Lightweight Encryption for Resource-Constrained Systems

Thesis presented

by

Bryan Moy

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Master of Science

in

Electrical Engineering

Stony Brook University

December 2020

Stony Brook University

The Graduate School

Bryan Moy

We, the thesis committee for the above candidate for the
Master of Science degree, hereby recommend
acceptance of this thesis

Dr. Emre Salman - Thesis Advisor
Associate Professor, Department of Electrical and Computer Engineering

Dr. Milutin Stanacevic - Second Reader
Associate Professor, Department of Electrical and Computer Engineering

This thesis is accepted by the Graduate School

Eric Wertheimer
Dean of the Graduate school

Abstract of the Thesis

Lightweight Encryption for Resource-Constrained Systems

by

Bryan Moy

Master of Science

in

Electrical Engineering

Stony Brook University

2020

Resource-constrained systems such as the Internet-of-things based devices have become increasingly more common. Each functionality within these systems needs to be optimized for power consumption. One such functionality is encryption to satisfy security and privacy concerns.

A lightweight SIMON cipher core was developed in 0.5μm CMOS technology targeting resource-constrained systems such as RFID applications and wireless sensors. The design employs a round-unrolled serial-to-parallel architecture to improve throughput while not compromising the benefits of low power consumption and area. The design was first verified via simulations. It was then fabricated and experimentally tested with a custom printed circuit board (PCB). The fabricated chip dissipates an average power of 1.03mW during encryption, achieves an encryption efficiency of 3.29Kb/sec/μW and consumes an area of 850μm$\times$850μm.

In the second part of the thesis, a primary building block of advanced encryption standard (AES) based cipher, substitution-box (S-Box), was investigated using Fermat's Little Theorem. It was compared against a more common lookup table based implementation of the S-Box in both FPGA and

ASIC platforms. Using a computed Galois-Field Arithmetic based S-Box value (as opposed to traditional lookup table) exhibited promising characteristics in higher performance systems since it could achieve higher throughput and operating frequency. Several different implementations of these designs were explored to assess their characteristics in terms of power, area, and throughput.

Dedication Page

For my loving parents, Wayne Moy and Shirley Moy

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AES - Advanced Encryption Standard
ASIC - Application Specific Integrated Circuit
CMOS - Complementary Metal-Oxide-Semiconductor
CPA - Correlation Power Analysis
CT - Ciphertext
DFF - D-Flip Flop
DUT - Device Under Test
EEA - Extended Euclidean Algorithm
ESD - Electrostatic Discharge
FLT - Fermat's Little Theorem
FPGA - Field Programmable Gate Array
HD - Hamming Distance
IC - Integrated Circuit
IoT - Internet-of-Things
LUT - Lookup-Table
MUX - Multiplexor
PT - Plaintext
RFID - Radio Frequency Identification
ROM - Read-Only Memory
SCA - Side-Channel Attack
SNR - Signal-to-Noise Ratio

Acknowledgement

I would like to take the opportunity to formally thank all the friends, family and staff at Stony Brook University who have helped me and guided me through undergraduate and graduate degrees.

I'm incredibly thankful for my thesis advisor, Dr. Emre Salman for his guidance and patience with during my time working with him. I learned the value of hard work and patience that I will carry with me into my career and beyond. Through him, I felt that things didn't need to be as complicated as I had made them out to be and that we can work towards excellence with cautious-optimism. Without him and his lab I would not have gotten this far.

I extend my thanks to Dr. Milutin Stanaćević, the second reader for this thesis, for taking the time to review and help me improve my contributions to the field, from my first integrated circuits class to my graduate thesis.

Many thanks for the lab members of NanoCAS who helped me during spontaneous lab visits. Manav Jain, Ivan Miketic, Mallika Rathore, Tutu Wan and Krithika Yethiraj provided years of insight and reassurance during difficult times.

The diligence, meticulousness and cooperation from my teammates William Lee and Nicholas St. John have made our design a success and I'm so proud to have had them alongside me for my senior year and to have made something so memorable.

Lastly I need to thank my mother and sister for their support and inspiration every time I've needed them. Despite whatever obstacles have come our way we have and always will persevere to make every day "good-la".
Thank you all.

Chapter 1

Introduction

With tremendous growth of the Internet-of-things (IoT) and increasing interconnectivity of devices and networks, the amount of sensitive data traffic has significantly increased. With networking starting to be present on everyday devices and appliances, there are growing concerns related to security and privacy [1]. The advanced encryption standard (AES), developed in 2001 by the National Security Agency (NSA), has been used for the better part of previous two decades as the standard/basis for securing wireless data. AES, however, is not sufficiently applicable to highly resource-constrained systems with energy harvesting due to high overhead in both power consumption and area.

Resource-constrained systems that transmit sensitive data have two primary security requirements: (a) efficient and lightweight encryption and (b) protection against possible attacks on the encrypted data. SIMON is a lightweight encryption algorithm that was also developed by NSA [2]. SIMON is designed to minimize power usage and latency while also providing a sufficient layer of security. It is a configurable algorithm depending upon the desired security level of the application. Low overhead in terms of both power and area

makes SIMON a strong candidate for energy harvesting applications [3, 4, 5, 6, 7, 8]. For example, an AC computing methodology was proposed to facilitate SIMON-based lightweight encryption for wirelessly powered devices [8]. Emerging technologies such as monolithic 3D integration were also leveraged to assess bit-serialized implementations of SIMON [9, 10, 11, 12, 13].

With the exchange of sensitive information, it has been highly critical to protect these systems against malicious attacks. On the hardware side, these attacks are often in the form of side-channel analysis based attacks [1, 14]. A side-channel attack can retrieve the secret key in encryption ciphers using leaked information from the encryption hardware such as the power consumption, electromagnetic radiation, thermal emissions, or even acoustic signals [15]. Correlation power analysis (CPA) is a common form of side-channel analysis attack that can be performed on encryption hardware to retrieve secret key in a relatively short amount of time (assuming an unprotected cipher) [14].

The primary objectives of this thesis are (1) to develop a prototype of a low power SIMON core with application to resource-constrained systems such as RFID circuits and wireless sensor networks, (2) experimentally analyze the characteristics of its power profile under a typical power-based side-channel analysis attack, (3) and explore several substitution-box (S-Box) based AES algorithms/topologies for encryption. The rest of the thesis is organized as follows. Chapter 2 provides the background of existing encryption algorithms considered in this work and the challenges related to hardware implementation. It also discusses the basis for side-channel vulnerabilities and methods to exploit them to retrieve the secret key. Chapter 3 discusses the SIMON encryption algorithm and a test chip of an implemented design in 0.5μm CMOS technology. There is also an assessment of a side-channel attack in hardware and the methodology used to attack the aforementioned SIMON implementa-

tion. Chapter 4 discusses typical implementations of an AES S-Box and compares several different hardware implementations including the quantification of ASIC vs. FPGA tradeoffs. Finally, the thesis is concluded in Chapter 5.

Chapter 2

Background

2.1 SIMON Encryption Algorithm

SIMON is a lightweight Feistel block cipher that was developed by the National Security Agency (NSA) in 2013, as detailed in [2]. SIMON is an attractive encryption algorithm because of its simpler set of operations compared to more complex algorithms such as AES, making it more applicable to resource-constrained systems that rely on energy harvesting.

SIMON also has multiple available configurations for variable degrees of security (depending on the application). Furthermore, SIMON can be implemented in varying forms of parallelism ranging from a bit-serialized implementation (least hardware resources) to fully parallel unrolled system (most hardware resources).

The encryption has an n-bit word plaintext ($2n$-bit block; X_{LR}) and an m-word key (mn-bit block; K_i). These different variations of SIMON algorithm/cipher are represented as SIMON *2n/mn*. The cipher is comprised of a round function and key expansion. The design discussed in this thesis is an

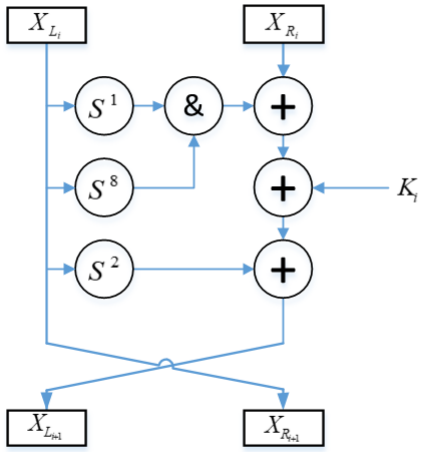

Figure 2.1: Round function

implementation of the SIMON32/64 configuration where $n = 16$ and $m = 4$.

The round function is a set of rotate-left (denoted by S^n where n is the number of rotations), XOR and AND operations defined by the following equation,

$$f(X_L, X_R, K_i) = [S^1(X_L)\&S^8(X_L)] \oplus S^2(X_L) \oplus X_R \oplus K_i, \qquad (2.1)$$

where X_L represents the higher significant bits of the plaintext, X_R represents the lower significant bits of the plaintext and K_i represents the key bits for that round. Round encryption is performed through the 3 left-rotates (S^n), 3 XORs and a single AND operation per bit, as shown in Fig. 2.1.

The key length determines the security strength of a given cipher. The key in SIMON is an mn-bit block seed used to generate keys for the remaining rounds. The key for SIMON can be configured with 2, 3 or 4 blocks. The seed is 4× 16-bit blocks that generate a key schedule for the remaining 28 rounds of encryption for a total of 32 rounds of encryption. The key expansion is performed through 2 right-rotates (S^{-n}) and 5 XORs per bit, as illustrated in

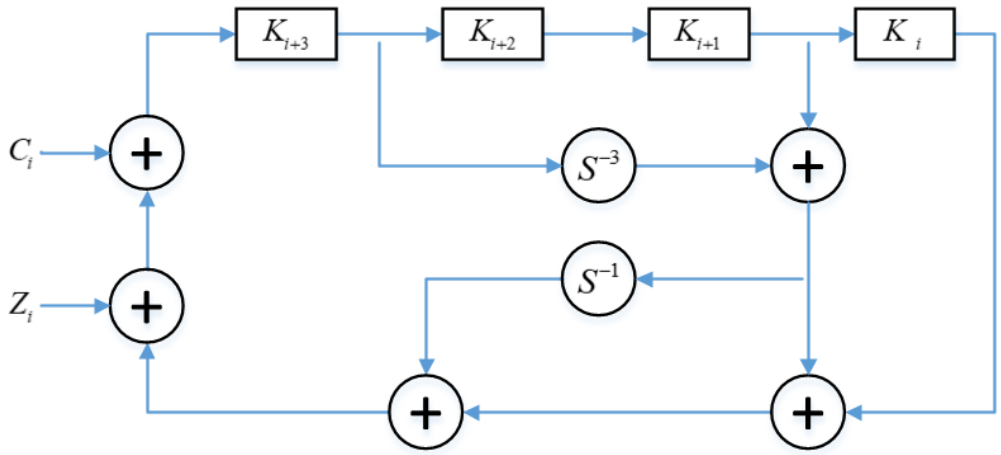

Figure 2.2: Key expansion with $m = 4$

Fig. 2.2. Constant C and Z bits are used by the key expansion to eliminate slide properties and circular shift symmetries [2].

