, (b) Simulated energy efficiency versus number of rules using different approach.

overall rule size.

3.3 Circuit Design And Implementation

Specialized CAM circuits are designed to enable an arbitrary range of the array to be searched for the proposed three-phase architecture and further to enhance the overall memory and energy efficiency. Fig. 3 presents the diagram of a PE in Phase-1, which is composed of dual-port CAM with range activation as the match table and dual-port SRAM as the transition table. Phase-2 operates in a similar manner with a PE in Phase-1, except that it demands wildcard matching.

An 8-T NOR-type CAM, inspired by the 4+2T single-port BCAM cell in [53], is designed to act as dual-port BCAM for exact matching in Phase-1 and single-port TCAM for wildcard matching in Phase-2. The extended functionality is made possible by a fixed-1s data encoding scheme. Area and power savings are thus achieved over traditional single-port 10-T BCAM and 16-T TCAM cell while maintaining the same functionality. As shown in Fig. 7, the 8-T CAM cell with two standard

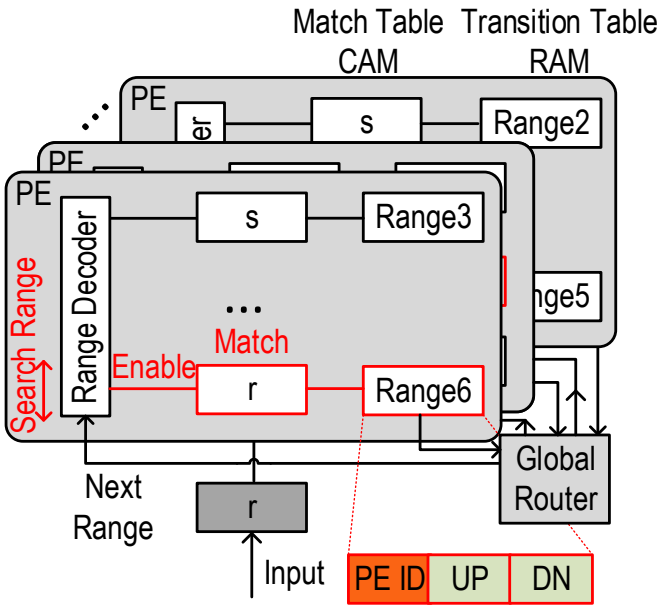


Figure 3.6 : Proposed pipelined range-matching architecture.

access transistors for writing and two extra transistors with source-controlled search lines (SLs) for matching. Read disturbance caused by BL discharge during search is avoided, which troubles 6-T CAM cell implementation [54]. The 8-T cells ensure robust writing in any technology. A 4+2T cell can be used if the body bias effect on transistors' threshold voltage is strong as in [53]. The access transistors can be eliminated in such technologies.

Normal 8-bit characters in ASCII code from the pattern are encoded to 11b fixed-1s code, each with five '1's and six '0's at different locations (Fig. 8a). The total number of '1's and '0's of each code is the same: only positions are different. Based on the pigeonhole principle, mismatch between the encoded search pattern and stored string will always lead to at least one low SL ('0') and high stored Data ('1') pair (Fig. 8b), forming a leakage path to the precharged ML (Fig. 7). In this search scheme, only one SL and search transistor are necessary. An 8-T cell with two search transistors

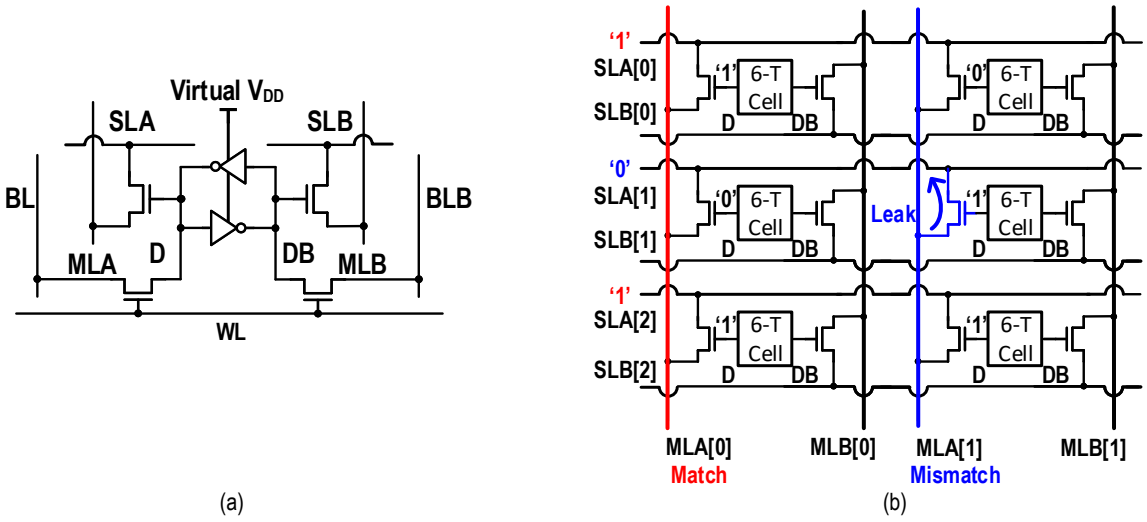


Figure 3.7 : (a) Proposed 8-T CAM Cell, (b) example search illustration, search "101" at port A, column 0 stores "101", column 1 stores "011".

can perform a dual-port search with Port B searching inverted input patterns. As a result, throughput, memory efficiency, and energy efficiency are almost doubled at the cost of using 11 bits to represent an 8-bit character.

Fine-grained row enabling selects a range of rows to search and keeps the rest power gated to the retention voltage to reduce static power and limit short-circuit power caused by shorted SLs in certain scenarios (Fig. 7b). This shorting path between SLs is a concern for power but does not significantly reduce search reliability because the search transistors are NMOS. Extensive Monte-Carlo simulations confirmed a ¿450mV sensing margin.

The core circuit for row enabling is a hierarchical range decoder (Fig. 9) based on switches, which converts the binary range (UP/DN in Fig. 6) into enable signals for all selected match lines within bounded decoding time. The switch network is divided into segments by the decoded one-hot UP and DN switches, and the boundary switches (L1_UP in level-1 and L2_DN in level-2) activate the enable and timing

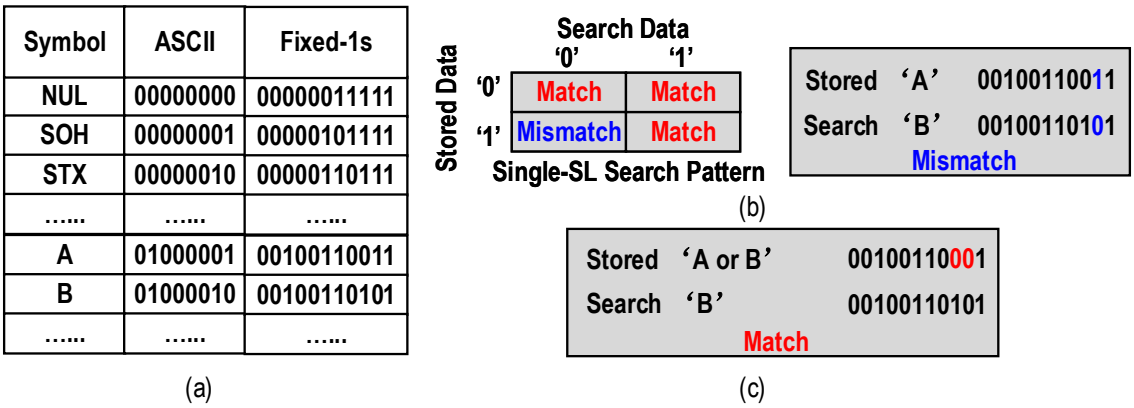


Figure 3.8 : (a) Encoding from ASCII code to fixed-1s code, (b) single side search example, (c) masking example under single side search.

signals in the selected range. For large ranges, level-1 decoder uses MSBs for a coarse range and accelerates the activation of level-2 by looking ahead for every eight rows, avoiding large RC delays caused by serially connected transistors and large parasitic capacitance when a large range is selected. The enabled range is exclusive of DN for level-1 and inclusive for level-2, leading to slightly different structures.

The proposed CAM cell and encoding scheme also support wildcard matching in Phase-2. Storing eleven '0's in the CAM will match all inputs and thus represents a "don't care" character as in TCAM (Fig. 8c). This approach can only be done on Port A and thus the other search port can be removed, resulting in a 7-T cell for single-port wildcard matching that replaces traditional 16-T TCAM cells. The associate SRAM in Phase-2 stores the matched pattern ID as well as the length of the matched patterns to facilitate clock gating after matching.

The Phase-2 search has a single port and takes two cycles, which is slower than the single-cycle and dual-port search in Phase-1. Data congestion can thus occur when Phase-1 reports intensive matches. Congestion includes two situations: two prefix-matches from Phase-1 in two consecutive cycles, and two prefix-matches from

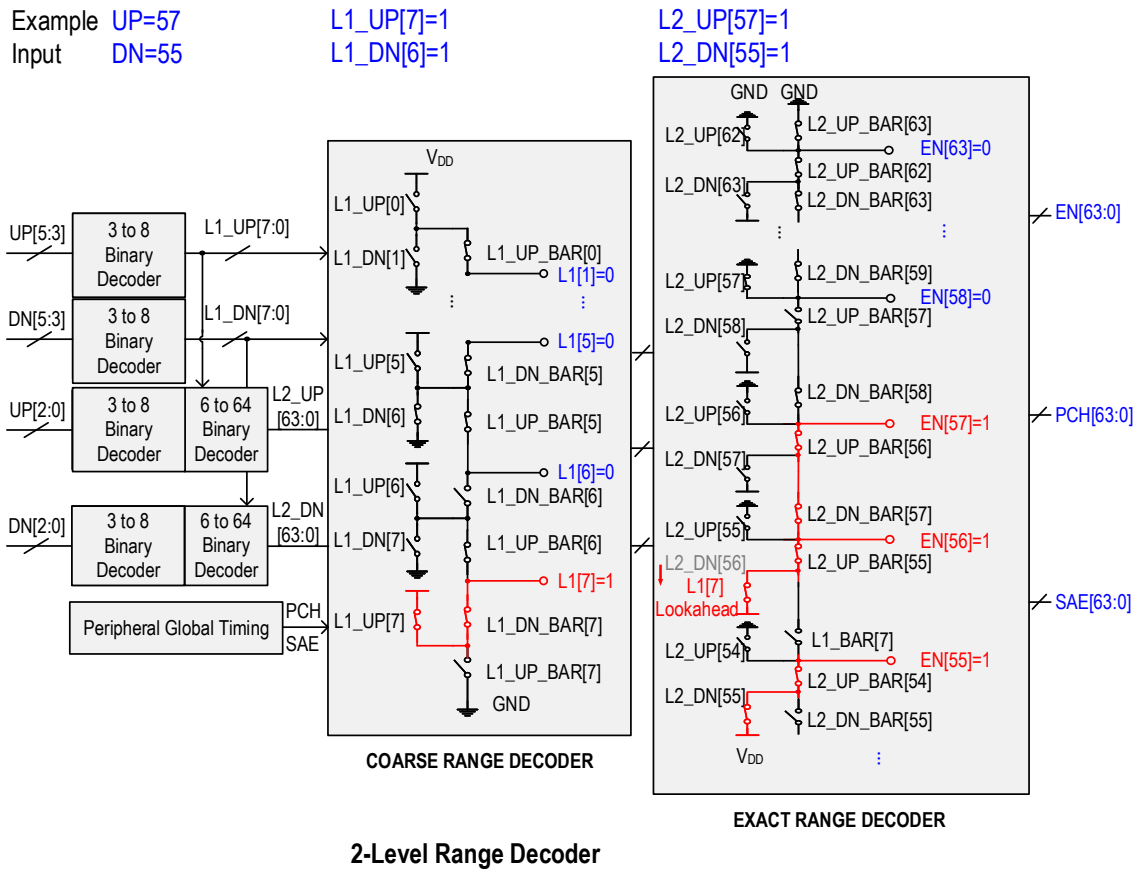


Figure 3.9 : 2-level range decoder and hierarchical clock gating.

Phase-1 from two ports simultaneously. An arbiter is developed to handle various situations of congestion and higher priority is given to port A.

3.4 Measurements

A prototype chip is fabricated in 65-nm CMOS LP process. The chip micrograph is shown in Fig. 10a. In this prototype chip, a separate SRAM for Stage1 and eight PEs are implemented for Phase-1. Each Phase-1 PE contains 64 rows of 8-T CAM and SRAM. Phase-2 includes four 64x220 7-T CAM banks and associate SRAM banks. A 240-rule Snort subset is used for testing. When a rule pattern is found, the engine

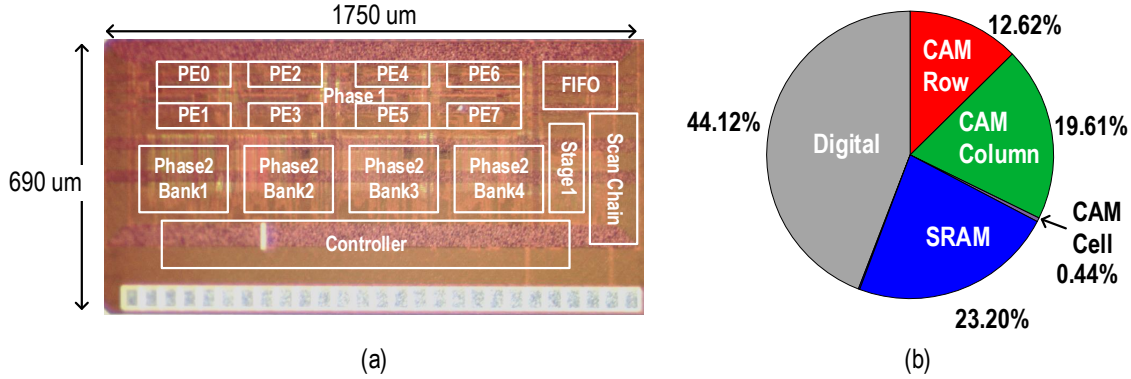


Figure 3.10 : (a) Chip micrograph, (b) Power breakdown at full workload.

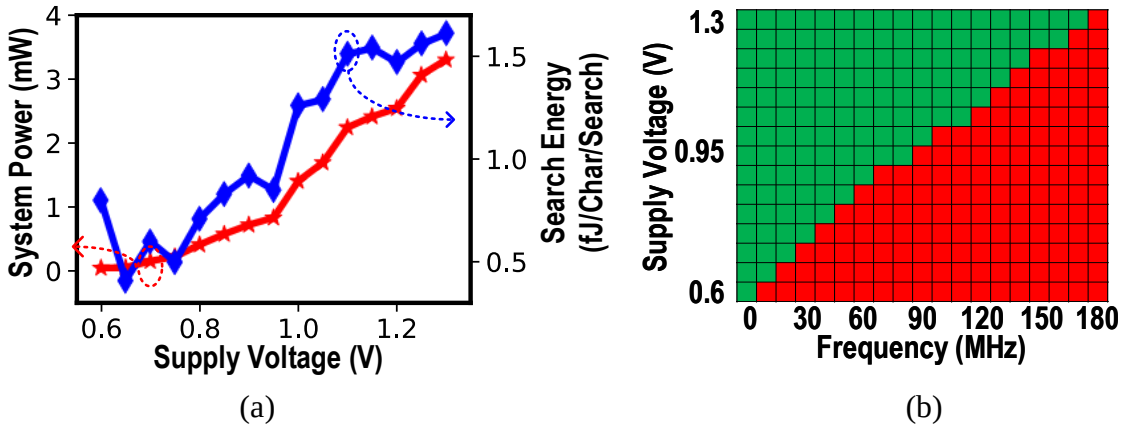


Figure 3.11 : (a) System power and search energy versus supply voltage(b) Shmoo plot of the system.

will report a "hit" and push the pattern ID to a buffer for Phase-3 processing in host processor. Input packets are randomly generated with controlled pattern hit rate for energy measurements.