The key schedule is generated by the following key expansion function,

$$T_i = S^{-3}(K_{i+3}) \oplus K_{i+1}, \tag{2.2}$$

$$K_{i+4} = K_i \oplus (T_i) \oplus S^{-1}(T_i) \oplus C_i \oplus Z_i. \tag{2.3}$$

For $i = 0$ until $i = 28$, corresponding indices from C and Z constants are used for the key expansion.

An interesting feature of SIMON is its flexibility in folding and unfolding in multiple dimensions. For example, a single round operation can be performed on a subset of bits or in bit-serial form. Each of the n rounds can be designed to compute a subset or all of the 32 encryption rounds at once through varied levels of parallelism.

	00	01	02	03	04	05	06	07	08	09	0A	0B
00	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B
10	CA	82	C9	7D	FA	69	47	F0	AD	D4	A2	9C
20	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1
30	4	C7	23	C3	18	96	05	9A	07	12	80	E2
40	9	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3
50	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39
60	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F
70	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21
80	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D
90	60	81	4F	DC	22	2A	90	88	46	EE	B8	14
A0	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62
B0	E7	C8	37	6d	8D	D5	4E	A9	6C	56	F4	EA
C0	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F
D0	70	3E	B5	66	48	03	F6	0E	61	35	57	B9
E0	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9
F0	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F

Figure 2.3: 256-entry S-Box

2.2 Advanced Encryption Standard (AES)

Introduced in 2003 and detailed in [16], AES has become one of the most widely used encryption algorithms. An intermediate step of the algorithm uses a lookup table (LUT) referred to as substitution-box (S-Box) as part of its encryption. This S-Box represents a mapping of an 8-bit byte to a new value. The contents of the 256-entry S-Box are affine transformed multiplicative inverses in a Galois Field for all of the 256 input combinations [16]. An efficient implementation of the S-Box step of AES is required since it typically consumes 80% of the area and account for 50% of the propagation delay [17].

These substitute values (as shown in Fig. 2.3) can be obtained by manually

computing them in a finite-field from the input or through a ROM based LUT. Some of the common implementations for finding the multiplicative inverse in a Galois Field include: LUT [18], Fermat's Little Theorem (FLT) [19], and the Extended Euclidean Algorithm for Greatest Common Divisor (EEA-GCD) [20].

Since the S-Box is computationally expensive to compute in a finite-field, it is often implemented as a LUT. In this form, the S-Box is a weak point for the encryption in terms of side-channel security because of its direct mapping of input to output value. Analysis of the clock cycle where the S-Box lookup occurs can be used to exploit the leaked power characteristics of the circuit [21]. Some implementations such as EEA-GCD have variable run time, making it susceptible to timing based side-channel attacks [20]. In chapter 4, an implementation of Galois Field Arithmetic (GFA) using Fermat's Little Theorem (FLT) is demonstrated.

2.3 Correlation Power Analysis Attacks

When an encryption is performed, conceptually there is only the key, input plaintext and output ciphertext. However, other distinctive traits of an encryption hardware are present during its operation. Electromagnetic fields [22], temperature [23], timing [14] and power consumption [14] can all be observed and measured for different inputs. These unintended side-effects are often called side-channel leakage, and can be exploited to retrieve secret key bits of the encryption algorithms via various statistical analyses. In recent years, interest has risen in the study of these side-channel attacks and the defense mechanisms (countermeasures) to protect the encryption circuits against these attacks [14].

In correlation power analysis attacks, by leveraging the statistical concept of correlation, an attacker can input a plaintext of their choice for encryption (with the unknown secret key) while recording the current/power drawn from the device under attack. After capturing a large number of power traces (typically in the range of several to tens or hundreds of thousands), an attacker can potentially correlate the measured results at a given time instance against another set of results obtained with specific key guesses [24]. The Pearson correlation coefficient is used as a metric to find a correlation value for two data sets X and Y between -1 and +1, where a correlation value of +1 indicates a positive and linear relationship, -1 is a negative linear relationship, and 0 indicates that there is no linear correlation. The Pearson correlation coefficient is determined by the following equation,

$$\rho(X,Y) = \frac{\mathbf{Cov}(X,Y)}{\sqrt{\mathbf{Var}(X)\mathbf{Var}(Y)}}, \tag{2.4}$$

where covariance function **Cov** and variance function **Var** refer, respectively to,

$$Cov(X,Y) = [([\mathbf{X} - \mathbf{E}[\mathbf{X}]) \cdot (\mathbf{Y} - \mathbf{E}[\mathbf{Y}])], \tag{2.5}$$

$$Var(X) = \mathbf{E}[([\mathbf{X} - \mathbf{E}[\mathbf{X}])^{\mathbf{2}}], \tag{2.6}$$

where $E[X]$ and $E[Y]$ refer, respectively, to the expected value for given sets X and Y. For a correct key guess, we can expect a higher correlation between the power drawn at a given moment and the Hamming distance associated with it. Hamming distance is defined as the number of 0-to-1 or 1-to-0 transitions from one data to another. For transitions of 0-to-1, there is a linear and positive relationship between power dissipation and setting of the bits. With a given key guess, a plaintext is transformed from one state to a next, and therefore

has a corresponding Hamming distance that can be used for CPA attack.

In encryption algorithms such as AES and SIMON, the plaintext is translated from its initial value into some partially encrypted value (intermediate state). This process is repeated for a given number of rounds until the data is fully encrypted. Using correlation, an attacker captures some number of power traces for a given plaintext at an intermediate state and correlates it to all possibilities of a key or sub-key. The correlation is seen through the number of bits switching from one round to the next (i.e. Hamming distance) and their key guesses. Once a sufficient number of power traces has been acquired, a Pearson correlation coefficient can be used to confirm the key guess as the correct key. Unlike a brute force attack, these sub-key guesses can exist within 2^{16} or less, making it feasible to retrieve key bits in some cipher implementations within minutes [14, 25]. In the case of SIMON32/64, there are 2^{64} key guesses for its 64-bit key in brute-force attacks, but successfully attacking a sub-key of size 4-bits could be as few as 256 key guesses (when applied to all of the 16-subkeys) as in [26], which is 17 orders of magnitude fewer key guesses than a brute force attack.

There have been successful attacks on bit serialized forms of SIMON with as few as 1,300 traces [26]. For more parallel systems, 16-bit and 64-bit data paths were attacked, but as parallelism increased, so did attack complexity, failing to retrieve the key even at 500K traces. Fully unrolled implementations are not suitable for lightweight hardware despite their advantages in computational acceleration. Options such as unrolling 7 rounds into cascaded encryptions (reduced cycle) have demonstrated promise as a compromise in performance for better security [26].

Bit-serial implementations are highly susceptible to this kind of attack because of the lack of other parallel switching elements that diffuse the encryp-

tion side-channel leakage such as power consumption (signal-to-noise ratio is reduced as the number of elements switching at a given time increases). For example, in [26] and [27], hundreds of thousands of traces are needed to retrieve the key bits. When parallelism is implemented in some degree, the bits flipping are layered, and therefore more noise from peripheral circuits are present in the system. Thus, signal-to-noise ratio drops to a point where correlation is too low to measure the power at a distinct location reliably and repeatedly. To mitigate this issue, an attacker can increase the number of key bits they attack for correlation, but this approach assumes the added key bits being guessed switch at the same time. It also increases the complexity of sub-key guesses to a magnitude where it is effectively not feasible (similar to a brute force attack) [26].

In modern countermeasures against side-channel attacks, masking circuits or varied architectures that do not compromise system operation or efficiency are adopted. An example for masking is described in [28]. Another approach described in [26] leverages SIMON's flexible configuration schemes to mitigate these side-channel leakages.

Chapter 3

Hardware Realization of Lightweight SIMON Core

3.1 System Architecture for SIMON 32/64

The SIMON algorithm can be implemented with various levels of parallelization, ranging from bit level serial operation to full encryption level parallelism, depending on application and related constraints on hardware overhead and performance. This thesis is focused on the implementation of a round-parallel based architecture, offering higher throughput than the bit-serialized architecture at the expense of greater power consumption and area, but not as costly as a fully unrolled pipelined parallel implementation. Thus, it is an intermediate level of parallelism that strikes a reasonable balance between performance and overhead. A top-level view of the system is shown in Fig. 3.1. The plaintext is stored in 32 D flip-flop (DFF) based registers and the key is stored in 64 DFF based registers for a total of 96 storage elements. The design takes in key and plaintext data through a serial interface and into a first-in-first-out

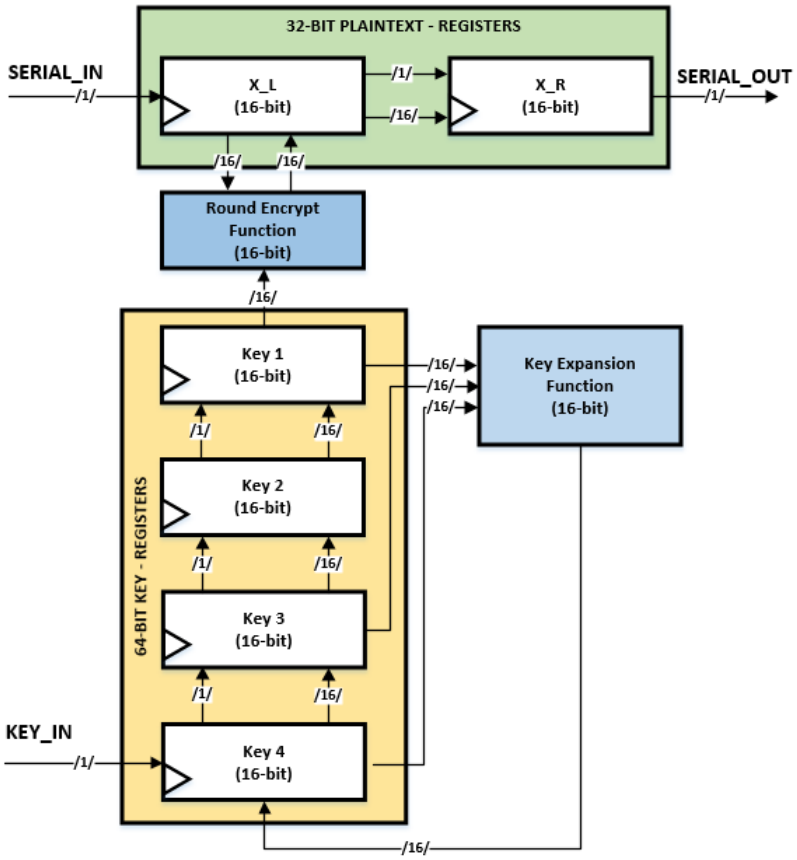

Figure 3.1: Top-level design with the proposed round-parallel based architecture

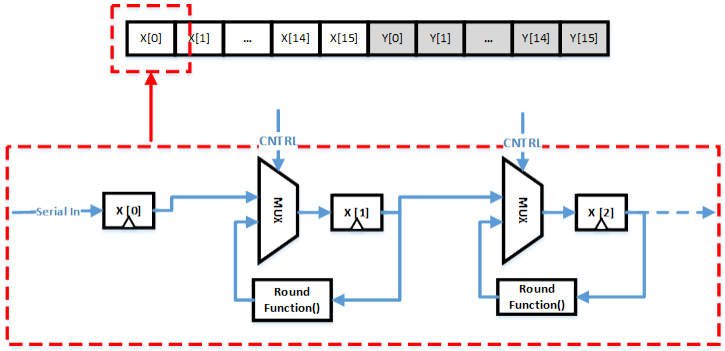

Figure 3.2: Interlaced MUX register scheme

(FIFO) register. Here, it performs encryption at the round level (32-bit parallelism). 2-1 MUX has been interlaced into the registers storing the plaintext and keys to change data flow to the parallel setup after the initial inputs have been imported serially (see Fig. 3.2).

The circuit requires the following clock cycles to perform 32-bit serial encryption:

- 64 cycles to import plaintext (PT) and key
- 32 cycles to encrypt
- 32 cycles to export ciphertext (CT)

In ENCRYPT mode, 32 plaintext registers drive the encryption hardware which replaces the previous state. The key scheduling is performed using a similar scheme. After 32 clock cycles, the plaintext is fully encrypted and the select inputs for the MUX revert to a shift-out FIFO state. The ciphertext is then shifted out serially. A control unit comprised of a 7-bit counter and decoders determine the data flow for key scheduling and round encryption. The timing characteristics of the entire process and the operating modes are illustrated in Fig. 3.3 and listed in Table 3.1.

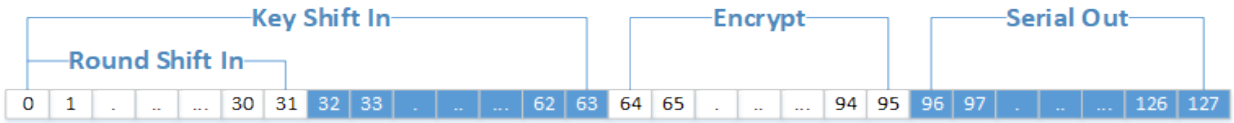

Figure 3.3: System operating regions illustrating the number of clock cycles for each region

CLK Counter	Round MUX Mode	Key MUX Mode	Description
0	FIFO	FIFO	Import plaintext (PT) and key
32	IDLE	FIFO	Finish storing PT, Store second half of key
64	ENCRYPT	ENCRYPT	Key is stored, begin encryption
96	FIFO	FIFO	Finished encryption, serially shift out ciphertext (CT)
128	FIFO	FIFO	CT sent out and system is reset

Table 3.1: Description of the system operating modes

Description	**Bit-Serial**	**Parallel**	**Round-Parallel (proposed)**
	Pins	Pins	Pins
VDD	1	1	1
VSS	1	1	1
CLK	1	1	1
RESET	1	1	1
Plaintext	1	32	1
Key	1	64	1
Ciphertext	1	32	1
Total	7	132	7

Table 3.2: Number of IO pins needed for varying levels of parallelism

Fully serial or fully parallel implementations have appealing attributes for, respectively, hardware cost and execution time, but their limitations make these approaches impractical in many applications. For example, a fully parallel version of SIMON would have a significant number of IO ports that many systems many not be able to utilize. It also has a large footprint since a separate encryption block is needed for all of the 32 bits of all 32 rounds (1,024 encryption blocks). This issue is exacerbated if the design is pipelined with internal DFF registers. Alternatively, a bit-serial based design, although achieves low area and small number of IO ports, has poor latency and can take 1,088 cycles to compute a single encryption. Using 32-bits of parallelism, the proposed design approach in this thesis reduces encryption time by a factor of 8.5. Furthermore, by reusing encryption blocks, the proposed design reduces overall footprint by as much as 64 and can be easily integrated into larger digital systems because of its simpler interface. Comparison of these different approaches is listed in Table 3.2 and Table 3.3.

The design also features an active-low data-ready signal, active-low asynchronous reset, and internal debug signals for flagging the system's state of

Description	**Bit-Serial**	**Parallel**	**Round-Parallel (proposed)**
Transfer Overheads	64	0	96
Bit-level Encrypt	32	0	1
Round-level Encrypt	32	0	32
Clock Cycles	**1,088**	**1**	**128**
# Encryption cells	2	2,048	32

Table 3.3: Expected timing overhead and resource utilization

operation. The design has potential for integration into larger designs as a standalone encryption core at both embedded and silicon level. Minimum gate sizing was used to reduce load capacitance at internal nodes and therefore overall switching power consumption. Clock gating with asynchronous logic was also used to decrease dynamic power dissipation by preventing unnecessary switching of internal nodes. Power gating was used to further improve power efficiency by shutting off the supply voltage for idle computational blocks. Tapered buffers were designed to provide external drive strength when interfacing the pads with external capacitances on the order of tens of picofarads at tens of megahertz. A separate VDDIO rail was developed to power the aforementioned tapered buffers. Electrostatic discharge diodes (ESD) were placed on all of the data IO pins to prevent voltage spikes from reaching the sensitive gates in the core computational units. Multiple power supply input pins were used for both VDD and VDDIO to improve power delivery and reduce switching noise. The primary functional blocks are highlighted in Fig. 3.4 where the top-level layout is illustrated. A photo of the die is also shown in Fig. 3.5.

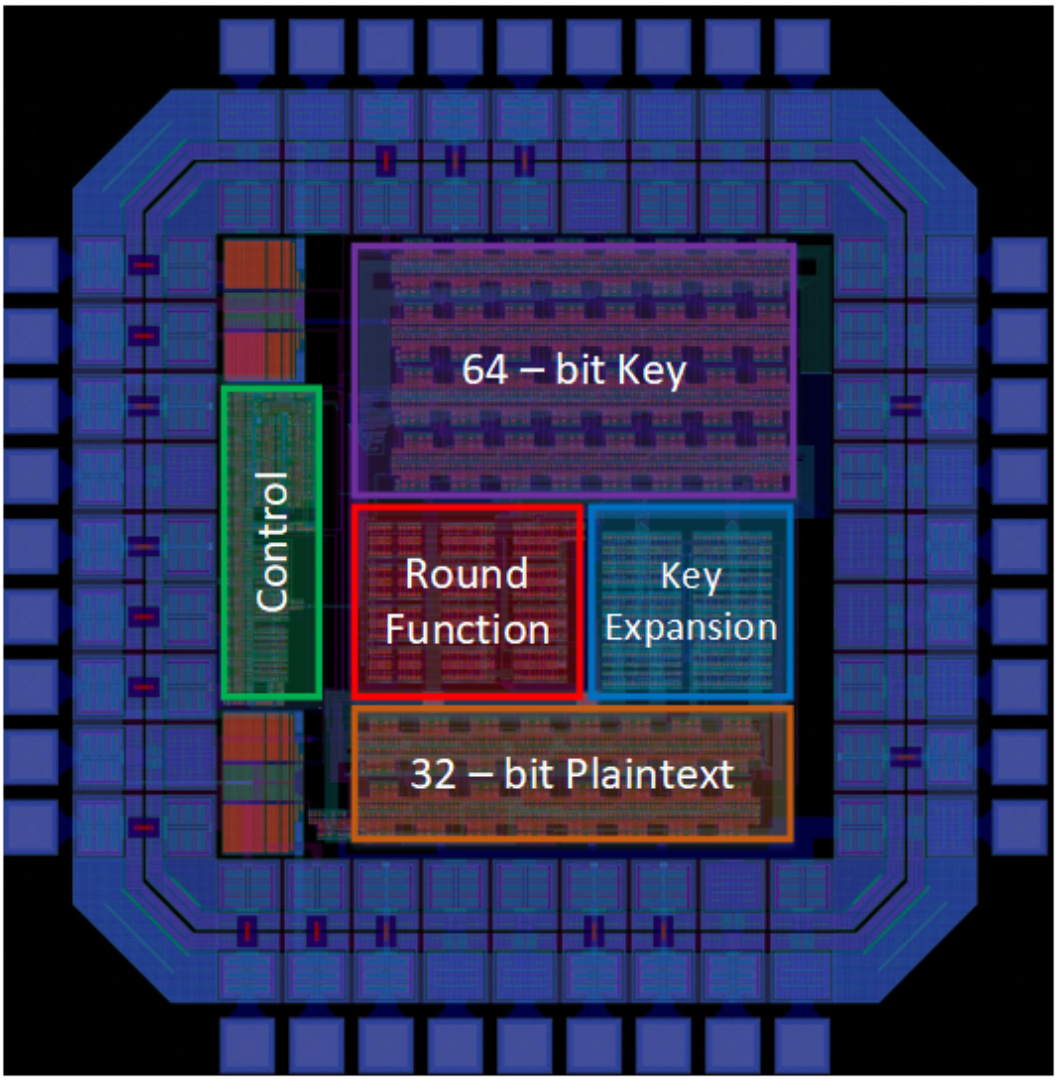

Figure 3.4: Top-level layout of the SIMON core illustrating the functional blocks, with an area of 0.85mm×0.85mm.