For fair comparisons with other designs, energy efficiency is defined as the energy per search divided by the total number of characters in the ruleset ("Char"). The energy efficiency varies from 1.54 to 2.1 fJ/Char/search at 1.2 V and 144 MHz depending on pattern hit rate (Fig. 11a). The more the malicious patterns in packets,

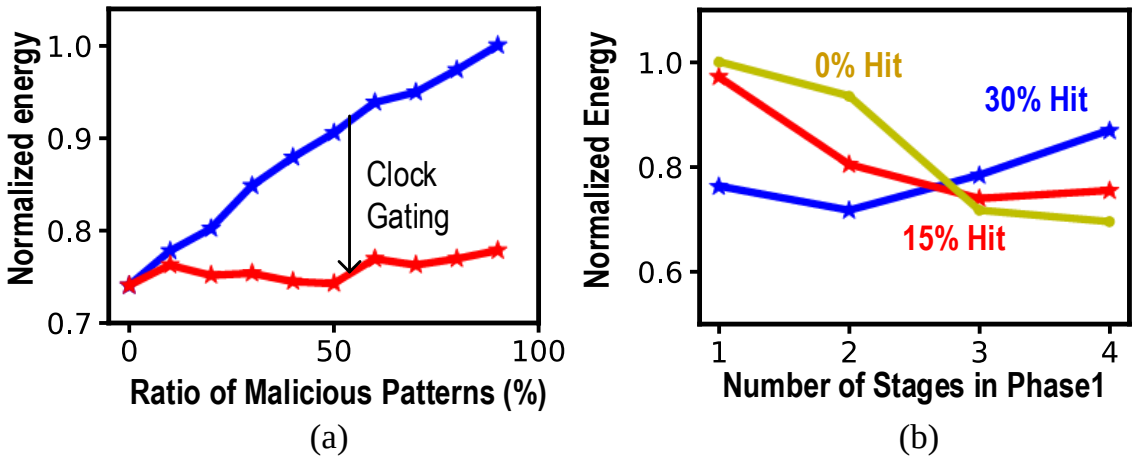


Figure 3.12 : (a) Normalized search energy versus ratio of malicious patterns (hit rate) with and without clock gating, (b) Normalized search energy versus number of stages in Phase-1 under different attack rate.

the more frequently Phase-2 will be activated. Phase-2 banks consume higher overall energy for a single search due to their larger capacity. Fig. 11b confirms that the energy consumption also depends on Phase-1 configurations and the plausibility of reconfigurability. Because of the higher Phase-2 energy, it is beneficial to avoid Phase-2 searches by including more stages in Phase-1. However, when the percentage of malicious packet rises above a certain threshold, it becomes more efficient to reduce the prefix depth of Phase-1 and pass more packets to Phase-2. Moreover, Fig. 11a shows that clock gating after hits saves up to 25% power and renders the overall efficiency less dependent on the hit rate. Fig. 10b presents the power breakdown in one scenario, when both ports are activated, Phase-1 is configured with four stages, and an input stream with 90% malicious packets. Fig. 12a plots the chip's voltage dependence of power consumption. The system shmoo plot is shown in Fig. 12b. Comparisons with related prior arts, including accelerators for DPI, header matching, and generic TCAMs, are listed in Table. I.

3.5 Summary

This chapter presents an energy- and memory- efficient CAM-based payload matching engine for NIDS in IoT [55]. Leveraging holistically designed architecture and circuits, a 65-nm prototype achieves best-in-class 1.54-fJ/search/pattern byte and 0.9-byte/pattern byte.

	This Work	HPCA 2020 [6]	MICRO 2017 [7]	JSSC 2009 [9]	ISCA 2005 [3]	ICNP 2004 [8]	ICNP 2006 [10]	ASSCC 2019 [17]	VLSI 2018 [14]	JSSC 2018 [15]	JSSC 2004 [13]	JSSC 2005 [12]
Application	Payload Match	Payload Match	Payload Match	Payload Match	Payload Match	Payload Match	Payload Match	General Purpose CAM	General Purpose CAM	Header Match	Header Match	Header Match
Architecture	**Pipeline Selective Enable**	AC SRAM	AC SRAM	Wu-Manber BiTCAM	Bit-Split FSM SRAM	Brute-Force TCAM	AC, Transition Compression TCAM	6-T CAM ML Clamping, Footer	Half and Half Compare TCAM	Don't Care Reduction CAM	Pipelined Hierarchical CAM	Hierarchical CAM
Technology	**65nm**	14nm	28nm	130nm	130nm[a]	65nm[b]	65nm[b]	28nm	14nm	65nm	180nm	100nm
Frequency (MHz)	**144**	5000	1200	380	720	250	250	369	1470	330	143	300
Throughput (MBps)	**288**	80000	9400	380	720	250	250	369	1470	330	143	300
Latency[f] (ns)	**41.7**	4	16.7	5.3	27.8	4	80	2.708	0.68	3	7	3.3
Power (mW)	**2.27**	22000	1080	131.22	4720	-	-	0.066 (10 MHz)	0.032 (10 MHz)	46.7	60.8	31.1
Memory Efficiency[c] (Byte/Char)	**~0.9 CAM ~0.6 SRAM**	12.8 SRAM	17.2 SRAM	0.66 CAM 0.62 SRAM	18 SRAM	2.5 CAM 20.6 SRAM	4.75 CAM 2 SRAM	1	1 CAM	0.34 CAM	1 CAM	1 CAM
CAM Energy Efficiency[d] (fJ/Char/search)	**0.61**	-	-	27	-	15.4	32	12.96 (10 MHz)	3.04 (10 MHz)	3.28	23.12	5.6
DPI Energy Efficiency[e] (fJ/Char/search)	**1.54**	22	26.8	43.2	293.4	40.4	44.5	-	-	-	-	-
Cell Area (um^2)	**1.33**	-	-	10.32	-	-	-	0.362	1.47	~5.37	-	22.4

a. Modeled by [9]
b. Modeled using the technology by this work
c. Defined as total CAM byte on chip/total pattern bytes
d. Defined as energy of CAM/total pattern bytes per search
e. Defined as energy of system/total pattern bytes per search
f. Defined as delay time from input to output of an exact match

Table 3.1 : COMPARISON TABLE OF THE OUR CAM-BASED NIDS WITH PRIOR NIDS AND CAM WORKS

Chapter 4

MeNTT: A Compact and Efficient In-Memory Number Theoretic Transform Accelerator

Lattice-based cryptography (LBC) based on Learning with Errors (LWE) problems is a promising candidate for post-quantum cryptography. Number theoretic transform (NTT) is the latency- and energy- dominant process in the computation of LWE problems. In this section, we present a compact and efficient in-MEmory NTT accelerator, MeNTT, via optimized computation in and near a 6T SRAM array. Specifically-designed peripherals enable fast and efficient modular operations. Moreover, a novel mapping strategy reduces the data flow between NTT stages into a unique pattern, which greatly simplifies the routing among processing units (i.e. SRAM column in this work), with reduced energy and area overheads. The accelerator achieves significant latency and energy reductions over prior arts.

4.1 Motivation

Most of the contemporary public-key crypto-systems like RSA and elliptic curve cryptography (ECC) rely on difficulty to solve integer factorization and discrete algorithms. However, with the advent of quantum computing and it's implementation using quantum computers much closer to the reality now, these problems are expected to be solved by Shor's algorithm [56] in polynomial time. Therefore quantum resistant crypto algorithms are being developed by researchers around the globe. Lattice based Cryptography (LBC) have emerged as prime candidates among the post

quantum protocols.

Lattice based protocols have come to light because of hardness of inherent Learning with Errors (LWE) problemss [57]. Modular LWE and Ring LWE are two primary variants of LWE problem. Ring LWE secures the message using polynomial operations between secret key and public key along with error addition. Polynomial multiplication is performed using NTT [58], a FFT like structure except that operations performed are modular arithmetic. NTT proves to be the bottleneck of computation in Ring LWE and occupies more than 70% of the area[mit] in ASIC accelerators. With N being power of 2, time complexity for computing NTT is $O(n \log n)$. Theoretically, N by 2 modulo multiplications can be performed in parallel for each NTT. But, it requires the accessing elements in parallel which proves to be the bottleneck. Most of existing NTT designs in FPGA [59] or digital ASIC have been optimizing the dataflow to enhance the overall performance and efficiency for a certain configuration of butterfly units and memory banks. The hardware configuration underlines a fundamental trade-off between area and efficiency. For example, [60, 61] minimize the area overhead by sequentially access and share a butterfly unit, while [62] attempted to layout and operate several butterfly units in parallel. Both the approaches couldn't completely address the issue of desiging compact and parallel accelerator.

Processing in-memory (PIM) [63] is a promising technology for memory-constrained computation, owing to its capabilities of highly parallelized computing with amortized energy for memory accesses and logic operations. PIM also enables enhanced data locality, avoiding frequent data transfers to and from the computing units. These additional capabilities while retaining the compact nature of the memory makes it a promising technology for accelerating NTT [64].

While PIM accelerators based on beyond-CMOS memory devices, such as ReRAM

or MRAM, hold great promise for future memory-centric computing with superb density and non volatility. SRAM-based PIM accelerators solely based on mainstream CMOS technologies undoubtedly represent a clearer path towards reliable mass productions and robust operations. SRAM PIM allows low-voltage read, write, and logic operations for energy savings, and could always take advantage of the latest CMOS processes. Further, the larger footprint of SRAMs is amortized by the relatively large PIM peripherals in practical implementations.

In this chapter, we present MeNTT, an in-6T-SRAM NTT accelerator with boosted area and energy efficiency. Our key contributions include:

- We develop novel 6T SRAM peripherals and a protocol to perform bit-serial modular addition/subtraction/multiplication arithmetic in 6T SRAM array with massive parallelism and less steps.
- We propose a mapping strategy to optimize the dataflow between NTT and INTT stages and dramatically reduce the routing overhead for large NTT/INTT.
- The proposed MeNTT accelerator is evaluated using a combination of transistor-level post layout simulation for the memory and synthesized netlist for the digital logic, all of which are designed and simulated in TSMC 65nm LP process.
- While the MeNTT implementations in this work are solely based on CMOS devices, the proposed modular arithmetic protocol and dataflow techniques can be easily extended to PIM with emerging memories.

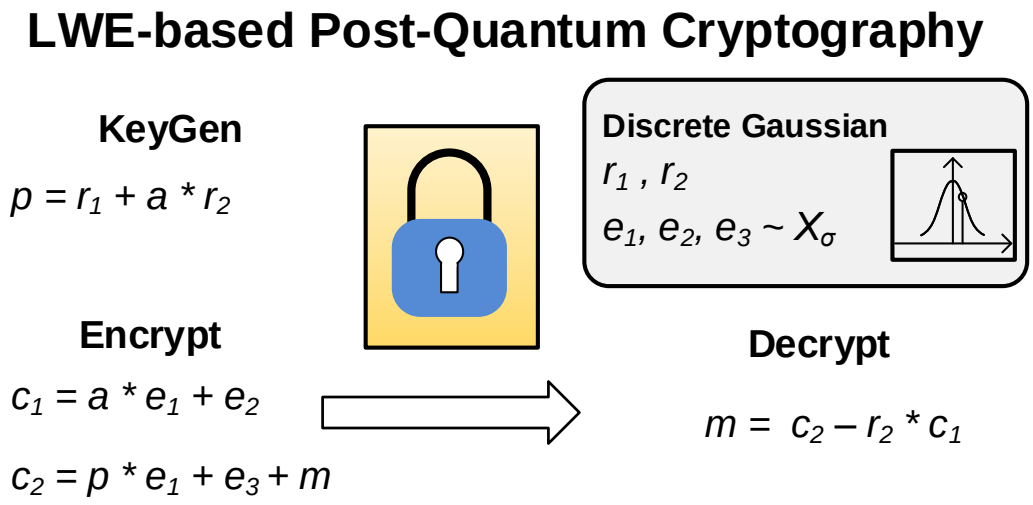


Figure 4.1 : The basic scheme of lattice-based cryptography.

4.2 Background and Related Work

In the section we will discuss mathematical background of the Ring LWE problem and the NTT/INTT algorithm based polynomial multiplication. We also provide a brief overview on recent progresses of PIM, especially those for cryptographic accelerations.

4.2.1 Ring Learning with Errors (LWE)

Let the pair of vectors **(a,b)** be related by the equation $b = a * s + e$ where **a** is a randomly sampled vector from R_q, **e** is a error vector sampled from a gaussian distribution. Here $R_q = Z_q[x]/(x^n + 1)$ is a ring of polynomilal where n is a power of 2 [65], q is a prime number. Ring LWE [66] states that it is difficult to find secret vector $\mathbf{s} \in R_q$ given the pair (**a,b**). Here a*s is a polynomial multiplication and is performed by transforming both a and b in NTT domain.

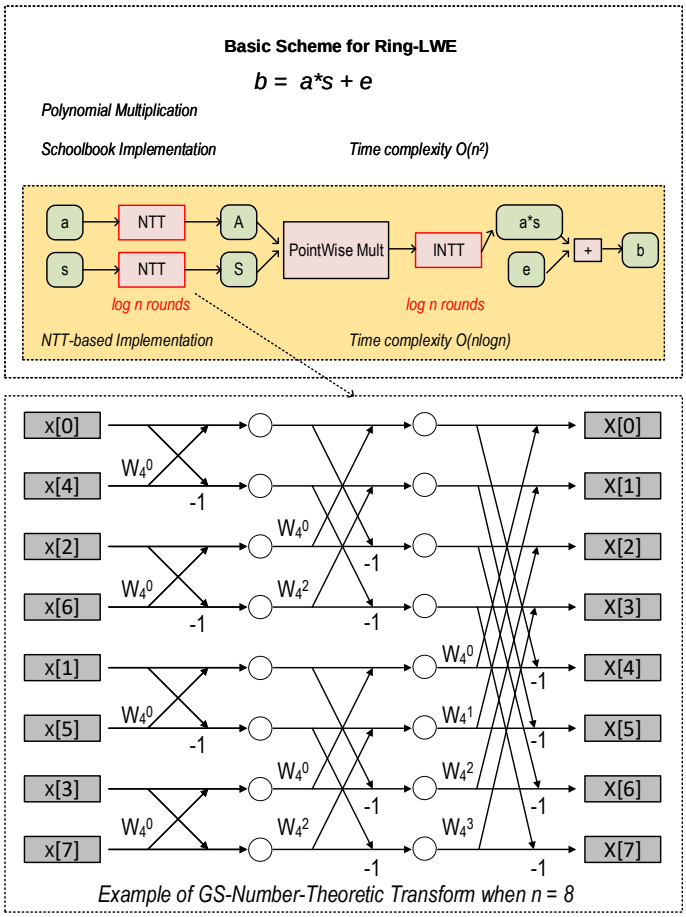


Figure 4.2 : The overall flow of Ring-LWE and polynomial multiplication in Ring-LWE.

4.2.2 Number Theoretic Transform (NTT)

Brute-force polynomial multiplication would take $O(n^2)$, while computing in NTT domain would take $O(n \log n)$. Let a,s are two polynomials sampled from R_q, whose coefficents are in range [0,q) where q is prime number. We denote NTT coefficients of polynomial $(a = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + ... + a_0)$ as $\hat{a}_{n-1}, \hat{a}_{n-2}, .., \hat{a}_0$ respectively.

$$b = INTT((NTT(a) * NTT(s)) \tag{4.1}$$

Algorithm 1 NTT Multiplication with Cooley-Tukey Method

Require: Given polynomial $a \in R_q$,and n-th root of unit $w_n \in Z_q$
Ensure: Polynomial $\hat{a} = NTT(a)$ such that $\hat{a} \in R_q$

```
for (stage =1; stage≤ log2 n; stage = stage+1) do
    m ← 2^stage
    w_m ← w_n^(n/m)
    for (k =0; k < n; k=k+m) do
        w ← 1
        for (j =0; j < m/2;j=j+1) do
            t ← w.â[k + j + m/2] mod q
            u ← â[k + j]
            â[k + j] ← u + t mod q
            â[k + j + m/2] ← u − t mod q
            w ← w.w_m mod q
        end for
    end for
end for
Return â
```

Polynomial multiplication of b=a*s is done using eq 4.1.After tranforming both vectors a and s in NTT domain, NTT coefficients of b is calculated by coefficient wise multiplication of a and s. Original coefficients are calculated back by transforming $\hat{b}_i$ using Inverse NTT.

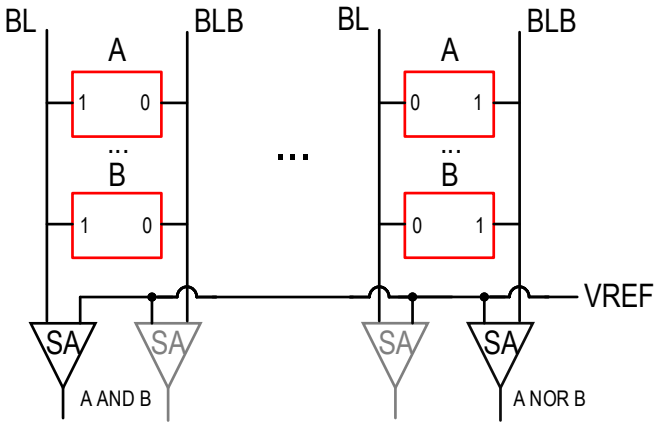
$$b = \sum_{i=0}^{N-1} (\hat{a}_i * \hat{s}_i) x_i \tag{4.2}$$

There are two variants of butterfly optimizations popular for the acceleration of polynomial multiplication: Cooley-Tukey and Gentleman-sande. The former is adopted in MeNTTand described in Algorithm 1. INTT is similar to NTT except that the twiddle factors are w^{-n} instead of w^n.

4.2.3 Processing in Memory for Cryptographic Acceleration

Similar to many other memory-centric computation problems, such as deep learning, NTT computing faces the "memory walls" because of the energy and throughput bottleneck between logic and memory [67]. To alleviate this problem, a new computing paradigm with in- and near-memory computing emerged to reduce data movement, amortize memory access energy, and energy-efficient mixed-signal logic operation within a memory array [63].

In the domain of cryptographic computing, the requirement on PIM is different from that of machine learning applications because of its zero tolerance to compute errors. Thus, bit-parallel and bit-serial operations are more suitable than the lossy computing mechanisms in current, charge, or voltage domains [68, 69, 70]. Bit-serial in-SRAM logic performed by accessing two words simultaneously (Fig. 4.3) is first utilized to modern cryptography accelerators by [71], which performs very wide word bitwise logic and finite field arithmetic in memory. However, it does not provide high parallelism expected for NTT acceleration. Performing arithmetic in bit-serial fashion was introduced in [72], which is promising for NTT and modular arithmetic because it achieves full accuracy computing with massive parallelism, by spending more clock cycles in each arithmetic operation. [64] leveraged a similar technique with projected high-performance and high-density ReRAM devices. It employs an unfolded architecture with straightforward control and routing schemes, which is possible with high-density ReRAM devices and slim peripherals, but will lead to an unacceptably large area if implemented on SRAM. On the other hand, recent ASIC NTT accelerator [62] have tried computing multiple butterfly operations in parallel and moving buffers closer to the computing units, the extensive use of registers for local buffering comes with a high area overhead. Thus, achieving the desired in-memory computing



A	B	BL	BLB
0	0	0	1
0	1	0	0
1	0	0	0
1	1	1	0

BL = A & B BLB = ~ (A | B)

Truth Table of Processing-in-SRAM

Figure 4.3 : Basic bit-wise logic operation in 6T SRAM by accessing two rows simultaneously.

performance and efficiency gains with compact physical footprint and cost is thus the primary goal of this work.

4.3 Data Storage and Arithmetic Flow in MeNTT

MeNTT performs complete polynomial multiplication computation in NTT and INTT with all the modular arithmetic steps in and near a 6T SRAM array, as shown in Fig. 4.4. The SRAM array functions as data storage as well as computing unit. Inter-column router route data at end of one round into corresponding columns for the computation of next stage and round. SRAM peripheral and controller are specifically designed for in- and near-memory modular arithmetic. During each round in

the radix-2 NTT, each column works as a butterfly unit, thus enabling massively paralleled computing to support large number of points. The operands and intermediate results are all stored in the same array with an allocation strategy shown in Fig. 4.5. As a result, the width n of the array represent the maximum number of points, while the number of rows is related to the supported bitwidth, as shown in Fig. 4.5. The modular addition, subtraction and multiplication are performed in a bit-serial manner, similar to the generic bit-serial logic achieved in [72], but with significantly reduced steps and energy optimized for modular arithmetic. The high parallelism enabled by the bit-serial approach improves the overall performance and energy-efficiency of NTT operations with large number of points. In our implementation, a single SRAM bank with 162 by 1024 cells is designed to support at most 1024-point and 32-bit NTT operations. The physical layout of the wide SRAM array can be folded as shown in Fig. 4.4, in order to reduce word line and inter-column routing length, and maintain a proper aspect ratio. 6T SRAM is desired because of its maturity and high density.

Modular arithmetic differs from regular arithmetic as more attention is paid to overflow and underflow issue. Most previous works take the strategy to calculate the result in integer field, then reduce it to the desired finite field. Barrett reduction and Montgomery reduction (Algorithm. 1, Algorithm. 2) are two most common approaches to reduce numbers into finite field. MeNTT proposes a reduce-on-the-fly technique for modular addition, subtraction and multiplication. The parallel bit-serial operation utilizes bit-serial comparison by deploying a serial comparator (Fig. 4.6). The comparison starts from LSB to MSB cycle by cycle as the bit-serial addition, subtraction and multiplication take place. Reduction is applied based on the comparison result.

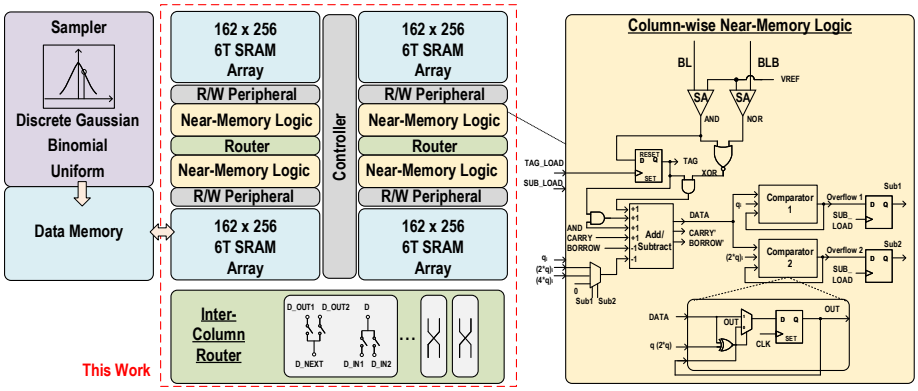


Figure 4.4 : Diagrams of MeNTT and custom circuitry for the key peripherals.

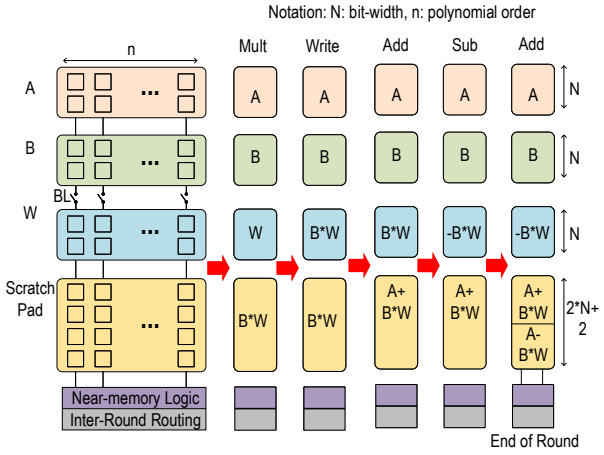


Figure 4.5 : Data arrangement and operation sequence in an NTT round.

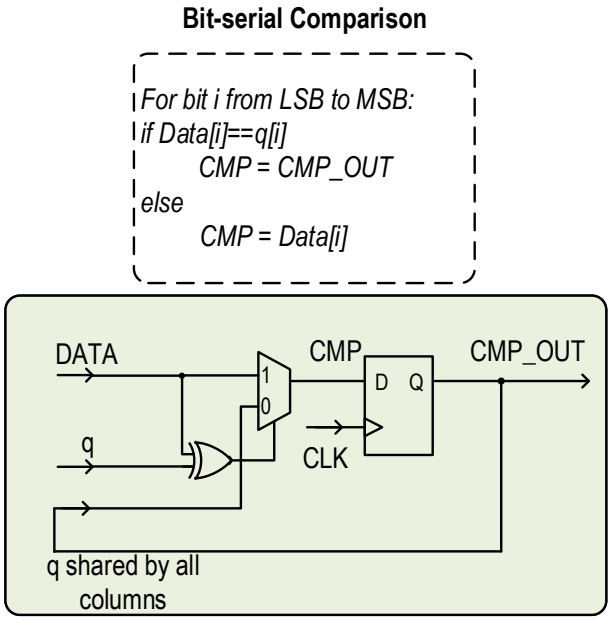


Figure 4.6 : Bit-serial comparator operation in column peripheral.

4.4 Proposed Modular Addition/Subtraction in Memory

The proposed modular addition and subtraction consist of a normal round of addition/subtraction with overflow/underflow detection called "Trial Add and Compare Phase", and a round of modular reduction called "Modular Addition Phase", as depicted in Fig. 4.7. The WL driver activates corresponding bits of two operand A and B sequentially. The value read on BL will be A AND B, while the value on BLB will be A NOR B. The peripheral maintains one-bit Carry for the addition, and acts like a full adder which adds A and B up sequentially and writes back to the array. In "Trial Add and Compare Phase", in the meantime of addition, the peripheral reads one bit of q, and make comparison sequentially. If the sum of A and B is larger than q, then an overflow bit is set. Then in "Modular Addition Phase", the column will calculate A+B-q if the overflow bit is set, and A+B if the overflow bit is not set. The column will calculate A+B if overflow bit is not set in "Modular Addition Phase" to

avoid timing side-channel leakage.

Subtraction will be conducted in a similar approach. The operation will be performed by first transforming B into its 2's complement. This is done by setting the initial Carry to 1 and use the value from BLB as input. During the time of normal subtraction, an underflow bit is recorded. Then a second-round subtraction with conditional addition of q depending on the underflow bit is performed to keep the result correct. Addition takes 2*(N+1) cycles and subtraction takes 3*(N+1) cycles.

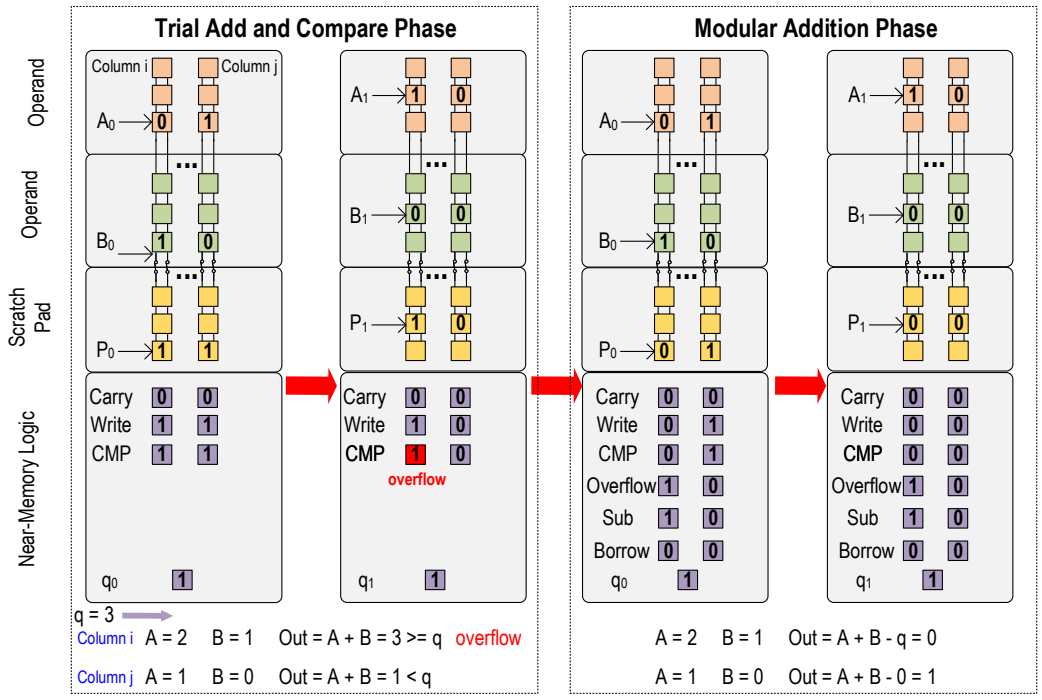


Figure 4.7 : Illustration of in-memory two-bit modular addition.

Algorithm 2 Barrett Modular Multiplication

Require: : $x, y \in Z_q,\ q, m$ *and* k *such that* $m = \lfloor 2^k/q \rfloor$
Ensure: : $z = x \ mod \ q$

$z \leftarrow x \cdot y$
$t \leftarrow (z \cdot m) \gg k$
$z \leftarrow z - (t \cdot q)$
if $(z \geq q)$ **then**
    $z \leftarrow z - q$
end if
Return z

Algorithm 3 Montgomery Modular Multiplication

Require: : $x, y \in Z_q,\ q, r$ *and* k *such that* $r > q,\ gcd(r, q) = 1,\ k = \frac{r(r^{-1} mod\ n)-1}{n}$
Ensure: : $c = x \ mod \ q$

$\overline{a} = a \cdot r \ mod \ q$
$\overline{b} = b \cdot r \ mod \ q$
$x = \overline{a} \cdot \overline{b}$
$s = x \cdot k \ mod \ r$
$t = x + s \cdot q$
$u = \frac{t}{r}$
if $(u \geq q)$ **then**
    $\overline{c} = u - q$
else
    $\overline{c} = u$
end if
$c = (\overline{c} \cdot r^{-1} mod\ n)$
Return c

4.5 Proposed Bit-Serial Modular Multiplication in Memory

Modular multiplication is the most time-consuming part of the NTT. Traditionally, modular multiplication is completed by first computing the raw product, then going through Barrett or Montgomery reduction(Fig. 4.9). In previous PIM works [72], this will involve three normal multiplications and take many more redundant space and cycles. Although for some special cases [64], the reduction can be simplified. There is no general optimization for the modular multiplication using traditional reduction

Algorithm 4 Modular Addition

Require: $x, y \in Z_q$
Ensure: $z = x + y \ mod \ q$
1: Trial Add and Compare Phase:
2: $s \leftarrow x + y$
3: $cmp \leftarrow s \geq q$
4: Modular Addition Phase:
5: **if** $(cmp = 1)$ **then**
6: $\quad z \leftarrow x + y - q$
7: **else**
8: $\quad z \leftarrow x + y$
9: **end if**
10: Return z

approach.

This work proposes a fast bit-serial multiplication scheme to complete a modular multiplication in $(N+1)^2$ cycles (Fig. 4.8). The overall algorithm is described in Algorithm 5. The multiplication is decomposed into shift-and-add operations. There is a Tag bit which is similar to the approach of [72]. The shift is done by choosing different address for rows in each round. The addition is controlled by the Tag bit from operand B and computed in the peripheral with carry and borrow. We make use of the nature of bit-serial computation and compare the partial sum with q and 2*q. A reduction will be performed in the next cycle to make sure the partial sum is contained in the correct range. The reduction is enabled by the subtraction and borrow circuits, and controlled by the two overflow bits. In this way, the modular multiplication can be completed in parallel with the shift-and-add itself with very little space and time overhead.

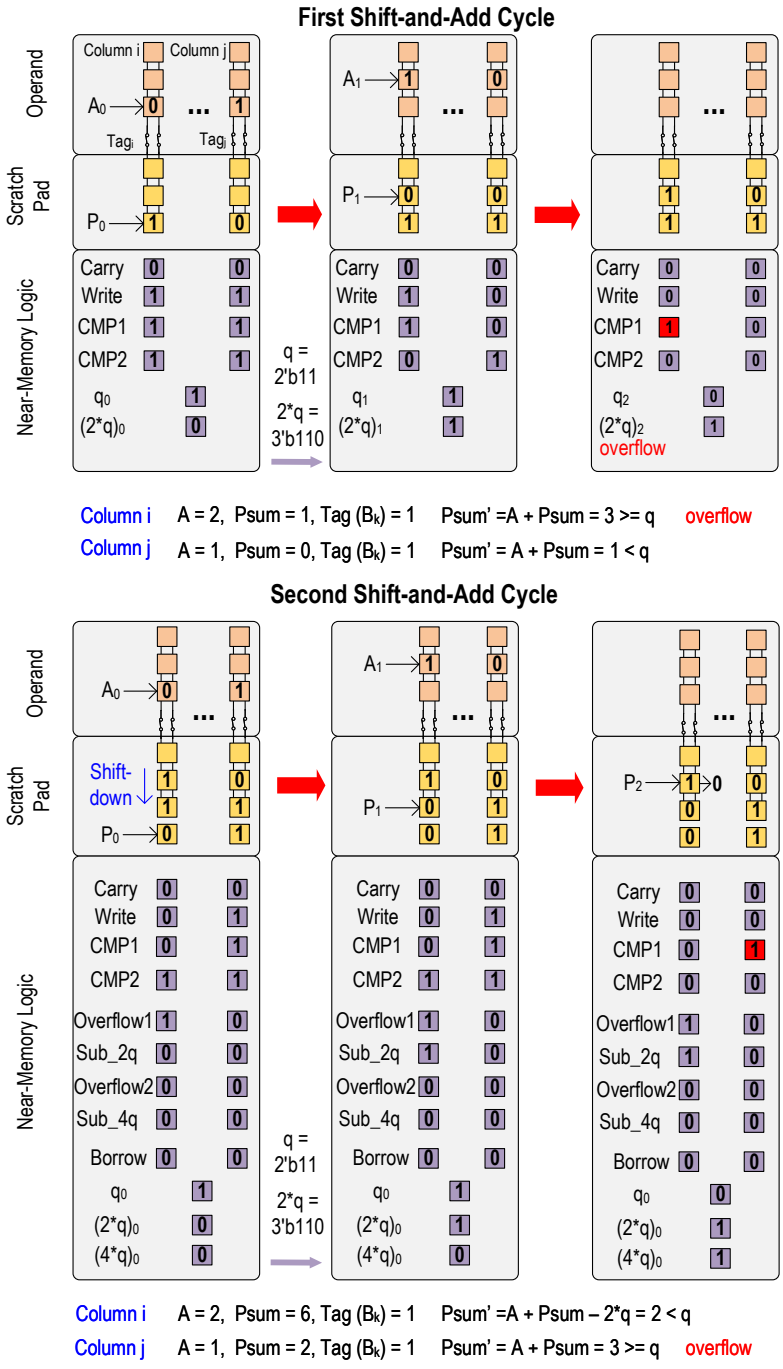


Figure 4.8 : Illustration of in-memory 2-bit modular multiplication (q is 2'b10, Tag is 1 for two cycles).