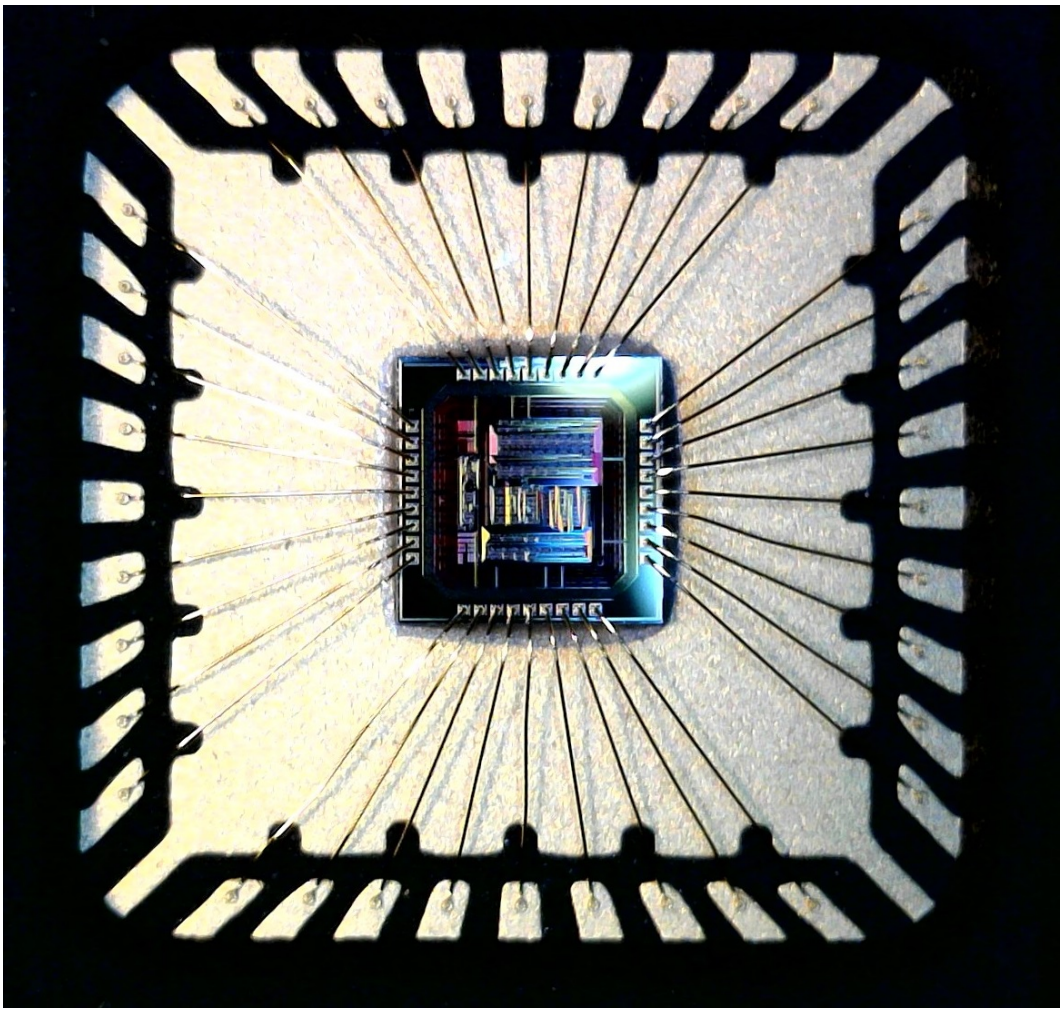

Figure 3.5: Die photo illustrating the SIMON core and QFN-36 package, with a footprint of 1.5mm×1.5mm

3.2 Test Methodology and Experimental Results

The design was fabricated by ON semiconductor's AMI 0.5μm CMOS technology node by the MOSIS multi wafer project. The design was placed into a quad-flat no-leads 36-pin package (QFN-36) and tested on an embedded

printed circuit board (PCB). This hardware realization is targeted towards RFID applications operating at 13.56MHz. Power supply has been separated into two rails: VDD-Core (for logic) and VDDIO (for input and output pads). Specifically, power is provided to the local computational blocks through VDD-core. The power rail used for driving loads at an embedded IO level (capacitance in the range of picofarads) is provided by VDDIO. Testing was done using an embedded system development board that communicates with a modified SPI driver to provide stimuli signals.

A 4-layer PCB was fabricated to verify the functionality, as shown in Fig. 3.6. The 16-pin header on the left side is used for driving input signals and providing power supply voltage. The supply voltage is provided via a low-dropout voltage regulator (LDO) with an input voltage configurable for either the 5V or 3.3V. Output pads are connected to unity gain buffers to reliably drive large load capacitance on the chip at higher frequencies. At the target frequency of 13.56MHz, the design requires decoupling capacitors to provide instantaneous current and maintain supply rail with low noise (10% of nominal voltage). The QFN-36 package was placed in a clamp shell socket for testing, as shown in Fig. 3.7.

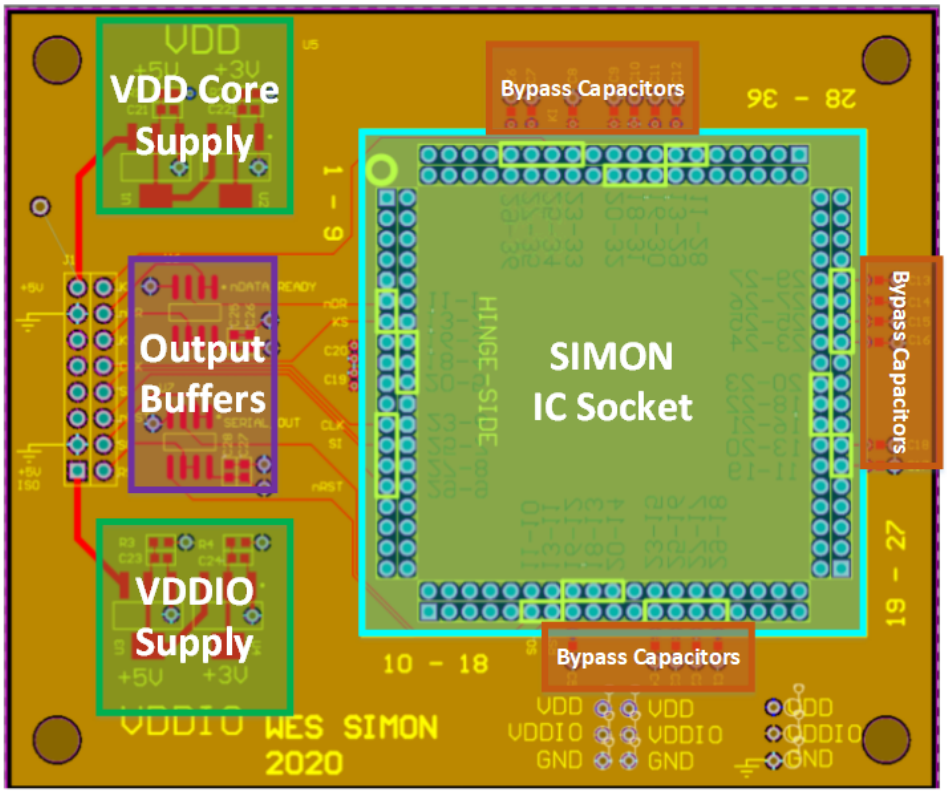

Figure 3.6: Printed circuit board used to test the fabricated SIMON core

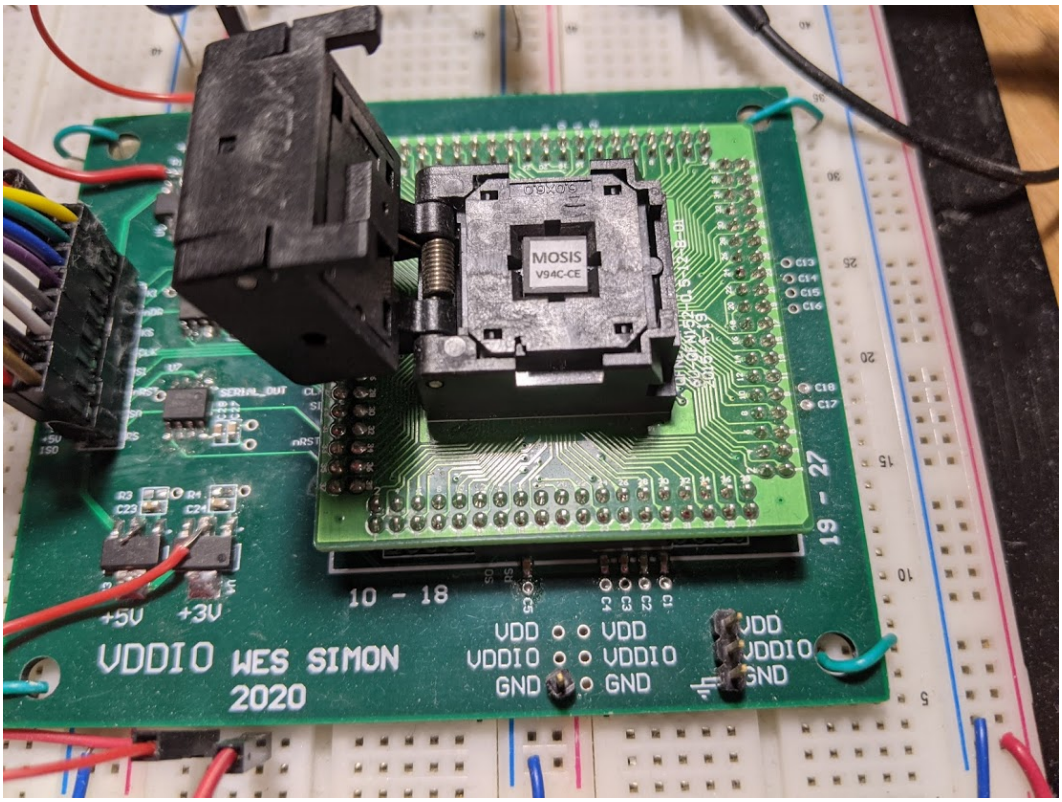

Figure 3.7: Clamp shell socket where the QFN-36 SIMON IC was placed

The test setup and flow, as conceptually illustrated in Fig. 3.8, is comprised of a DC switching power supply, a Rigol DS1054Z oscilloscope with 250MSa/s, an STM32F767 development board and a custom PCB that was described above. A C# graphical user interface (GUI) was coded to interface with the oscilloscope via LAN connection and communicate with the development board via USB-CDC connection. This test setup is depicted in Fig. 3.9. The SIMON IC encrypted the input correctly across over 200,000 plaintexts at various frequencies ranging from 100KHz to 18MHz. To demonstrate accurate functionality, some example waveforms obtained from the scope are illustrated in Figs. 3.10 to 3.13 for different keys and plaintexts.

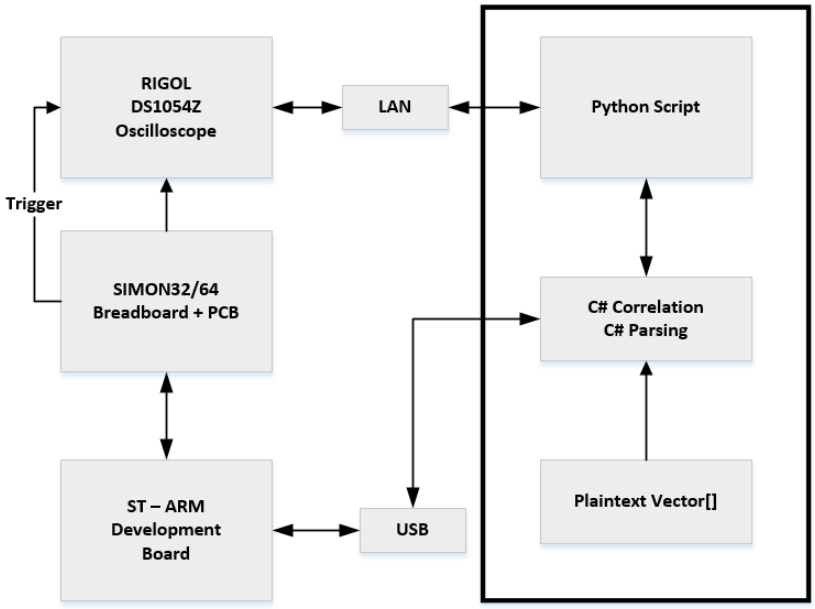

Figure 3.8: Measurement setup and flow including the equipment used

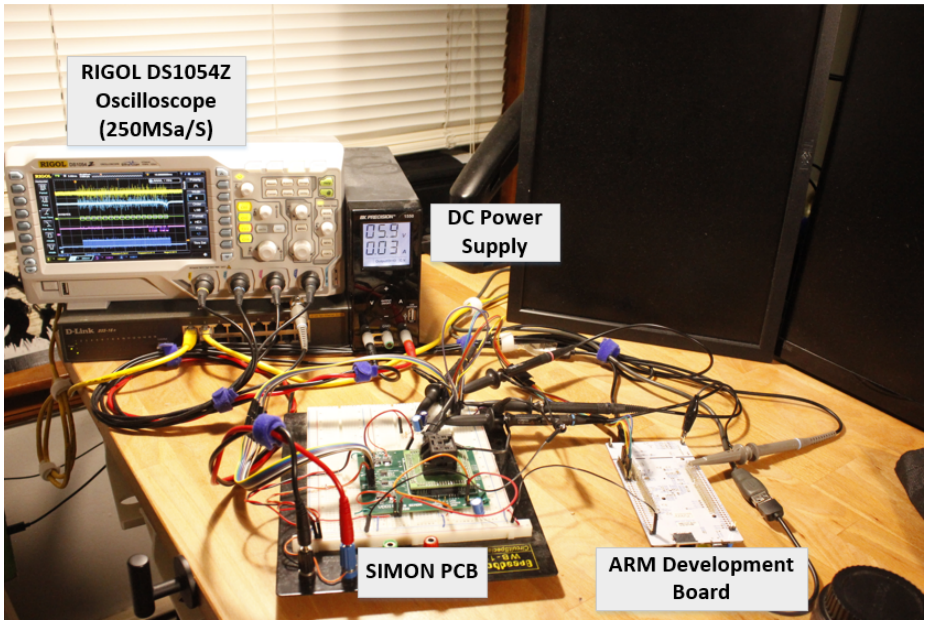

Figure 3.9: Measurement setup used to demonstrate accurate functionality and investigate power-based side-channel analysis attack

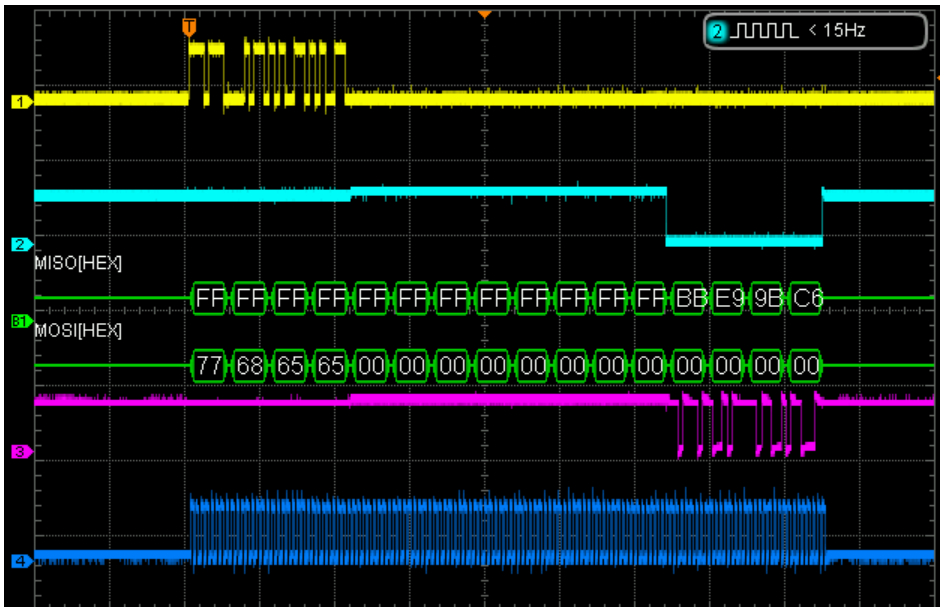

Figure 3.10: Functional operation for PT vector *0x6565 6877* and CT result *0xC69B E9BB* for key *0x1918 1110 0908 0100* on oscilloscope

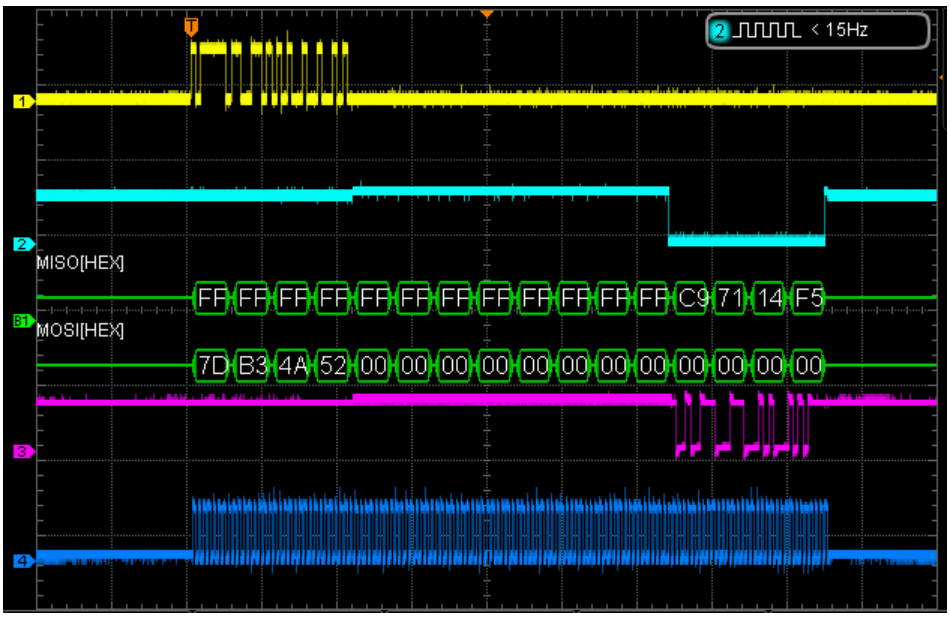

Figure 3.11: Functional operation for PT vector *0x524A B37D* and CT result *0xF514 71C9* for key *0x1918 1110 0908 0100* on oscilloscope

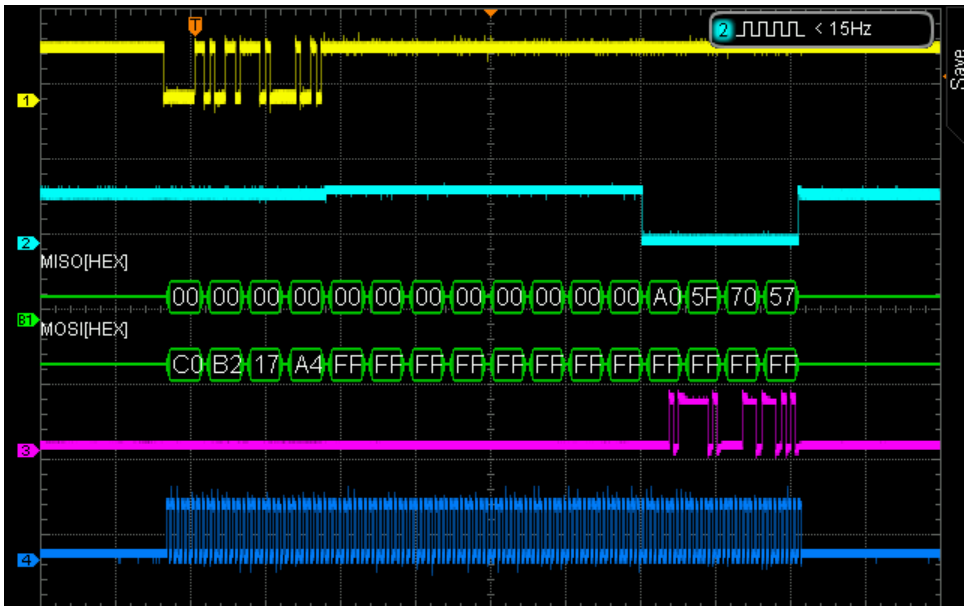

Figure 3.12: Functional operation for PT vector *0xA417 B2C0* and CT result *0x5770 5FA0* for key *0xA9B3 C891 DFF3 5912* on oscilloscope