In order to make the algorithm correct, the operand A has to be smaller than

Complexity / Method	Multiplication	Cycles
Barrett	3	~$5N^2$
Montegomery	3	~$6N^2$
Proposed	1	~N^2

Figure 4.9 : Comparison of traditional modular multiplication methods and proposed method.

q/2. Therefore we use 2*q instead of q as the field constant. Then MeNTT reduce the partial sum below q with an extra round. A switch is controlled by the Tag to separate the operand A from influencing the BL readout result. It decides whether the partial sum is added by an operand or kept the same. The reduction with q or 2q is done in peripheral with simple digital logic shown in Fig. 4.4.

4.5.1 NTT snd INTT Dataflow

Fig. 4.2 shows the CT radix-2 style NTT for the acceleration of polynomial multiplication in a ring. Each round of NTT contains groups of butterfly operation which can be decomposed to a multiplication, an addition and a subtraction. Each round has own grouping of points depending on the index.

Intra-round flow

In a single round, the data arrangement is shown in Fig. 4.10. The N-bit operands A and B are stored sequentially in same column followed by the twiddle factor W.The results of addition, subtraction and multiplication are computed and stored in the scratchpad area. Therefore the temporary results for an NTT round can be computed in sequential order and updated into the operand area for follow-up computations. All operation of addition, subtraction, copying, inverting and multiplication required for

a round can be completed inside a single column in a bit-serial manner. Considering the algorithm for modular multiplication requires extra bits for each column, the size of scratchpad is set to 2*N+2. The space can be fully utilized to store both A+W*B and A-W*B, which is the final output for this round.

Inter-round data movement

Once the computation of a single round is finished, the data need to be read out and written into the operands area for next round. Traditional in-place NTT has complicated data routing, varying from round to round. This makes the data movement different in different rounds of NTT and INTT. A configurable crossbar routing is required to enable various address matching, adding to the area and energy overhead.

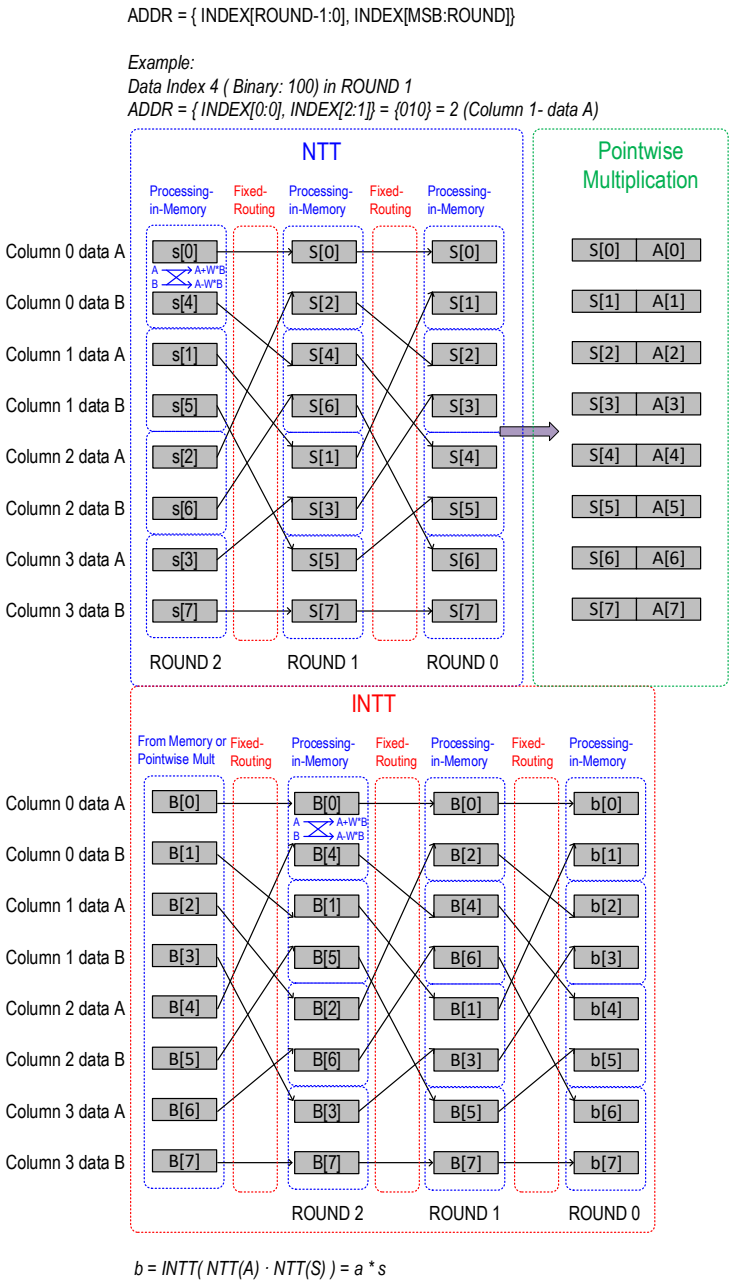


Figure 4.10 : Proposed MeNTT dataflow enabled by a new mapping strategy.

We propose a mapping strategy that makes the data movement between rounds constant. As shown in Fig. 4.10, the data are stored in a specific column with address. The physical address is a function of the current round and its original

index, as described below:

$$addr = \{index[round - 1 : 0], index[msb : round]\} \tag{4.3}$$

For example data[4] in Fig. 4.10 has index '100'. In round 1 of NTT, the actual address is {'0','10'} according to the mapping, which leads to addr of 2, translating to Column-1, operand A. The key observation for this approach is that the actual physical output-input address mappings are the same for each round. For example, the output of address 3 always goes to the input address 1 in every round. Therefore the crossbar connections can be reduced to simple switches. This schedule make it possible to construct a single-bank processing-in-memory block to compute the NTT process iteratively. INTT follows a similar mapping strategy and shares the same physical routing. The pointwise mulpication between NTT and INTT can be completed in place. In such a scheme, the whole polynomial multiplication operation is performed in a single SRAM bank with bit-wise modular arithmetic and constant round to round and stage to stage routing. The MeNTT is therefore highly area-efficient.

4.6 Evaluation and Discussion

We evaluated MeNTT with TSMC 65nm LP CMOS PDK. Post-layout SPICE simulation results for the SRAM are combined with energy and area estimation of digital circuits from Design Compiler to have an accurate estimation of the system. Different configurations of bitwidth and polynomial order are evaluated to compare our work with prior arts for the different protocols. Evaluation results are shown in Fig. 4.11, Fig. 4.12, Fig. 4.13 and Fig. 4.14. MeNTT is configurable for different bit-width

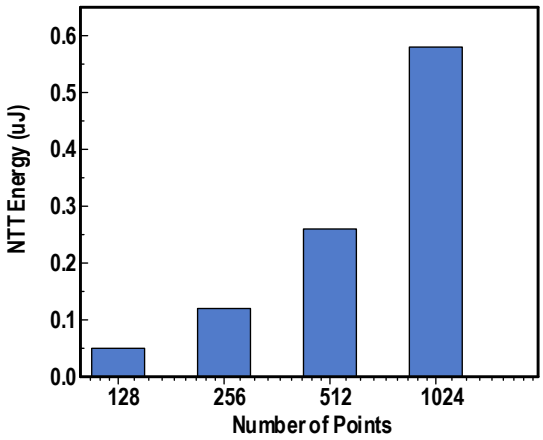


Figure 4.11 : NTT energy versus different polynomial order at bitwidth of 14.

and polynomial order. Column-wise power-gating make it possible to accommodate NTT of smaller polynomial order with energy saving. The read/write address and peripheral instructions are sent from the control block. Thanks to the routing schedule, the 6-T SRAM for PIM is able to process arbitrary polynomial order within the size constraint. Good scalability is maintained as SRAM can be extended to more columns in MeNTT.

4.6.1 Comparisons with software solutions

TABLE. 4.1 compares MeNTT with software, FPGA, ASIC and previous PIM designs for NTT computation. As discussed in previous sections, traditional software implementation has obvious bottleneck for Ring-LWE computations with the increase in polynomial order and bit-width. The energy cost and latency are higher than hardware approaches by several orders. Under current Von-Neumann architecture, the highly parallel butterfly-operation-NTT can only be executed in a normal serial approach. Butterfly operations are executed one by one with I/O operations on memory. While the modular arithmetic its self usually take only one or several

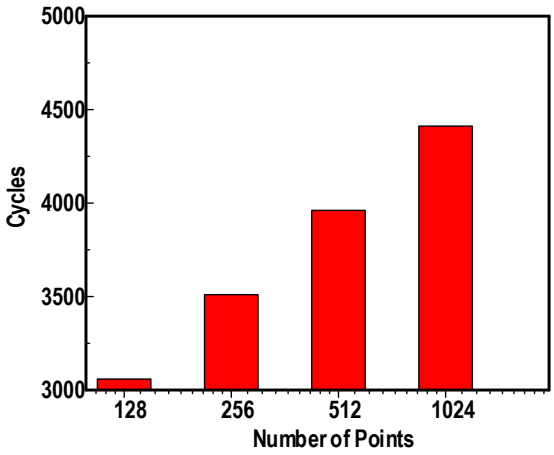


Figure 4.12 : NTT cycle count versus different polynomial order at bitwidth of 14.

cycles each, the data transfer between memory and processor may involve cache and main memory read/write which can cost 10s of cycles, making the whole process extremely expensive. From TABLE. 4.1, one can observe the significant energy and latency overhead in X86 based software approach. Despite the fact that a CPU can run at higher speed than custom hardware, it is not a perfect solution for low-power and high-performance security applications. Another disadvantage is the robustness against side-channel analysis. Since all operations are executed in ALU in serial, it is relatively easy for an attacker to retrieve power and timing information leak for side-channel attack. Due to the low efficiency of memory access in software approach, hardware acceleration is widely desired and adopted for FFT and NTT applications.

4.6.2 Comparison with FPGA solutions

On the other hand, FPGA-based hardware approaches exhibit improved performance thanks to the customized architecture and datapath to elevate parallelism and efficiency for NTT computation. To support the modular arithmetic in NTT flow, DSP

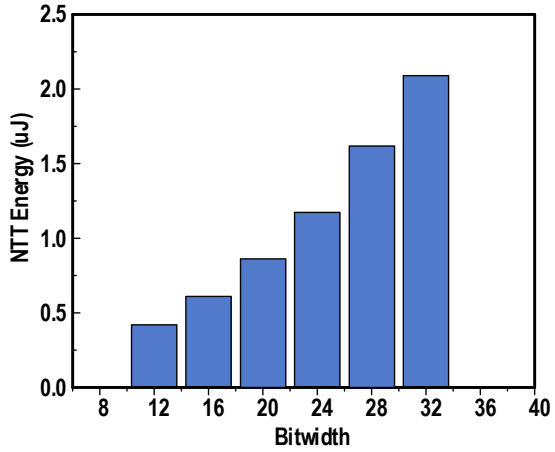


Figure 4.13 : NTT energy versus different bitwidth at polynomial order of 1024.

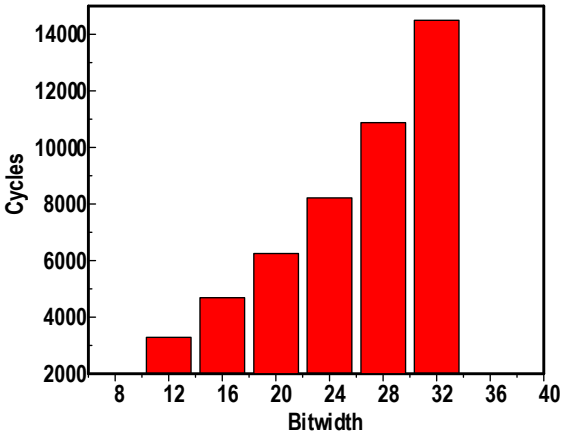


Figure 4.14 : NTT cycle count versus different bitwidth at polynomial order of 1024.

blocks [59] or custom multipliers are required to provide sufficient computing capability. Optimized data path are designed to accommodate the round by round NTT and INTT in a pipelined [73] approach which achieved higher throughput with considerable area overhead. While FPGA might be a decent choice for cloud platforms, it is instrinsically bulky, power hungry and inflexible. In order to support different schemes, FPGAs usually need to be programmed and re-compiled , which takes time of up to days. From 4.1 we can see that although FPGA solutions beat traditional

software approach in terms of throughput and latency, they come with excessive energy and area cost compared with ASIC and PIM solutions. Therefore FPGA is more suitable for prototype and cloud platform when dealing with post-quantum cryptography.

4.6.3 Comparison with ASIC solutions

The disadvantages of software and FPGA approaches inspired exploration in custom hardware solutions. ASIC implementations make use of standard SRAM or registers and compute modular arithmetic in digital circuits [60]. Although ASICs usually beats FPGAs in speed, energy and area, they come with much higher cost. Another major drawback for this approach is the amortized BL energy spent on reading data from SRAM. The processing speed is also limited by traditional memory bandwidth. While embedding registers for local storage in each computing unit increases throughput by increasing parallelism as in [62], the area overhead and limited scaling potential are main drawbacks.

A major consideration for ASIC application in NTT accelerator is the quantitative relationship of polynomial order and bitwidth. MeNTT executes modular arithmetic in bit-serial and word-parallel order, thus performs better when polynomial order gets higher. While traditional software, FPGA and ASIC approach carry out butter-fly unit operation word by word, and do not scale well when polynomial order increases, which is very likely to happen as more protocols of Ring-LWE are proposed.

ASIC solutions require higher design and fabrication cost, and need extra overhead to handle different schemes with different polynomial order, q and bitwidth. MeNTT provides higher degree of reconfigurability by providing general modular arithmetic in a highly parallel computing approach. The computation and data movement can be

reprogrammed with minimal effort by changing WL accessing sequence and peripheral configurations. The SRAM rows and columns can be gated in different cryptography schemes for power and time saving. MeNTT also provides higher normalized energy efficiency and area efficiency, as show in Table. 4.1, mainly benefiting from in- and near-memory computation.

4.6.4 Comparisons with PIM solutions

Compared to previous PIM studies that focus on leveraging parallel word-serial or bit-serial operation for general-purpose arithmetic in SRAM [71] and ReRAM [64, 72], the optimized modular arithmetic and dataflow help MeNTT outperforms prior PIM works in energy and area efficiency as well as latency. While [64] achieved higher throughput by introducing multiple pipelines to break the data path into shorter pieces, our approach target at ultra-compact single SRAM bank implementation, which is more applicable to resource-constrained applications. Note that the reported latency for [64] is for specially selected q with very small hamming weight. The latency varies a lot with different choices of schemes and q. More latency will occur for many other generic choices of q. Last but not least, our proposed techniques can be applied to generic bit-serial PIM architectures for performance and efficiency enhancement, regardless of the memory technologies.

4.7 Summary

In summary, this chapter presents, MeNTT, a novel PIM architecture for NTT acceleration. With the proposed bit-serial modular arithmatic protocol and mapping strategy, it achieves superior efficiency and throughput with a compact footprint. A fully functional mixed-signal implementation of the system verifies its feasibility in

physical design, and provides realistic estimation of its performance for comparison.

Algorithm 5 Modular Multiplication

Require: $x, y \in Z_q$, where y_k represents k-th bit of N-bit y
Ensure: z = (x.y) mod q

1: $psum = 0, overflow\_4q = 0, overflow\_2q = 0$
2: **for** $(k = N - 1;\ k \geq 0;\ k = k - 1)$ **do**
3: **for** $(j = 0; j < N; j = j + 1\ )$ **do**
4: **if** $(overflow\_4q = 1)$ **then**
5: $psum_j \leftarrow (psum_j \ll 1) + (x_j * y_k) - q_{j-2} + carry - borrow$
6: **else if** $(overflow\_2q = 1)$ **then**
7: $psum_j \leftarrow (psum_j \ll 1) + (x_j * y_k) - q_{j-1} + carry - borrow$
8: **else**
9: $psum_j \leftarrow (psum_j \ll 1) + (x_j * y_k) + carry - borrow$
10: **end if**
11: $write_j = LSB(psum_j)$
12: $carry = (psum_j > 1)?1 : 0$
13: $borrow = (psum_j < 0)?1 : 0$
14: $cmp_{2q} = (psum_j == q_{j-i})?cmp_{2q} : psum_j$
15: $cmp_{4q} = (psum_j == q_{j-2})?cmp_{2q} : psum_j$
16: **end for**
17: **if** $(cmp_{4q} == 1)$ **then**
18: $overflow\_4q \leftarrow 1$
19: **else**
20: $overflow\_4q \leftarrow 0$
21: **end if**
22: **if** $(cmp_{2q} == 1)$ **then**
23: $overflow\_2q \leftarrow 1$
24: **else**
25: $overflow\_2q \leftarrow 0$
26: **end if**
27: **end for**
28: **if** $(psum \geq q)$ **then**
29: $psum \leftarrow psum - q$
30: **end if**
31: $z \leftarrow psum$ z

Table 4.1 : COMPARISON TABLE OF MENTT WITH PREVIOUS SOFTWARE, FPGA, ASIC AND PIM IMPLEMENTATIONS OF NTT ACCELERATOR.