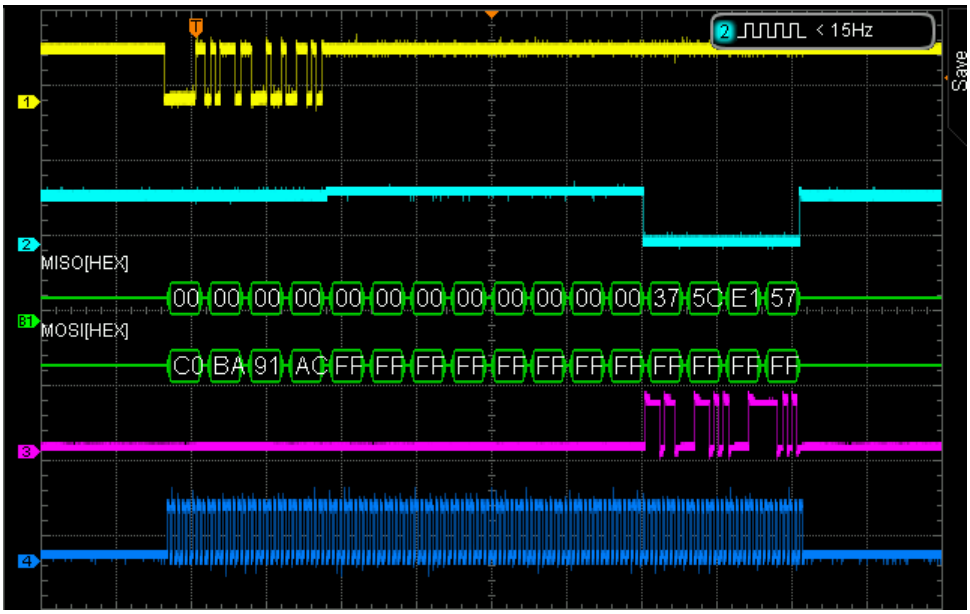

Figure 3.13: Functional Operation for PT vector *0xAC91BAC0* and CT result *0x57E1 5C37* for key *0x1029 3847 56AF EDB3* on oscilloscope

3.3 Comparison of Simulated and Measured Data

The design was characterized through core power consumption, area, and throughput, as listed in Table 3.4. According to these results, the post-layout simulation results and measurement data are sufficiently close in core power dissipation where the measured power is slightly less. Simulations slightly overestimated the power results because of the larger load capacitance used on all 4 output pins. A majority of the total power in this system is consumed by VDDIO for shifting out the encrypted data. Encryption efficiency of the core was measured as the throughput per μW (3.25 for simulation vs. 3.29 for measurement). Post-layout simulations (see Fig. 3.14) demonstrate that the increased latency due to parasitic impedances is negligible in this technology node at the target frequency of 13.56MHz. Simulated results demonstrate a maximum operating frequency of 26MHz. The measurement results demonstrate a maximum operating frequency of 18MHz due to the development board used, which was the bottleneck.

	Simulation	**Measurement**
Average Core Power(μW)	1041	1029
Core Energy (mJ)	9.83	9.72
Core Efficiency (Kb/sec/μW)	3.25	3.29
Avg IO Power(mW)	10.33	36.41

Transistor count	6278
Area	850μm x 850μm
Throughput(Kbps)	3390

Table 3.4: Comparison of simulated and measured characteristics of the SIMON core

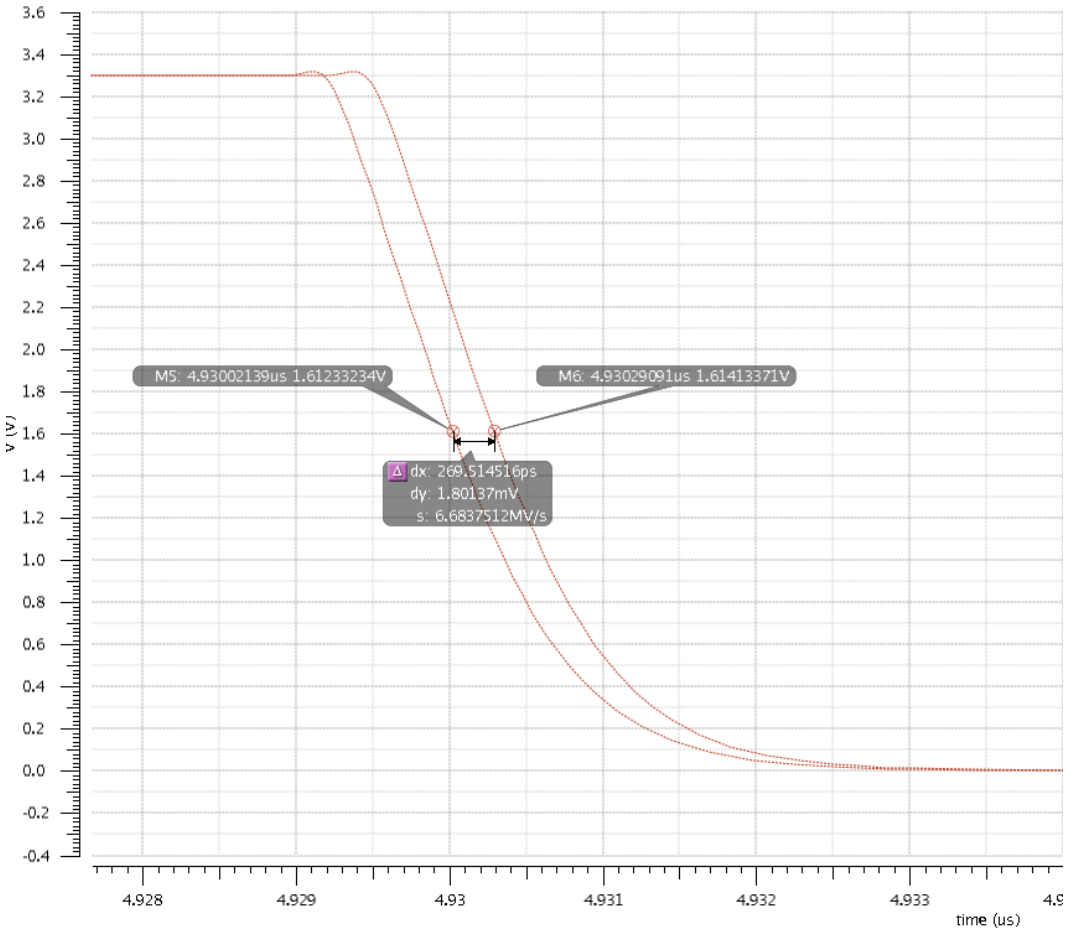

Figure 3.14: Simulated post layout signal with parasitic impedances plotted with the ideal signal, illustrating a slight delay of 270ps

3.4 Side Channel Attack Methodology

A series of power-based side-channel attacks were performed on the fabricated chip by using correlation and measuring differential power for various input traces. The measurement setup shown in Fig. 3.9 was used. Correlation power analysis (CPA) based attacks correlate power drawn from the device at a given instance to key guesses.

For SIMON 32/64, in the first four rounds of encryption, the first four keys of the key schedule are used and are the optimal points in time to correlate the known input plaintext against key guesses because the initial keys have not been diffused into the encryption. An attacker has knowledge of the input plaintext, but not the key being used. For a given plaintext and key, there is a partial ciphertext generated during encryption. The Hamming distance between the original plaintext and the aforementioned partial encryption should reflect in the power traces. When measured across tens or hundreds of thousands of power traces during these first four cycles, an attacker can correlate a set of key guesses with the power profile to determine which key guess matches the power profile best. The Pearson correlation coefficient generalized by (2.4) is applied to the data sets of this system,

$$r(k,t) = \frac{\Sigma_{n=1}^{N}(h_{n,k} - \bar{h_k}) \cdot (p_{n,t} - \bar{p_t})}{\Sigma_{n=1}^{N}(h_{n,k} - \bar{h_k})^2 \cdot (p_{n,t} - \bar{p_t})^2}. \tag{3.1}$$

The hypothetical power matrix is *h(n,k)*, where $n = 1, 2...N$, where N is the total number of plaintexts and $k = 1, 2, ...K$, where K is the total number of sub-key guesses. The measured power matrix is *p(n,t)*, where $n = 1, 2...N$, where N is the same total number of plaintexts and $t = 1, 2...T$, where T is the total number of samples. For a given time instance t in power trace n, there is an instantaneous power. The power p at this instance t across all N plaintext encrypted traces should correlate with a hypothetical power matrix. Correlating these two data sets, there should exist a correlation against the measured power traces at some time instant t and some key guess k. The largest correlation coefficient value should reveal what the key is. Knowledge of the timing characteristics of the circuit at the transistor-level can substantially reduce the maximum number T required for a successful attack.

Based on the architecture of the proposed SIMON implementation, 32-bits of a round are computed in each clock cycle, 16 of which are loaded in from a previous state and 16 new bits are generated through the round function. Splitting the attack on the 64-bit key into the 4× 16-bit sub-keys used in the first four rounds reduces the complexity of guesses. In a brute force attack, the attacker would need, in the worst case, 2^{64} guesses before obtaining the correct key. Alternatively, in CPA attack, by observing the first clock cycle of encryption, an attacker only needs to guess at most 2^{16} key guesses and determine which one correlates best for that sub-key. The same procedure can be performed on the subsequent 3 cycles for a total of 4× 16-bit sub-keys (64-bit seed key). For a given 16-bit sub-key there are 65,536 key guesses, but the guess complexity can further be reduced by observing Hamming distance of smaller subsets (even 1-bit level sub-keys). As such, attacks could be performed on static CMOS based ciphers with only a few hundred key guesses and only ≈ 1,300 power traces [26].