	This Work				**ISSCC 2019 [5]**			**CICC 2018 [9]**		**DAC 2020 [11]**			**ICASSP 2020 [21]**			**Tech Report [11]**		
Method	**Bit-serial 6T SRAM**				Serial Butterfly			Parallel Butterfly		Bit-serial RRAM			FPGA			Software		
Technology	**65nm**				40nm			40nm		RRAM			Virtex-6			X86		
Frequency (MHz)	**151**				72			300		909			184			2000		
Bitwidth (ns)	**14**	**14**	**14**	**16**	13	13	14	13	14	16	16	16	16	16	16	16	16	16
Polynomial Order	**256**	**512**	**1024**	**1024**	256	512	1024	256	512	256	512	1024	256	512	1024	256	512	1024
Latency (ns)	**23k**	**26k**	**29k**	**30.7k**	17.9k	39.2k	85.5k	533	1.6k	68.7k	75.9k	83k	21.6k	47.6k	102k	85k	169k	349k
Throughput (NTT/sec)	**42k**	**38k**	**35k**	**33k**	56k	25k	12k	1.8M	609k	553k	553k	553k	46k	21k	10k	12k	5.9k	2.9k
Energy (nJ/NTT)	**121**	**263**	**580**	**614**	166	411	894	31	96	2580	5020	10040	2150	5280	12520	570k	1179k	2483k
Normalized Energy (nJ/NTT)[a]	**121**	**263**	**580**	**614**	780	1932	4202	146	451	2580	5020	10040	2150	5280	12520	570k	1179k	2483k
Area (mm^2)	**0.26**				0.28			1.4		-			-			-		
Normalized Area (mm^2)[a]	**0.26**				0.74			3.7		-			0.73/0.85/0.96			-		

a. Modeled using the technology by this work

Chapter 5

MePLER: A 20.6-pJ Side-Channel-Aware In-Memory CDT Sampler

Ideal learning with errors (LWE) requires adding Gaussian noise to the encrypted message to ensure the security of its scheme. As LWE and its variants such as Ring-LWE and Module-LWE are getting more attention in the applications of post-quantum cryptography and homomorphic encryption, their computing cost is an obstacle for the usage in resource-constraint devices. Previous chapter already addressed the issue of number theoretic transformation (NTT). Gaussian sampling is even-more power- and time-consuming than NTT, as can be observed by the example shown in figure 5.1. As shown in Fig. 5.2, the energy consumption of sampling in a widely-used PQC scheme NewHope occupies 77% of the total energy. This calls for solutions for provide efficient acceleration of sampling. Moreover, sampling is also widely used in statistic problems such as Bayesian Neural Network and Particle Filter. There have been increasing demands as more and more AI applications are required at edge devices. An efficient and low-cost sampler in hardware will benefit the development of AIoT.

In this section, we propose to address issues of hardware sampler using range-matching content-addressable memory. The sampler handles arbitrary distribution using cumulative distribution table (CDT) approach and provide constant-time sampling. We present MePLER, an in-Memory CDT sampler, featuring custom cell derived from NAND-Type CAM for range-matching, pipelined and segmented array

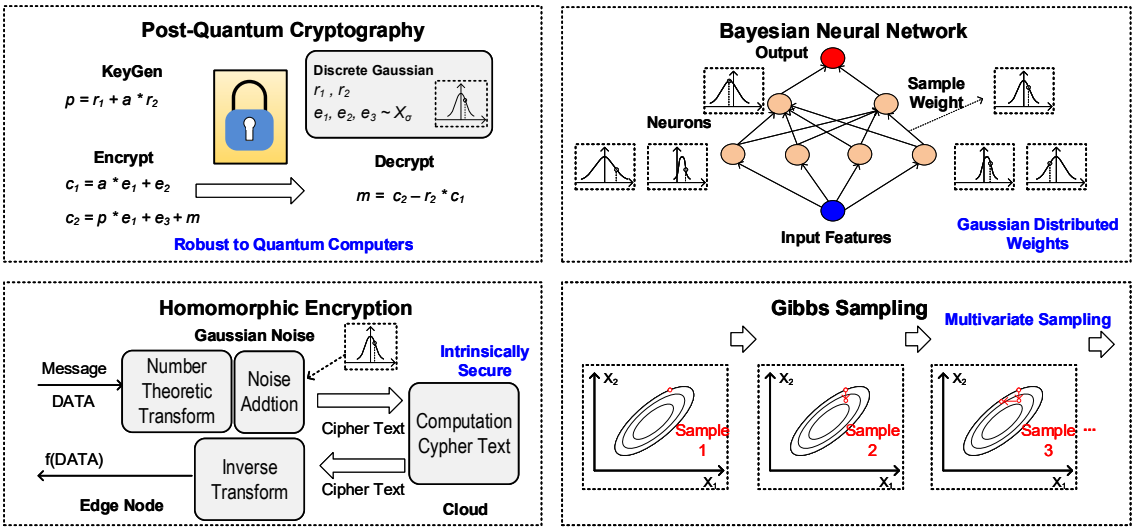


Figure 5.1 : Applications of random sampling in cryptography and machine learning.

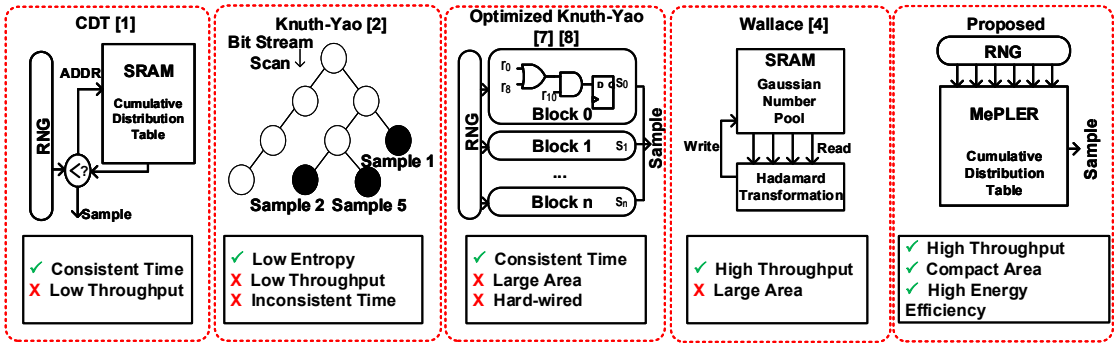
Operation	Energy % in Software
Sampling	77%
NTT	16%
Modular Arithmetic	7%

Figure 5.2 : Energy break down in the execution of NewHope.

for reduced energy, and suppressed timing and power side-channel leakage. The precision and sample range are configurable for different sampling requirements. A 65nm prototype achieves constant 85.9-MSps, 1-sample/cycle throughput, 20.6-pJ/sample efficiency, and 0.03-mm^2 footprint.

5.1 Motivation and Related Work

Sampling from a given distribution has been a common procedure in many scientific problems. There are many well-studied methods to implement the sampling. The most commonly used ones include: rejection sampling, binomial sampling, cumula-



Scheme	Percentage of Sampling in Total Cost
Post Quantum Cryptography (Software) [1]	78%
Post Quantum Cryptography (ASIC) [1]	28%
Homomorphic Encryption (Software) [3]	60.2%

Sampling Method	Latency	Constant Time?	Table Size	Table Lookup	Hardware Friendly
Rejection	~8	No	0	0	No
Bernoulli	~1.47	No	$\lambda\log_2(2.4\tau\sigma^2)$	$\log_2(\sigma)$	No
CDT-binary	$\log_2(\tau\sigma)$	Yes	$\sigma\tau\lambda$	$\log_2(\tau\sigma)$	Yes
CDT-linear	$\tau\sigma$	Yes	$\sigma\tau\lambda$	$\tau\sigma$	Yes
Knuth-Yao	$\sim\log_2(\sqrt{2\pi e}\sigma)$	No	$\sim 0.5\sigma\tau\lambda$	$\log_2(\sqrt{2\pi e}\sigma)$	Yes
Wallace	1	Yes	$n\lambda$	$k\lambda$	Yes
Proposed	1	Yes	$\sigma\tau\lambda$	$\sigma\tau\lambda$ (parallel)	Yes

σ: Variance τ: Tailcut Factor λ: Precision

Figure 5.3 : Comparisons of sampling methods.

tive distribution table (CDT) sampling [60] [74], Knuth-Yao (KY) sampling [62] and Bernoulli sampling. The performance and memory requirements of above sampling methods are compared in figure 5.3. In the figure, σ denotes the distribution's variance, τ denotes the upper limit or range of sampling, λ denotes the precision or bitwidth.

Rejection sampling is the most straightforward way to sample from a distribution. The sampler only needs to know the probability density function (PDF). Then it generates a random sample and a random number between 0 and 1. If the random number is smaller than the PDF value at the given sample, then the sample is accepted. Otherwise the sample is rejected and go back to the initial step. For example, in Figure 5.4, the sampler generated a random sample of 10. By looking up the PDF, the probability of 10 is 0.02, which corresponds to the probability of 10 being sampled.

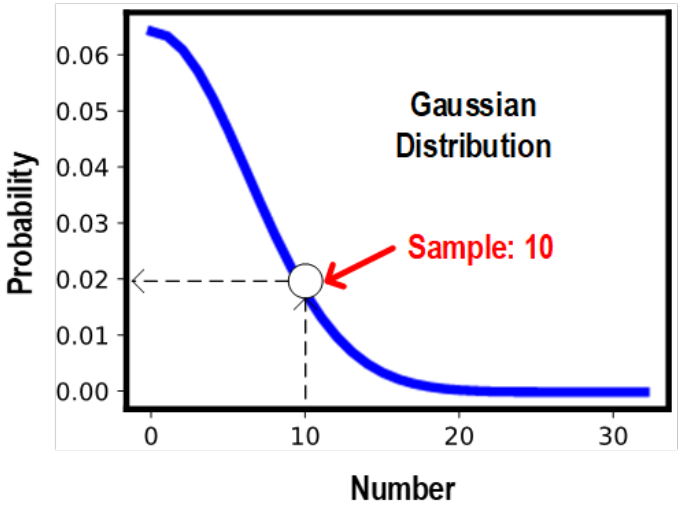


Figure 5.4 : Illustration of a rejection sampler.

Then the sampler generate another random number 0.01. since 0.01 is smaller than 0.02, 10 is an accepted sample. Rejection sampling does not require additional table and is the simplest approach. But as can be observed from the example, the sample has a very high probability of being rejected. The expected number of samples for an accepted sample if 8, which means the rejection sampling has low efficiency.

Bernoulli sampling [74] is an optimized sampling method over rejection sampling which improves the throughput of less than 2 trials per sample. However, the timing is still not constant for Bernoulli sampling, thus may leak information to side-channel attacks.

CDT sampling is another straightforward approach from mathematical point of view. Instead of using PDF for sampling, it make use of the cumulative distribution function (CDF). The process is basically using the inverse function of CDF. The sampling procedure can be explained by the example in Figure 5.5/ First, the sampler generate a random number between 0 and 1, which is 0.92 in the figure. Then it look it

up in the CDF function as y value to find the corresponding x value, which corresponds the the sample of 10. CDT sampler requires to store the CDF in memory as lookup table. But its throughput can be high and is hardware-friendly. Many hardware implementations have been proposed using CDT sampler. The main drawback is the tradeoff between speed and side-channel security. By using binary search, the CDT sampler can perform sampling in logarithm time. But the cycles taken is related to the final sample output, which will leak timing and power information in side-channel attacks. While some works used linear scan to mitigate this problem, the throughput and energy efficiency became worse.

KY sampling is another widely used sampling technique. It made use of the PDF and traverse a tree constructed by the distribution and perform sampling. It is also relatively easy to implement and optimize in hardware with high throughput. But its structure decides that it will take inconsistent cycles for sampling.

Besides the general sampling techniques, there are fast implementations for specific distributions. For example, binomial distribution can be sampled by generating two random bitstreams and taking the sum of hamming distance. Gaussian distribution can be approximated by binomial distribution in some cases where the errors in the distribution can be tolerated. Box-Muller transform is another technique to sample from a gaussian distribution. However, if is difficult to implement sinosuidal and square root functions in hardware.

CDT sampling and KY sampling using binary decision tree are widely adopted in hardware, because of their simple logic and table-based structures. Conventional KY samplers suffer from non-constant and low throughput, while recent optimizations realize (near-)constant and high throughput at the cost of large footprint and hard-wired logic, making them not suitable for ASIC. Traditional CDT samplers perform

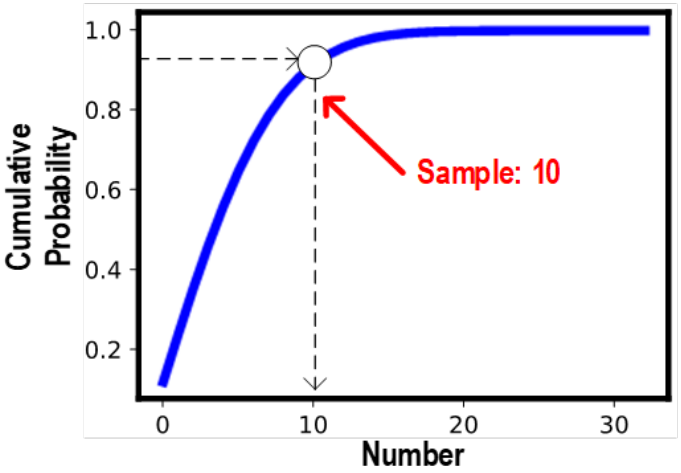


Figure 5.5 : Illustration of a CDT sampler.

linear or binary search to find the sample corresponding to the randomly generated probability. The search delay renders CDT samplers relatively slow and side-channel leaky. [75] forces all searches to go through the full table to avoid timing information, but at the penalty of reduced speed and efficiency. To this end, we present MePLER based on a pipelined range-matching CAM, with 20.6-pJ energy, constant 85.9-MSps throughput, 0.03-mm2 footprint, and suppressed timing/power side-channel leakage. MePLER can be easily programmed for arbitrary distribution with configurable precision and range.

5.2 Range-Matching-CAM

In this work, we present a sampler based on content-addressable memory (CAM) capable of matching ranges rather than exact words. Custom CAM cells and peripherals are designed to enable range-matching for CDT sampling. The sampling is executed in one sample per cycle, enabling high speed sampling with no timing information leakage. Differential match lines (ML) and random row masking ensures obfuscated

energy consumption during the search, further reduce the risk of side-channel attack on power trace and EM emission.

CAM is traditionally deployed in routers and switches for IP lookup and routing. There is also usage in TLB block for address lookup in Cache. The key feature for CAM is the capability of matching the input with all stored items in the bank, providing parallel search and improving throughput at the price of high instantaneous power. Conventional CAM will return 'match' or 'mismatch' when the input matches the item or mismatches the item, where the item can be exact the same with input, or have masked bits, and matches the input on other bits. In order to make use of the parallel search capability of CAM and apply it into the sampling problem, we proposed a CAM design to perform range-matching with input as depicted in Figure 5.6. The CAM is a modified NAND-type CAM as ML is connected serially by cells in a row, or word. In this design, the CDF table is stored in the CAM vertically in a sorted order. The probability for the largest sample is stored at the top and the smallest sample is stored at the bottom. The proposed CAM array has a dimension of 64 by 64. It supports a maximum sample range of 0 to 63, and the maximum bit precision of 64. For each input number corresponding the the probability to be matched, each row of the CAM array will be smaller than, greater than or equal to the number. The sampler will find the higher row smaller than or equal to the input and take the row number as the sample output. This design works exactly as a CDT sampler and is capable of handling sampling of arbitrary distributions within the range and precision. The throughput is exactly 1 sampler per cycle.