When designs are unrolled into parallel processes, the signal-to-noise ratio (SNR) degrades, which increases attack complexity. This design employs the unrolling of the round function into a 32-bit parallel data path. Unrolling the design further would result in potentially more parallel structures to resist side-channel attacks while also increasing the hardware cost (power and area), making the implementation less applicable to lightweight devices. A random 32-bit plaintext is chosen for thousands of samples to develop a power profile and the list of plaintexts is used to create a Hamming distance profile of identical length to correlate against the power profiles.

For example, a 1-bit HD model at round 2 has a dependency on 4× round-1 key bits and its own round-2 key bit. For the 3-bit HD from X_{L1} to X_{L2}, the key dependency function is provided in Fig. 3.15 and has 256 possible keys

(excluding XOR dependencies of the second round). Given that the XOR operation is a conditional inversion, these round-2 XORs would only invert the value of the deduced key and result in a negative correlation coefficient. Bit $X_L3[0]$ is a function of rotated bit and XOR bit operations on the plaintext from the first round. Since the plaintext is known in this first round, so are its rotations, leaving only the XOR operation with the key bit as a variable. The $X_L3[2:0]$ primary dependencies for key guess are given by,

$$\begin{aligned}
X_{L3}[0] &= f(K_1[0], K_1[8], K_1[14], K_1[15], k_2[0]),\\
X_{L3}[1] &= f(K_1[0], K_1[1], K_1[9], K_1[15], k_2[1]),\\
X_{L3}[2] &= f(K_1[0], K_1[1], K_1[2], K_1[10], k_2[2]),\\
X_{L3}[2:0] &= f(K_1[0], K_1[1], K_1[2], K_1[8], K_1[9], K_1[10], K_1[14], K_1[15]).
\end{aligned} \tag{3.2}$$

For each bit added to the HD model at a round operation, the complexity increases exponentially making it infeasible to attack larger HDs or HDs at deeper intermediate states. There is a single fixed point in time for a single encryption power profile where correlation between the instantaneous power and Hamming distance is the greatest. Assuming an attacker has only limited information about the encryption algorithm/implementation, an attacker then would not know when this happens and therefore would need to correlate across a, potentially large, matrix of time values during that clock cycle or multiple cycles to find the greatest correlation. A set of voltage traces used to measure power during an encryption is illustrated in Fig. 3.16. There is a clear change in power behaviour of the circuit for its different operational modes. Note that the sampling resolution of the power signal should be sufficiently high to have enough samples per cycle, thereby detecting changes in power for the given plaintext.

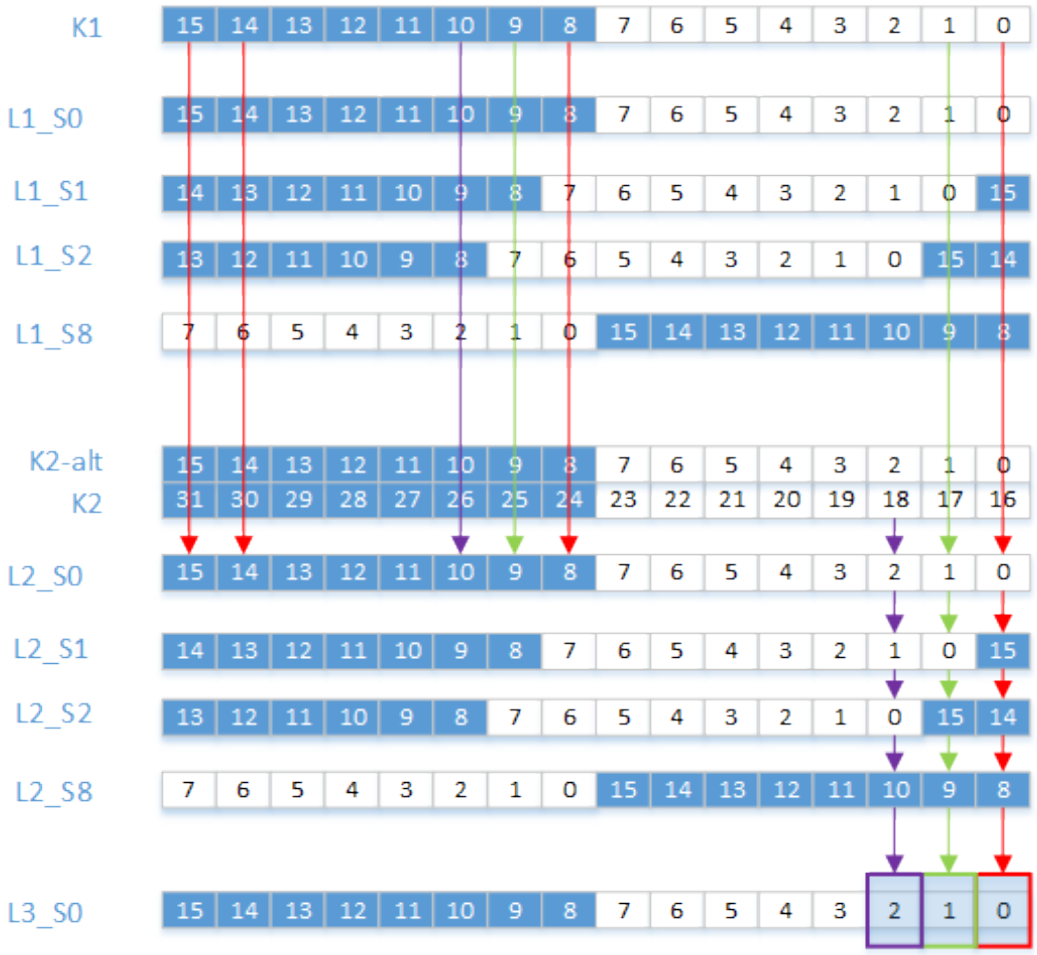

Figure 3.15: The key dependency function for the 3-bit HD from X_{L1} to X_{L2}

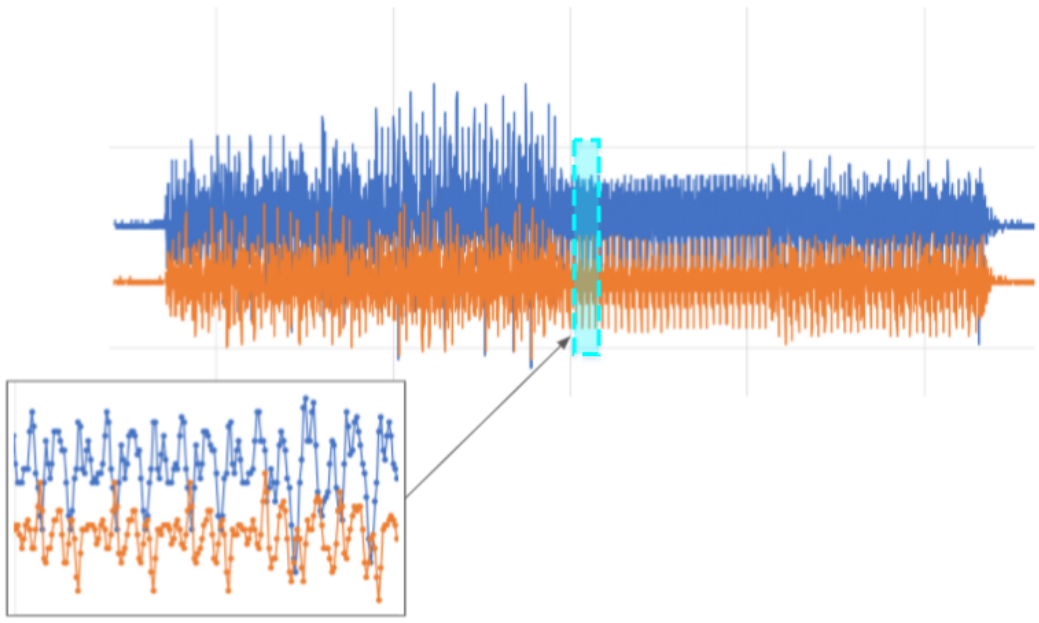

Figure 3.16: A set of voltage traces used to measure power during encryption, illustrating a clear change in power consumption behaviour of the circuit for its different operational modes

In this study, a series shunt resistor is added to measure the power consumption by measuring the voltage across and the current through this resistor. The resistance should be large enough to yield reasonable signal-to-noise ratio to accurately measure power consumption, but at the same time, it should maintain a reasonable operating voltage for the chip. For this system, a 10kΩ resistor was used. Using a 10kΩ shunt resistor, it is expected that there will be 10mV of drop for every 1μA of current drawn, causing a peak-to-peak supply noise V_{pp} of 600mV. Based on post-layout simulations, the expected location for the greatest correlation is within the second half of the rising edge of a clock signal. An example of a power trace captured using this method is shown in Fig. 3.16.

In this study, up to 80K plaintexts were used for the correlation power analysis attack described above, but the sub-keys of the SIMON cipher could not be recovered based on the correlations obtained. This behavior could be explained by the following: (1) bypass (decoupling) capacitors have a non-negligible impact on the load current, (2) supply rail noise affects the measurements, (3) the sampling frequency of the scope is not sufficient, (4) the impact of parasitic impedances are non-negligible when measuring instantaneous power, (5) larger number of input traces is required due to the proposed round-parallel architecture. Related to item 3 above, Fig. 3.17 demonstrates that the difference between the switching time of the targeted X_{LR} registers and other peripheral circuits such as the key generation and control unit are on the order of hundreds of picoseconds. Thus, a higher sampling rate would be needed to distinguish the peak current drawn by the target signal from other peaks drawn by the peripheral circuits.

Potential future work to facilitate a successful attack includes reducing bypass capacitors on the supply rail while maintaining functionality and lowering

the clock frequency to reduce the amount of current provided by the bypass capacitors. Furthermore, a redesign of the PCB could reduce overall wirelength (and related parasitic impedances) in the system. Operating the system at 5V would increase current and potentially make the correlation more apparent by reducing ripple at the output of the LDO. Finally, more plaintexts could be needed to establish the correlation and retrieve the encryption keys.