The detailed demonstration of the range-matching process is depicted in Figure 5.7. The example used 3-bit number for range-matching. The two rows in CAM store '110' and '100' respectively from top to bottom. MSB is stored from the left and LSB

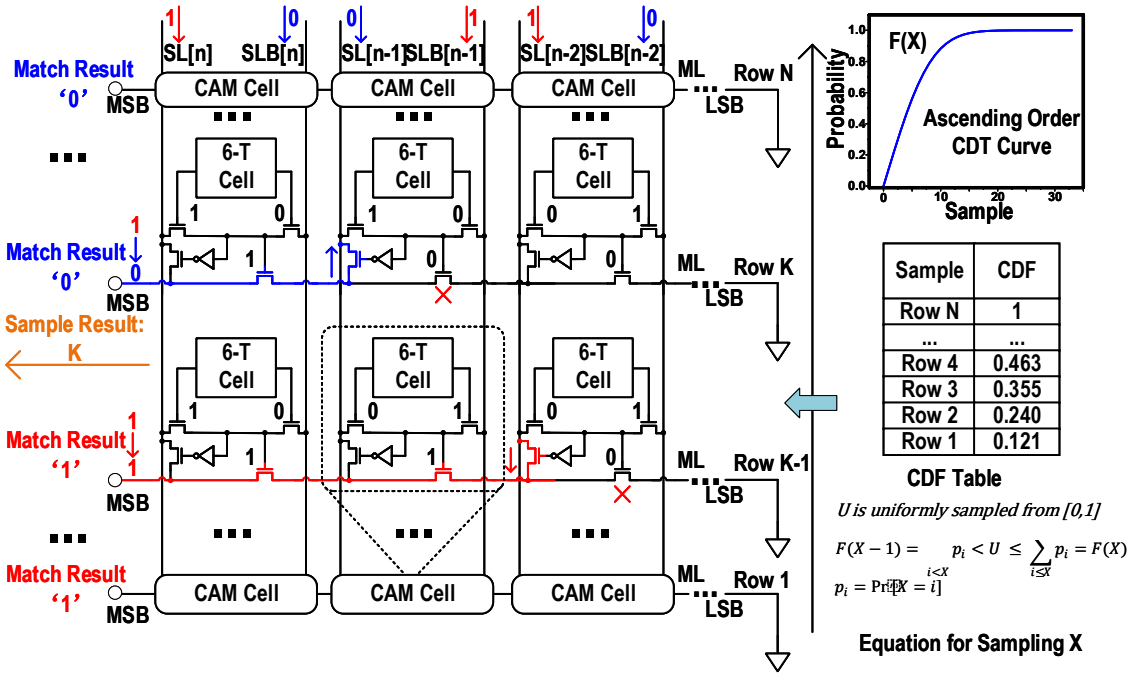


Figure 5.6 : Block diagram of proposed range-matching-cam array and schematic of CAM cell.

at the right. The MLs are precharged to high before each search. The input number '101' is then applied on the search lines (SL), which results in the high voltage on SL[2], SLB[1] and SL[0], and low voltage on SLB[2], SL[1] and SLB[0]. As is the case in traditional NAND-type CAM, the ML will be discharged to ground if the input matches the stored value, because the pass transistor of each CAM cell connecting the ML will be turned on when each input bit matches the stored bit. The voltage controlling the pass transistor is the XOR of input and stored value.

CDT sampling is an instantiation of inversion sampling that requires a pre-computed cumulative density function (CDF) table (Fig. 5.6). Then it finds the interval in the table that a uniform random sample from [0,1] falls into. The index of the interval will be a random sample following the given CDF. Our key idea is performing

this range matching with specially designed NAND-type CAMs, requiring one extra NMOS controlled by an inverter over the standard 9-T cell. The CDF table is stored in the CAM in an ascending order. If the input data matches a row in the table, the match line (ML) is shorted from MSB to LSB with all pass-gates turned on. When there is a mismatch, the value of ML will be decided by the highest mismatched bit. Within this bit, the pass-gate will disconnect the ML and drive ML to the input search line (SL). As a result, the CAM serially performs matching from MSB to LSB. The range matching result will appear at MSB end of ML, which will be "0" if input is smaller or equal to the stored value, and "1" if input is larger than the stored one.

In the proposed range-matching CAM, the pass transistor will be turned off when input mismatch with stored value. And with the additional transistor connecting to SL in each cell, each mismatched cell will drive its own piece of ML to the corresponding input value. For input '1' and stored '0', the ML will be high. For input '0' and stored '1', the ML will be low. The ML at MSB will be driven to the first bitcell of mismatch because all cells before the mismatch is matched and connect the ML until the mismatch cell. Therefore the voltage on MSB ML will be '1' when the first mismatch cell has input '1' and stored '0'. ML will be '0' vice versa. This is exactly the same approach people compare a number with another one, by comparing each bit from MSB to LSB and find the first mismatch.

The main target of this design is Discrete Gaussian distribution for lattice-based cryptography (LBC). In most popular LBC schemes, the sampling range is under 64 which can be supported by the proposed sampler in just one cycle. In some cases where the variance of Gaussian distribution is large, such as BLISS, the sampling can be completed in two steps called Gaussian Convolution. Two samples from a Gaussian distribution of smaller variance can combine into a sampler from a Gaussian

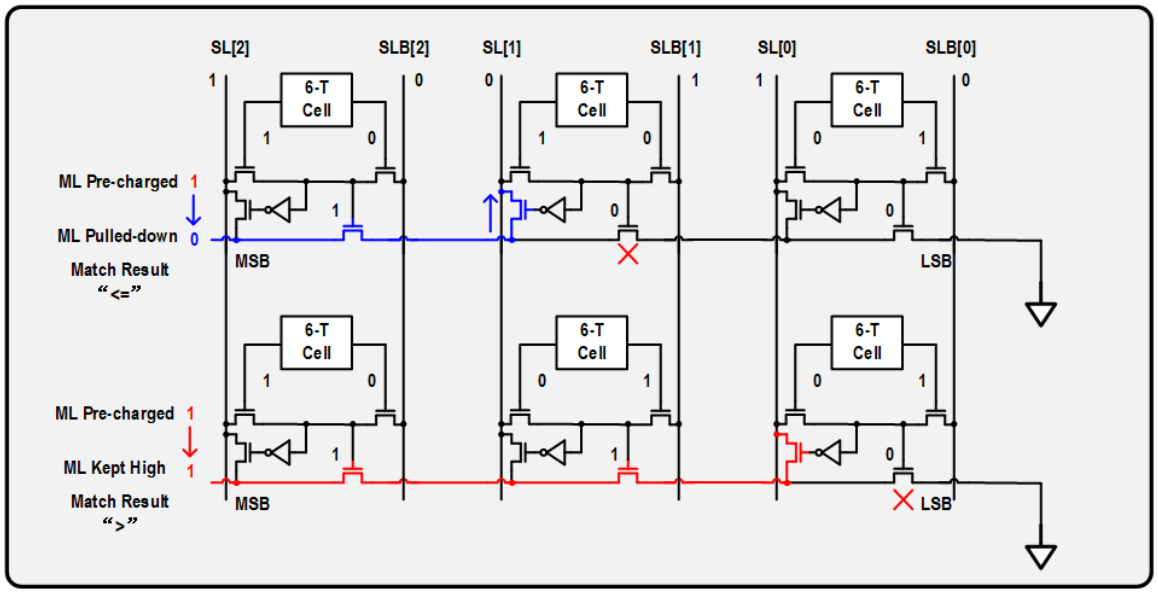


Figure 5.7 : Range-matching process in the proposed MePLER.

distribution of larger variance by scaling and addition. In the case of BLISS where the variance is 215.73, two samples from a distribution of variance of 19.53 can be used to combine into a sample in the original distribution [76]. Even more steps can be taken to split the sampling into smaller distributions. Therefore the proposed sampler can support the sampling requirements for most LBC schemes.

5.3 Segmented Search

The proposed range-matching-CAM is capable of searching for a number in a given distribution with a random input. However, it searches all 64 bits in 64 rows for every search. High power consumption will be induced but a large portion is redundant. This is because in most cases, the first mismatch occurs at locations close to MSB due to the random nature of input number. There is no need to search following up bits and spend energy on SLs and MLs. Therefore we propose using pipelined segments

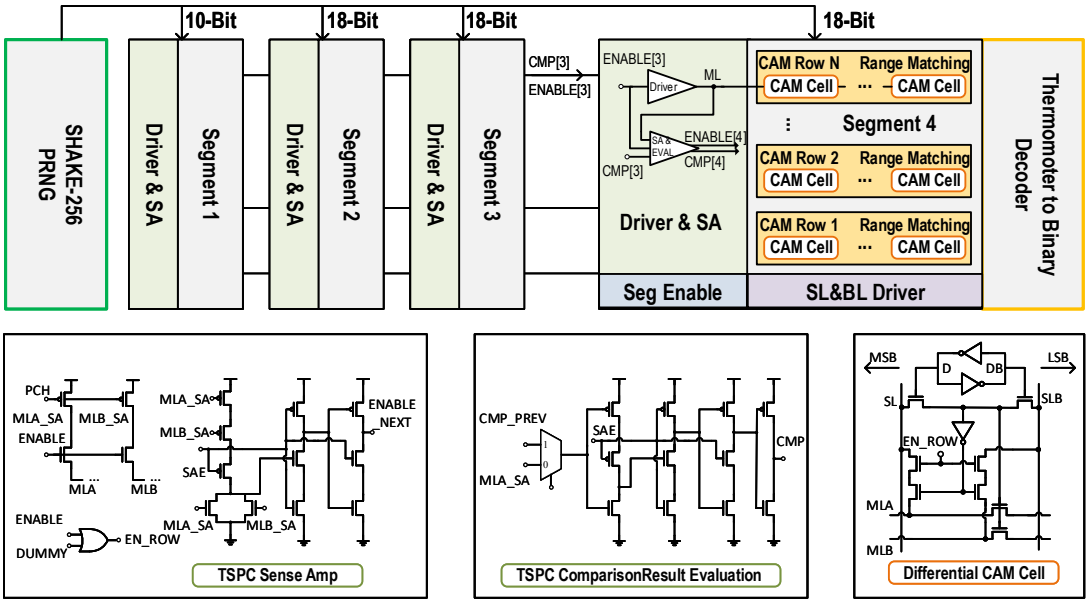


Figure 5.8 : Proposed segmented range-search MePLER.

to separate searching into stages. Each segment contains a number of bits, and if all searches terminates at a certain segment, the next segment will not be activated for searching. The differential cell layout is depicted in Fig. 5.9

In order to distinguish terminated searches, the MLs are implemented in differential approach. The item matches completely with input only when both differential MLs are low. Further search is required in next segment only when an item completely matches with the input. Therefore, the peripheral in the segments sense the matching of items and will activate next segment only when at least when complete match is detected.

The choices of bitwidth for each segment decides the average power consumption. Since the input of sampler is from a SHAKE256-based PRNG, it is completely random. The probability of matching and activating each segment can be calculated based on the distribution. Post-layout simulation results of energy and area are used

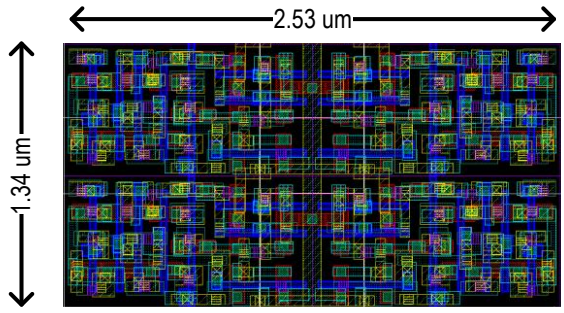


Figure 5.9 : 2x2 layout of the differential MePLER cell.

to construct the model for segment optimization. Figure 5.10 shows an example of optimization when the array is divided into 3 segments. An optimal point can be found to make average sampling energy optimal when first stage has 10 bits and second stage has 10 to 12. After careful modelling and simulation, a configuration of 4 segments with 10, 18, 18 and 18 bits were chosen to optimize the sampling energy. The 4-segments design were implemented together with a single-segment baseline implementation to compare the energy, latency and robustness against side-channel attacks.

Due to the random nature of input, the range-matching will terminate in first few bis in most cases. Matching remaining LSBs waste energy on both ML and SL. The serial pass-gates also induce large delay. Thus, we propose dividing the array into pipelined segments. Each segment contains cells, column peripherals and row peripherals (Fig. 5.8). Segmented array avoids redundant search when search terminates early and increases throughput through pipelining. The segmented design also requires differential MLs to generate the enabling signal for next segment, because it is only needed when the previous segment has exact match. This state can only be represented by having both MLs at "0". Moreover, differential MLs reduce the energy difference when input varies, thus suppressing power side-channel leakage.

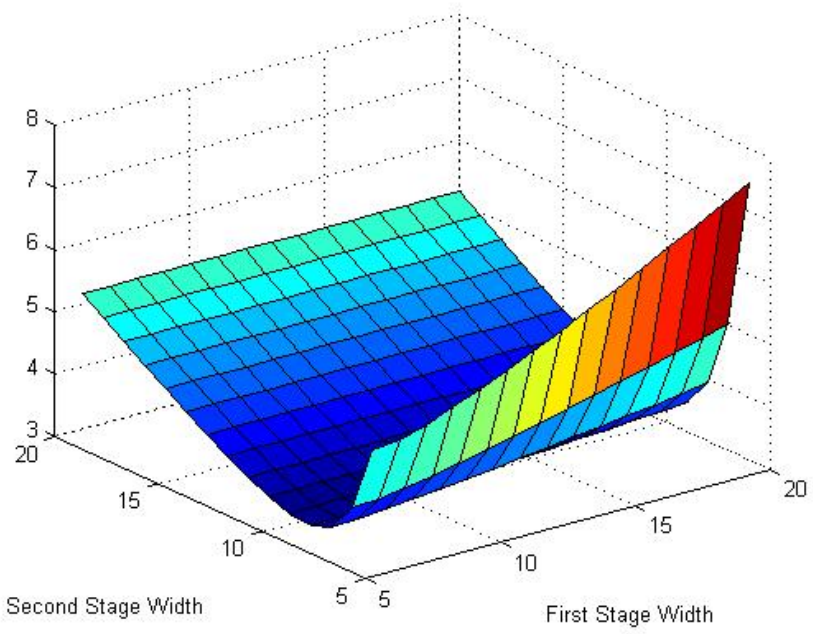


Figure 5.10 : Optimization of power consumption with respect to bitwidth selection in segments.

Only enabled rows will perform comparisons in a segment, while other rows will pass the result from last segment through multiplexers and TSPC registers.

5.4 Configurable Search Range and Precision

The LBC schemes have various parameters in sampling from sampling range, precision to distribution type. In order to optimize the design for different schemes, configurable search range and precision were implemented to grant the design flexibility and efficiency.

The search-range configuration is controlled by the row-wise power-gating as depicted in Figure 5.11. Each row have a enable signal to control the ML and peripheral during search. When the sampler is sampling from a small range, redundant rows

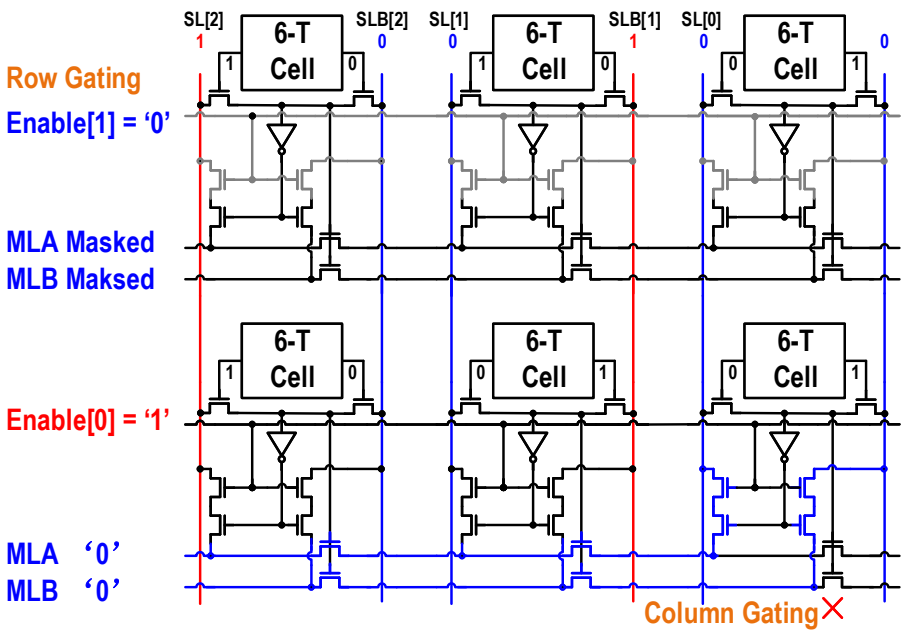


Figure 5.11 : Row-wise and column-wise power-gating for configurable search range.

will be disabled so that the ML and peripheral will be de-actived and the result is always 'smaller than'. Thus the energy spent on redundant rows can be saved.

The bit precision configuration is controlled by the column-wise power-gating. An example is shown in Figure 5.11. MSBs on the left are enabled in the search according to the bit precision, so that the input data will charge the SLs and perform searching. For columns that are beyond the required bit precision, the SL's are power-gated and will not be charged in the searching operation. The column next to the last bit of active searching columns will store all 1's so that the MLs at LSB of active searching area will be connected to the ground at this column instead of being float. The column-wise gating saves both SL and ML energy for redundant bits and improves the flexibility and efficiency of the range-matching CAM.

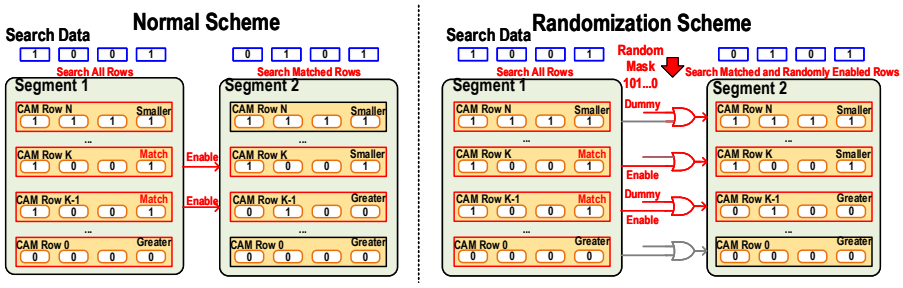


Figure 5.12 : Random masking of rows for defence against side-channel attack.

5.5 Row-Masking for Robustness against Side-Channel Attacks

In post-quantum cryptography, noises drawn from Gaussian or other distribution is crucial to obfuscate the message and guarantee the security level. The noise have to be secret in order to make the algorithm robust. Therefore the security of the sampler is important for the robustness of the hardware implementation of the accelerator of LBC.

Previous implementations of hardware sampler suffers from timing attack and power analysis attack. Our work has a fixed sampling cycle and is robust against timing attacks. However, due to the segmented structure, power domain information needs to be masked in order to defend against side-channel attacks in power analysis.

The difference in power comes from the different rows and segments being activated for search. In order to mitigate this, a mask signal is designed to enable rows when they are not being searched by the real search operation. The mask signal reuse the PRNG output for the search operation of last cycle. The mask signal is ORed with the enable signal to activate rows. The difference is that masked rows will not have enable output for next segment. The number of mask signals available for the

'dummy search' can also be controlled to adapt to different distributions. We applied k nearest-neighbors (KNN) approach in simulation trying to predict the output from power traces. With the masking approach, the prediction accuracy reduced from 30% to 10% in a specific distribution which is close to random guess. Timing and power signals are major sources of SCA. This design naturally has a constant 1 sample/cycle speed, thus is robust to timing attacks. In order to increase the resistance in power SCA, a random masking scheme is designed (Fig.7). Beside the rows to be automatically enabled for searching, a temporal random mask with a programmable maximum is generated by the PRNG to activate unused rows and obfuscate the power signatures. We utilized a differential power analysis (DPA) scheme in Fig. 5.13.

Our prototype includes a 64×64 array, supporting a sampling range of -63 to 63 and a precision of 64-bit. This precision is sufficient for 128-bit post-quantum security. For wider Gaussian distributions with larger sigma, multiple steps of Gaussian convolution can be used to effectively enlarge the sampling range [74]. Four-stage pipelines with 10, 18, 18 and 18 bits per segment is chosen, through optimization of energy, area and delay for Gaussian CDT and random matching patterns. Uniform random inputs for sampling are generated by a SHAKE-256 PRNG. The thermometer coded CAM results are lastly converted to binary output.

5.6 Measurements

The chip micrograph is as depicted in Figure 5.14. Fabricated in 65nm LP process, the differential 16-T CAM cell takes $802F^2$ in logic rule (Fig. 5.9). MePLER generates random samples with ideal distributions (Gaussian and Gamma distributions in Fig. 5.15 and 5.16). The shmoo plot is shown in Fig. 5.17. The energy ranges from 5 to 35 pJ across 1.7 to 1.4 V. Column-wise power-gating saves up to 30% energy at low

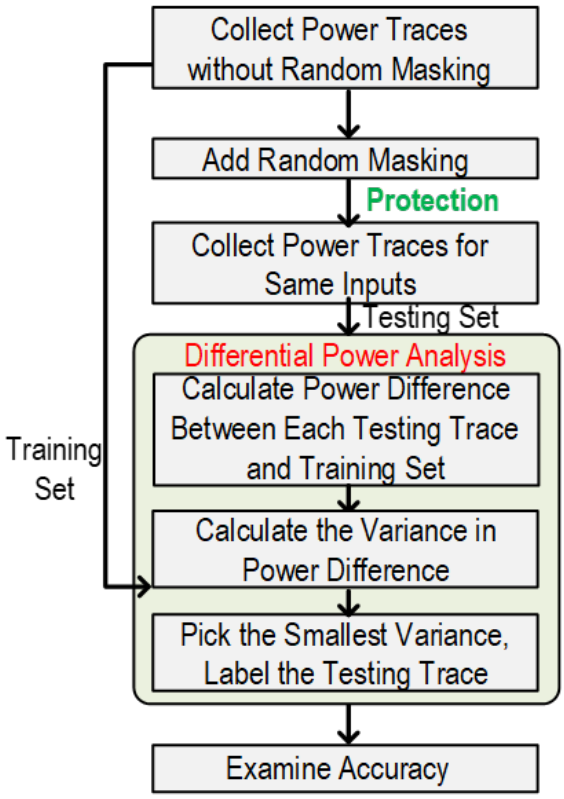


Figure 5.13 : Flow of differential power analysis.

precisions. Random masking significantly reduces DPA success rate with less than 10pJ/sample energy overhead (Fig. 5.18).

5.7 Summary

In this chapter, we propose a range-matching-CAM-based CDT sampler. The sampler features constant 1 sample/cycle with 20.6pJ/sample energy efficiency. Optimized segmented search ensures low power consumption. Configurable searching range and precision help it adapt to different distributions with reduced energy overhead. Random-masking strengthens its defense against side-channel attack.

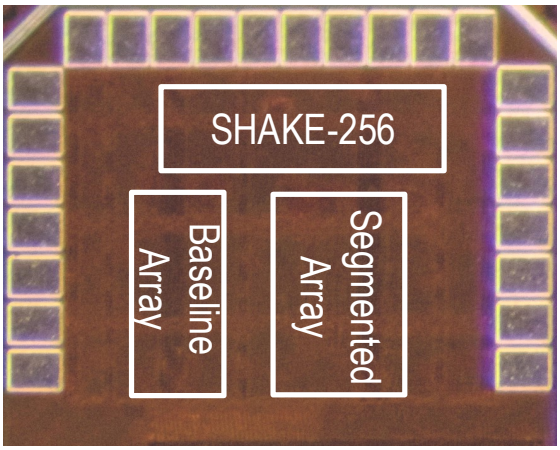


Figure 5.14 : Micrograph of the designed MePLER chip.

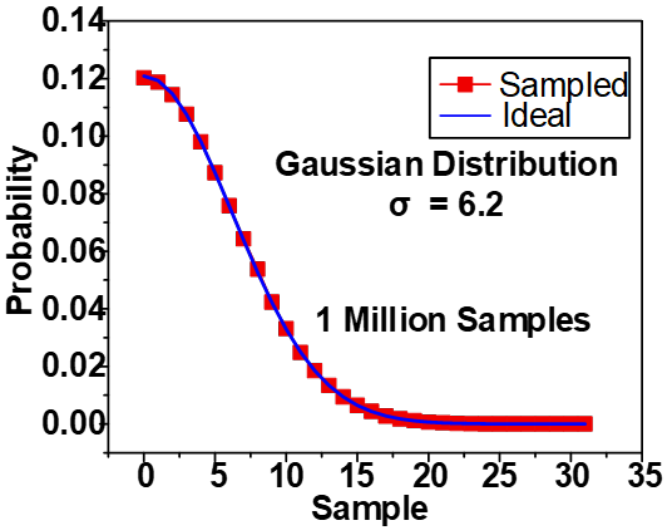


Figure 5.15 : Sampler distribution from a Gaussian distribution versus ideal distribution.

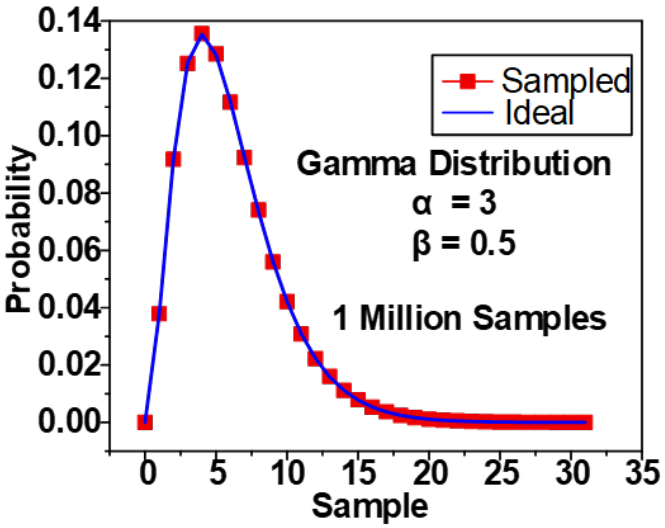


Figure 5.16 : Sampler distribution from a Gamma distribution versus ideal distribution.

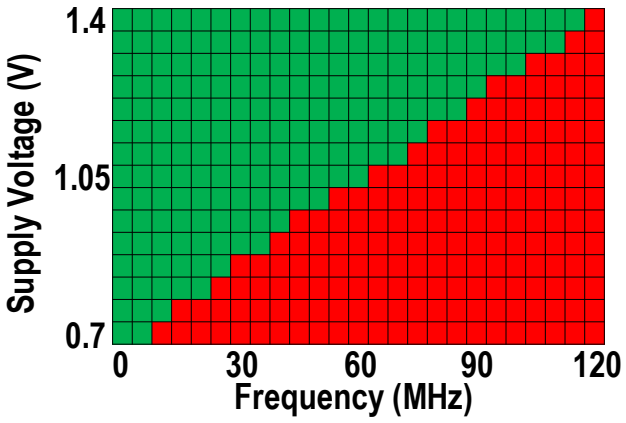


Figure 5.17 : System shmoo plot of MePLER.

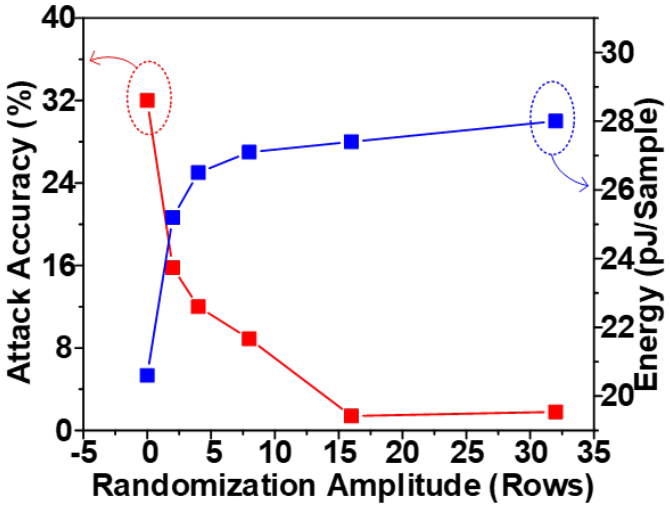


Figure 5.18 : DPA accuracy and energy versus number of available rows for masking.

	This Work	This Work	ISSCC 19 [1]	CICC 18 [2]	ToC 20 [8]	ToC 18 [7]	ToC 18 [6]	ToC 18 [6]	ToC 19 [5]	DAC19 [9]
Method	**CDT CAM Differential**	CDT CAM Single-end	Linear CDT	Knuth-Yao	Knuth-Yao	Knuth-Yao	Ziggurat	Binary CDT	Bernoulli	Knuth-Yao
Technology	**65nm**	65nm	40nm	40nm	FPGA Virtex-6	FPGA Virtex-6	FPGA Virtex-6	FPGA Virtex-6	Software AVX2	Software i7-6600U
Frequency (MHz)	**85.9**	21	72	300	353	204	60.3	297	-	-
Throughput (MSps)	**85.9**	21	1.125	20	353	204	6.7	59.4	-	-
Cycles/Sample	**1**	1	64	14.9	1	1	9	5	85	2293
Energy (pJ/Sample)	**20.6 (Sample) 13.2 (PRNG)**	60.9 (Sample) 13.2 (PRNG)	2406	5078	-	-	-	-	-	-
Precision	**64**	64	32	256	64	112	64	64	128	128
Range	**64**	64	64		82	82	32	32	1935	26
Memory	**0.5KB**	**0.5KB**	0.25KB	65KB	335 Slice 1169 LUT 599 FF	2682 LUT 997 FF	785 Slice 563 LUT 785 FF	43 Slice 112 LUT 19 FF	2KB	-
Area (mm^2)	**0.034**	0.026	~0.1	0.4	0.23[a]	1.88[a]	0.55[a]	0.03[a]	-	-
SCA Aware	**Power & Timing**	Only Timing	Only Timing	No	Only Timing	Only Timing	No	No	No	No

a. Estimated for 65nm process

Table 5.1 : COMPARISON OF MEPLER WITH PREVIOUS HARDWARE SAMPLER WORKS.

Chapter 6

Conclusion

In the thesis, I presented my work towards hardware security primitives for resource-constrained devices. I realized hybrid approach of algorithm, architecture and circuit innovations to design and implement blocks and systems to perform the functionality of key generation/storage, network intrusion detection, accelerator for NTT/INTT and CDT sampling for various applications in root of trust and chain of trust.

My contribution can be summarized and categorized into the following aspects and projects:

1. I designed a self-regulated PUF for key generation, achieving 0.3% native BER and 0.076 fJ/bit energy efficiency [12] [13]. Native regulation provides resistance to supply voltage variation and lead to 0.057 %/0.1 V BER sensitivity against voltage variations. The proposed in-cell reconfiguration scheme reduced native BER by two orders of magnitudes to 0.00182% with no area overhead. Moreover, a fast searching method of emulating temperature variation by sweeping body bias is applied to locate unstable bits with low testing cost. The bit cell of this PUF occupies only 562 F^2 . The trade-off of stability, energy, throughput and area achieved state-of-the art among CMOS PUF designs. All applicable NIST 800-22 and 800-90B randomness tests were passed for this PUF using 10 chips' measurements. A revised version [14] achieved 0 BER by utilizing a self-healing approach through supply voltage scanning.

2. I implemented a software-hardware co-optimized NIDS hardware using innovative CAM design. This is the first ever real chip to implement hardware firewall

for IoT devices [55]. Leveraging holistically designed architecture and circuits, a 65-nm prototype achieves best-in-class 1.54-fJ/search/pattern byte energy efficiency and 0.9-byte/pattern byte memory efficiency. [15]

3. I proposed a computation-in-6-T-SRAM NTT/INTT accelerator for post-quantum cryptography, which deployed optimized bit-serial operation for generic modular arithmetic, and can be applied to many other cryptography applications.

4. I designed MePLER, a CDT sampler featuring constant 1 sample/cycle with 20.6pJ/sample energy efficiency, which is the best in the world as of now. The prototype was fabricated in 65nm technology. Random-masking strengthens its defense against side-channel attack. [16] It solved the throughput and side-channel robustness issue in existing hardware samplers.

5. For all the presented projects, the focused application is providing secure hardware components in resource-constraint devices. I concentrated on memory-centric architecture to explore novel yet effective approach beyond Von-Neumann architecture for security. Combining hardware/architecture/circuit innovation with algorithm/software innovation, optimal performance and energy/area-efficiency are achieved, which fulfilled the goal to secure resource-constraint devices from prevalent attacks.

For the future, possible improvement can be made in several perspective including but not limited to:

1. Integration of proposed highly-stable PUF, memory-centric hardware firewall and post-quantum cryptography accelerator together with standard main processor and memory, working towards a complete solution composed of root of trust and chain of trust. To support similar or mode advanced functionality with widely used TPM, TEE or other security scheme in resource-constraint devices, works proposed

in this thesis provides realistic solutions with superior performance and low cost.

2. Augmenting current approach of self-regulated and self-healing from weak PUF to the application of strong PUF, which would be a great enhancement in the number of available challenge-response pairs.

3. Reusing the 8-T CAM-based NIDS engine in areas such as data mining and gene sequencing, where automata plays an important role and needs custom acceleration.

4. Expanding the usage of MePLER from lattice-based cryptography to more general security and machine learning applications such as Bayesian Neural Network and Monte-Carlo Markov Chain, which can be of important usage in computer vision and autonomous driving,

Bibliography

[1] E. Bursztein, "Inside Mirai the infamous IoT Botnet: A Retrospective Analysis." https://elie.net/blog/security/inside-mirai-the-infamous-iot-botnet-a-retrospective-analysis/, 2017.

[2] H. Joel, "AMD's Secure Processor Firmware Is Now Explorable Thanks to New Tool." https://www.extremetech.com/computing/292722-amds-secure-processor-firmware-is-now-explorable-thanks-to-new-tool, 2019.

[3] N. Chen, "The Benefits Of Antifuse OTP." https://semiengineering.com/the-benefits-of-antifuse-otp/, 2016.

[4] P. Marks, "Unintended Consequences." https://cacm.acm.org/news/238385-unintended-consequences/fulltext?mobile=false, 2016.

[5] B. Davenport, "The challenges of automotive functional safety verification." https://www.techdesignforums.com/practice/technique/the-challenges-of-automotive-functional-safety-verification/, 2016.

[6] D. Das, M. Nath, B. Chatterjee, S. Ghosh, and S. Sen, "Stellar: A generic em side-channel attack protection through ground-up root-cause analysis," in *2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 11–20, IEEE, 2019.

[7] Picotech, "PicoTech Probe Stations." http://www.picotech.co.il/?page_id=366,

2020.

[8] H. N. Saha, A. Mandal, and A. Sinha, "Recent trends in the Internet of Things," in *2017 IEEE 7th annual computing and communication workshop and conference (CCWC)*, pp. 1–4, IEEE, 2017.

[9] K. L. Lueth *et al.*, "State of the IoT 2018: Number of IoT devices now at 7B–Market accelerating," *IoT Analytics*, vol. 8, 2018.

[10] K. Boeckl, K. Boeckl, M. Fagan, W. Fisher, N. Lefkovitz, K. N. Megas, E. Nadeau, D. G. O'Rourke, B. Piccarreta, and K. Scarfone, *Considerations for managing Internet of Things (IoT) cybersecurity and privacy risks*. US Department of Commerce, National Institute of Standards and Technology, 2019.

[11] L. Shen, "The NIST cybersecurity framework: Overview and potential impacts," *Scitech Lawyer*, vol. 10, no. 4, p. 16, 2014.

[12] D. Li and K. Yang, "25.1 A 562F^2 Physically Unclonable Function with a Zero-Overhead Stabilization Scheme," in *2019 IEEE International Solid-State Circuits Conference-(ISSCC)*, pp. 400–402, IEEE, 2019.

[13] D. Li and K. Yang, "A self-regulated and reconfigurable CMOS physically unclonable function featuring zero-overhead stabilization," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 1, pp. 98–107, 2019.

[14] Y. He, D. Li, Z. Yu, and K. Yang, "36.5 An Automatic Self-Checking and Healing Physically Unclonable Function (PUF) with¡ 3× 10-8 Bit Error Rate," in *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 64, pp. 506–508, IEEE, 2021.

[15] D. Li and K. Yang, "A Dual-Port 8-T CAM-Based Network Intrusion Detection Engine for IoT," *IEEE Solid-State Circuits Letters*, vol. 3, pp. 358–361, 2020.

[16] D. Li, Y. He, A. R. Pakala, and K. Yang, "MePLER: A 20.6-pJ Side-Channel-Aware In-Memory CDT Sampler," in *2021 Symposium on VLSI Circuits*, pp. 1–2, IEEE, 2021.

[17] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, 2014.

[18] J. W. Lee, D. Lim, B. Gassend, G. E. Suh, M. Van Dijk, and S. Devadas, "A technique to build a secret key in integrated circuits for identification and authentication applications," in *2004 Symposium on VLSI Circuits. Digest of Technical Papers (IEEE Cat. No. 04CH37525)*, pp. 176–179, IEEE, 2004.

[19] S. Devadas, E. Suh, S. Paral, R. Sowell, T. Ziola, and V. Khandelwal, "Design and implementation of PUF-based 'unclonable' RFID ICs for anti-counterfeiting and security applications," in *2008 IEEE international conference on RFID*, pp. 58–64, IEEE, 2008.

[20] M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Lightweight secure PUFs," in *2008 IEEE/ACM International Conference on Computer-Aided Design*, pp. 670–673, IEEE, 2008.

[21] K. Yang, Q. Dong, D. Blaauw, and D. Sylvester, "14.2 A physically unclonable function with BER $<10^{-8}$ for robust chip authentication using oscillator collapse in 40nm CMOS," in *2015 IEEE International Solid-State Circuits Conference-(ISSCC) Digest of Technical Papers*, pp. 1–3, IEEE, 2015.

[22] X. Xi, H. Zhuang, N. Sun, and M. Orshansky, "Strong subthreshold current array PUF with 2^{65} challenge-response pairs resilient to machine learning attacks in 130nm CMOS," in *2017 Symposium on VLSI Circuits*, pp. C268–C269, IEEE, 2017.

[23] S. Jeloka, K. Yang, M. Orshansky, D. Sylvester, and D. Blaauw, "A sequence dependent challenge-response PUF using 28nm SRAM 6T bit cell," in *2017 Symposium on VLSI Circuits*, pp. C270–C271, IEEE, 2017.

[24] Y. Cao, C. Q. Liu, and C. H. Chang, "A low power diode-clamped inverter-based strong physical unclonable function for robust and lightweight authentication," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 11, pp. 3864–3873, 2018.

[25] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, "Modeling attacks on physical unclonable functions," in *Proceedings of the 17th ACM conference on Computer and communications security*, pp. 237–249, 2010.

[26] "Reconfigurable ECC for adaptive protection of memory, author=Basak, Abhishek and Paul, Somnath and Park, Jangwon and Park, Jongsun and Bhunia, Swarup," in *2013 IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1085–1088, IEEE, 2013.

[27] S. Mathew *et al.*, "A 0.19 pJ/b PVT-variation-tolerant hybrid physically unclonable function circuit for 100in 22nm CMOS," in *Proc. 2014 IEEE Int. Solid-State Circuits Conf.(ISSCC)*, pp. 278–279.

[28] S. Okumura, S. Yoshimoto, H. Kawaguchi, and M. Yoshimoto, "A 128-bit chip identification generating scheme exploiting SRAM bitcells with failure rate of

4.45× 10^{-19}," in *2011 Proceedings of the ESSCIRC (ESSCIRC)*, pp. 527–530, IEEE, 2011.

[29] A. Alvarez, W. Zhao, and M. Alioto, "14.3 15fJ/b static physically unclonable functions for secure chip identification with <2% native bit instability and 140× Inter/Intra PUF hamming distance separation in 65nm," in *2015 IEEE International Solid-State Circuits Conference-(ISSCC) Digest of Technical Papers*, pp. 1–3, IEEE, 2015.

[30] J. Li and M. Seok, "Ultra-compact and robust physically unclonable function based on voltage-compensated proportional-to-absolute-temperature voltage generators," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 9, pp. 2192–2202, 2016.

[31] J. Lee, D. Lee, Y. Lee, and Y. Lee, "A 445F^2 leakage-based physically unclonable function with lossless stabilization through remapping for IoT security," in *2018 IEEE International Solid-State Circuits Conference-(ISSCC)*, pp. 132–134, IEEE, 2018.

[32] B. Karpinskyy, Y. Lee, Y. Choi, Y. Kim, M. Noh, and S. Lee, "8.7 Physically unclonable function for secure key generation with a key error rate of 2E-38 in 45nm smart-card chips," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 158–160, IEEE, 2016.

[33] S. Taneja, A. B. Alvarez, and M. Alioto, "Fully synthesizable PUF featuring hysteresis and temperature compensation for 3.2% native BER and 1.02 fJ/b in 40 nm," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 10, pp. 2828–2839, 2018.

[34] K. Yang, Q. Dong, D. Blaauw, and D. Sylvester, "8.3 A 553F^2 2-transistor amplifier-based physically unclonable function (PUF) with 1.67% native instability," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 146–147, IEEE, 2017.

[35] S. Satpathy, S. K. Mathew, V. Suresh, M. A. Anders, H. Kaul, A. Agarwal, S. K. Hsu, G. Chen, R. K. Krishnamurthy, and V. K. De, "A 4-fJ/b delay-hardened physically unclonable function circuit with selective bit destabilization in 14-nm trigate CMOS," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 4, pp. 940–949, 2017.

[36] M. Seok, G. Kim, D. Blaauw, and D. Sylvester, "A portable 2-transistor picowatt temperature-compensated voltage reference operating at 0.5 V," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 10, pp. 2534–2545, 2012.

[37] M. J. Pelgrom, A. C. Duinmaijer, and A. P. Welbers, "Matching properties of MOS transistors," *IEEE Journal of solid-state circuits*, vol. 24, no. 5, pp. 1433–1439, 1989.

[38] S. NIST, "800-22: Documentation and Software-Random Bit Generation— CSRC," *URL: https://csrc. nist. gov/Projects/Random-Bit-Generation/Documentation-and-Software (accessed: 12.02. 2020)(in Russian)*.

[39] E. Barker and J. Kelsey, "NIST DRAFT Special Publication 800-90b recommendation for the entropy sources used for random bit generation," 2012.

[40] S. Stanzione, D. Puntin, and G. Iannaccone, "CMOS silicon physical unclonable functions based on intrinsic process variability," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 6, pp. 1456–1463, 2011.

[41] M.-Y. Wu, T.-H. Yang, L.-C. Chen, C.-C. Lin, H.-C. Hu, F.-Y. Su, C.-M. Wang, J. P.-H. Huang, H.-M. Chen, C. C.-H. Lu, *et al.*, "A PUF scheme using competing oxide rupture with bit error rate approaching zero," in *2018 IEEE International Solid-State Circuits Conference-(ISSCC)*, pp. 130–132, IEEE, 2018.

[42] Y. Pang, B. Gao, D. Wu, S. Yi, Q. Liu, W.-H. Chen, T.-W. Chang, W.-E. Lin, X. Sun, S. Yu, *et al.*, "25.2 A reconfigurable RRAM physically unclonable function utilizing post-process randomness source with $<6\times 10^{-6}$ native bit error rate," in *2019 IEEE International Solid-State Circuits Conference-(ISSCC)*, pp. 402–404, IEEE, 2019.

[43] B. Caswell and J. Beale, *Snort 2.1 intrusion detection*. Elsevier, 2004.

[44] T. Kojm, M. Cathey, C. Cordes, *et al.*, "ClamAV anti-virus." http://www.clamav.net. Accessed: 2019-10-30.

[45] L. Tan and T. Sherwood, "A high throughput string matching architecture for intrusion detection and prevention," in *32nd International Symposium on Computer Architecture (ISCA'05)*, pp. 112–122, IEEE, 2005.

[46] N. L. Or, X. Wang, and D. Pao, "MEMORY-based hardware architectures to detect ClamAV virus signatures with restricted regular expression features," *IEEE Transactions on Computers*, vol. 65, no. 4, pp. 1225–1238, 2015.

[47] P. Dlugosch, D. Brown, P. Glendenning, M. Leventhal, and H. Noyes, "An efficient and scalable semiconductor architecture for parallel automata processing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3088–3098, 2014.

[48] E. Sadredini, R. Rahimi, M. Lenjani, M. Stan, and K. Skadron, "Impala: Algorithm/architecture co-design for in-memory multi-stride pattern matching," in *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 86–98, IEEE, 2020.

[49] A. Subramaniyan, J. Wang, E. R. Balasubramanian, D. Blaauw, D. Sylvester, and R. Das, "Cache automaton," in *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 259–272, 2017.

[50] F. Yu, R. H. Katz, and T. V. Lakshman, "Gigabit rate packet pattern-matching using TCAM," in *Proceedings of the 12th IEEE International Conference on Network Protocols, 2004. ICNP 2004.*, pp. 174–183, IEEE, 2004.

[51] C.-C. Wang, C.-J. Cheng, T.-F. Chen, and J.-S. Wang, "An adaptively dividable dual-port BiTCAM for virus-detection processors in mobile devices," *IEEE journal of solid-state circuits*, vol. 44, no. 5, pp. 1571–1581, 2009.

[52] M. Alicherry, M. Muthuprasanna, and V. Kumar, "High speed pattern matching for network IDS/IPS," in *Proceedings of the 2006 IEEE International Conference on Network Protocols*, pp. 187–196, IEEE, 2006.

[53] Q. Dong, S. Jeloka, M. Saligane, Y. Kim, M. Kawaminami, A. Harada, S. Miyoshi, D. Blaauw, and D. Sylvester, "A 0.3 V VDDmin 4+ 2T SRAM for searching and in-memory computing using 55nm DDC technology," in *2017 Symposium on VLSI Circuits*, pp. C160–C161, IEEE, 2017.

[54] S. Jeloka, N. B. Akesh, D. Sylvester, and D. Blaauw, "A 28-nm Configurable Memory (TCAM/BCAM/SRAM) Using Push-Rule 6T Bit Cell Enabling Logic-in-Memory," *IEEE Journal of Solid-State Circuits*, vol. 51, pp. 1009–1021, Apr.

2016.

[55] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, 2017.

[56] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *arXiv:quant-ph/9508027*, Jan. 1996. arXiv: quant-ph/9508027.

[57] O. Regev, "On Lattices, Learning with Errors, Random Linear Codes, and Cryptography," *J. ACM*, vol. 56, Sept. 2009.

[58] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of Computation*, vol. 19, pp. 297–297, May 1965.

[59] T. Pöppelmann and T. Güneysu, "Area optimization of lightweight lattice-based encryption on reconfigurable hardware," in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2796–2799, June 2014. ISSN: 2158-1525.

[60] U. Banerjee, T. S. Ukyab, and A. P. Chandrakasan, "Sapphire: A Configurable Crypto-Processor for Post-Quantum Lattice-based Protocols," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. Volume 2019, pp. 17–61 Pages, Aug. 2019. arXiv: 1910.07557.

[61] S. S. Roy, F. Vercauteren, N. Mentens, D. D. Chen, and I. Verbauwhede, "Compact Ring-LWE based Cryptoprocessor," Tech. Rep. 866, 2013.

[62] S. Song, W. Tang, T. Chen, and Z. Zhang, "LEIA: A 2.05mm^2 140mW lattice encryption instruction accelerator in 40nm CMOS," in *2018 IEEE Custom Integrated Circuits Conference (CICC)*, (San Diego, CA), pp. 1–4, IEEE, Apr. 2018.

[63] N. Verma, H. Jia, H. Valavi, Y. Tang, M. Ozatay, L. Chen, B. Zhang, and P. Deaville, "In-Memory Computing: Advances and Prospects," *IEEE Solid-State Circuits Magazine*, vol. 11, no. 3, pp. 43–55, 2019. IEEE Solid-State Circuits Magazine.

[64] H. Nejatollahi, S. Gupta, M. Imani, T. S. Rosing, R. Cammarota, and N. Dutt, "CryptoPIM: In-memory Acceleration for Lattice-based Cryptographic Hardware," Tech. Rep. 276, 2020.

[65] J. Buchmann, F. Göpfert, T. Güneysu, T. Oder, and T. Pöppelmann, "High-Performance and Lightweight Lattice-Based Public-Key Encryption," in *Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security - IoTPTS '16*, (Xi'an, China), pp. 2–9, ACM Press, 2016.

[66] V. Lyubashevsky, C. Peikert, and O. Regev, "On Ideal Lattices and Learning with Errors Over Rings," Tech. Rep. 230, 2012.

[67] M. Horowitz, "1.1 Computing's Energy Problem (and What We Can Do about It)," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 10–14, Feb. 2014.

[68] M. Kang, M.-S. Keel, N. R. Shanbhag, S. Eilert, and K. Curewitz, "An Energy-Efficient VLSI Architecture for Pattern Recognition via Deep Embedding of

Computation in SRAM," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8326–8330, May 2014.

[69] H. Valavi, P. J. Ramadge, E. Nestler, and N. Verma, "A 64-Tile 2.4-Mb In-Memory-Computing CNN Accelerator Employing Charge-Domain Compute," *IEEE Journal of Solid-State Circuits*, vol. 54, pp. 1789–1799, June 2019.

[70] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Strachan, M. Hu, R. S. Williams, and V. Srikumar, "ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, pp. 14–26, 2016.

[71] Y. Zhang, L. Xu, K. Yang, Q. Dong, S. Jeloka, D. Blaauw, and D. Sylvester, "Recryptor: A reconfigurable in-memory cryptographic Cortex-M0 processor for IoT," in *2017 Symposium on VLSI Circuits*, pp. C264–C265, 2017.

[72] C. Eckert, X. Wang, J. Wang, A. Subramaniyan, R. Iyer, D. Sylvester, D. Blaaauw, and R. Das, "Neural Cache: Bit-Serial In-Cache Acceleration of Deep Neural Networks," in *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, pp. 383–396, June 2018. ISSN: 2575-713X.

[73] H. Nejatollahi, S. Shahhosseini, R. Cammarota, and N. Dutt, "Exploring Energy Efficient Quantum-resistant Signal Processing Using Array Processors," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1539–1543, May 2020. ISSN: 2379-190X.

[74] 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, 2016.

[75] U. Banerjee, A. Pathak, and A. P. Chandrakasan, "2.3 An Energy-Efficient Configurable Lattice Cryptography Processor for the Quantum-Secure Internet of Things," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, pp. 46–48, Feb. 2019. ISSN: 2376-8606.

[76] C. Peikert, "An efficient and parallel Gaussian sampler for lattices," in *Annual Cryptology Conference*, pp. 80–97, Springer, 2010.