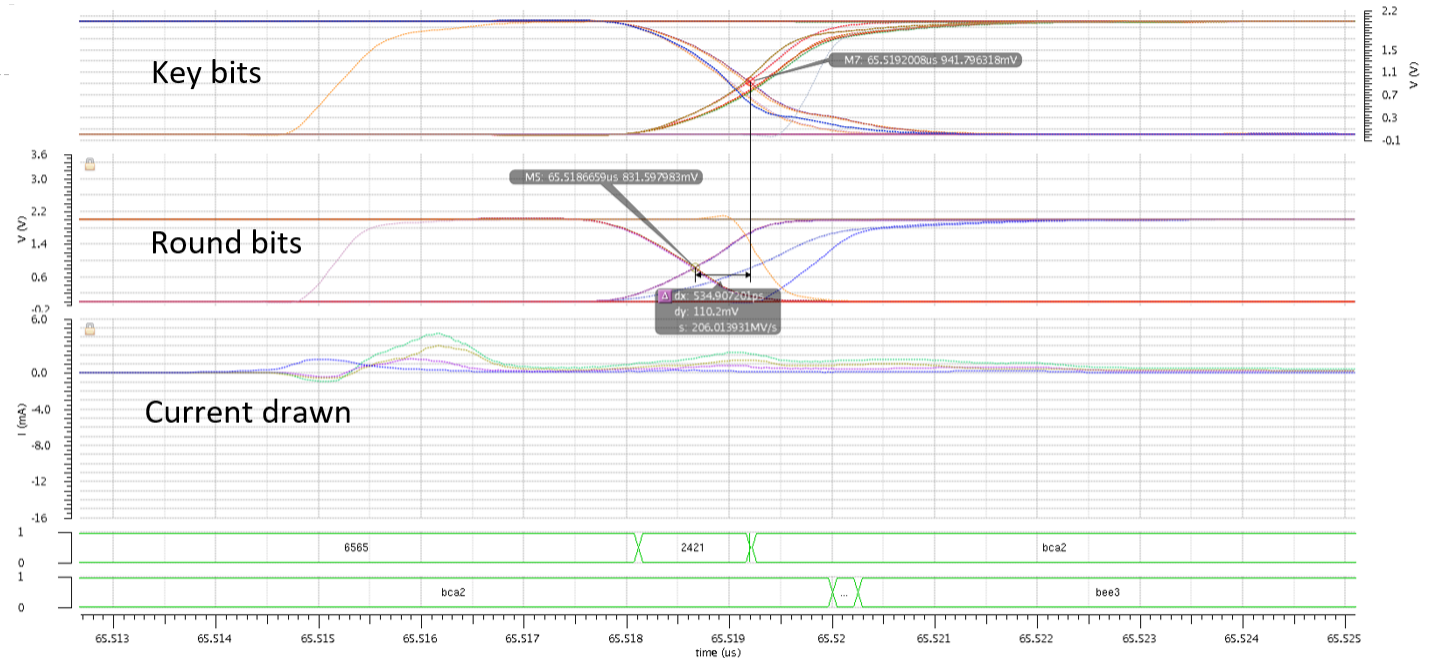

Figure 3.17: Data switching activity at the clock edge of first partial encryption, illustrating that the difference in the switching time of the targeted X_{LR} registers and other peripheral circuits such as the key generation and control unit is on the order of hundreds of picoseconds

Chapter 4

Comparative Analysis of S-Box Implementations on ASIC and FPGA

A critical step in AES hardware is the design and implementation of S-Box. The S-Box is a non-linear manipulation and mapping of values in a Galois field. A Galois field or finite-field is a field that contains a finite number of elements in the notation $GF(p^n)$ where p is a prime number and n is the size of the polynomial. More specifically, in AES, the field of 2^8 and irreducible polynomial $x^8 + x^4 + x^3 + x + 1$ are used. This step alone can consume as much as 80% of the encryption core's area and is a heavily used block for the algorithm [17]. LUT based approaches using a pre-stored memory and multiplexing logic face difficulty in pipelining since they have a limited number of stages that can be added for a performance speedup [29]. Furthermore, concerns of side-channel leakage in typical implementation using LUTs in this step have made it an active area of research for alternative techniques [15].

Galois field arithmetic (GFA) is a method that is used to compute the inverse in the field rather than using aforementioned LUTs. A common issue with in-field arithmetic is the computational complexity and efficiency. Architectures that can do these operations with little compromise in performance and greater side-channel security are highly desirable. In this thesis, implementations of Fermat's Little Theorem are compared to typical LUT implementations in both FPGA and ASIC platforms.

4.1 Finding Multiplicative Inverses in Galois Fields

There are several algorithms to find the multiplicative inverse of a number in a Galois field. The most popular one is a 256-element LUT, desired for its small footprint and operating speed [21]. A pipelined 256-1 LUT that completes operation in two clock cycles, as shown in Fig. 4.1, is used in this work as a reference to compare other approaches. A 256-to-1 MUX can be used to implement LUT, but it is typically not a preferred method because of its sensitivity to side-channel attacks based on select inputs [21]. Various methods exist to mask select inputs to enhance side-channel resistance at the expense of additional hardware overhead [21, 30, 31, 32].

Fermat's Little Theorem (FLT) in a 2^8 finite-field states that the inverse of a number can be computed using the following [33],

$$\begin{aligned} p^{N-2} &= p^{-1}, \\ x^{256-2} &= x^{-1}, \end{aligned} \tag{4.1}$$

where in the case of Rijndael's S-Box used in AES, $N = 256$ and x is the input

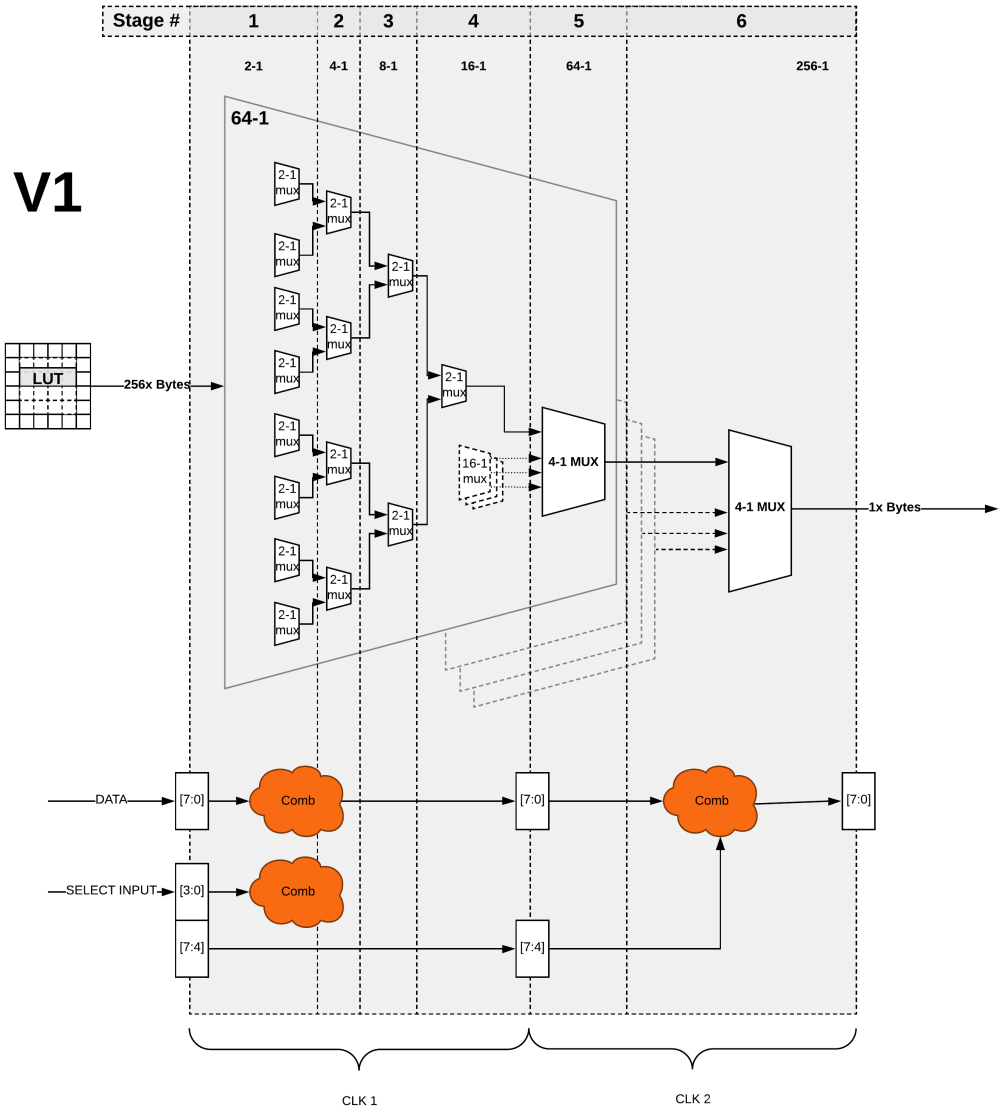

Figure 4.1: Pipelined AES 256-1 MUX based LUT

that will be substituted [34]. By computing x^{254}, the multiplicative inverse can be found for any given input x. FLT implementation is simple since the design only has multipliers and these multipliers are only XOR operations when operating in a finite field. Multiplication and modulus operations are shown in Fig. 4.2. The S-Box substitution with both multiplicative inverse and affine transformations is provided by,

$$\begin{aligned} &Y = \mathbf{Inv}(\mathbf{X}), \\ &Y = X^{254}, \\ &Y = ((((((X^2X)^2)^2X^3)^2)^2)^2x^6x^1)^2, \\ &Result = Y \oplus [Y << 1 \oplus Y << 2 \oplus Y << 3 \oplus Y << 4 \oplus 0x63]. \end{aligned} \tag{4.2}$$

This relative simplicity provides an opportunity to fold and unfold the multiplication operations at various stages to optimize power, area, and efficiency. Specifically, each multiplication can be folded or unfolded from bit-serial to full encryption level operation depending upon the application. Using different multiplication blocks, a design can be extended to support much higher speedups in pipelining. For example, Karatsuba multipliers operate the same in Galois fields as they do with the set of natural numbers, and are used to reduce gate complexity of each multiplier by half while maintaining the same gate-depth [35]. A breakdown of how a Karatsuba multiplier is used in these designs is shown in Fig. 4.3. The original "long multiplication" used in Fig. 4.2 uses 136× XOR and AND gates while the Karatsuba implementation uses 70× gates, a reduction of almost 2×.

Figure 4.2: Traditional "long" multiplier and modulus operation for Galois field

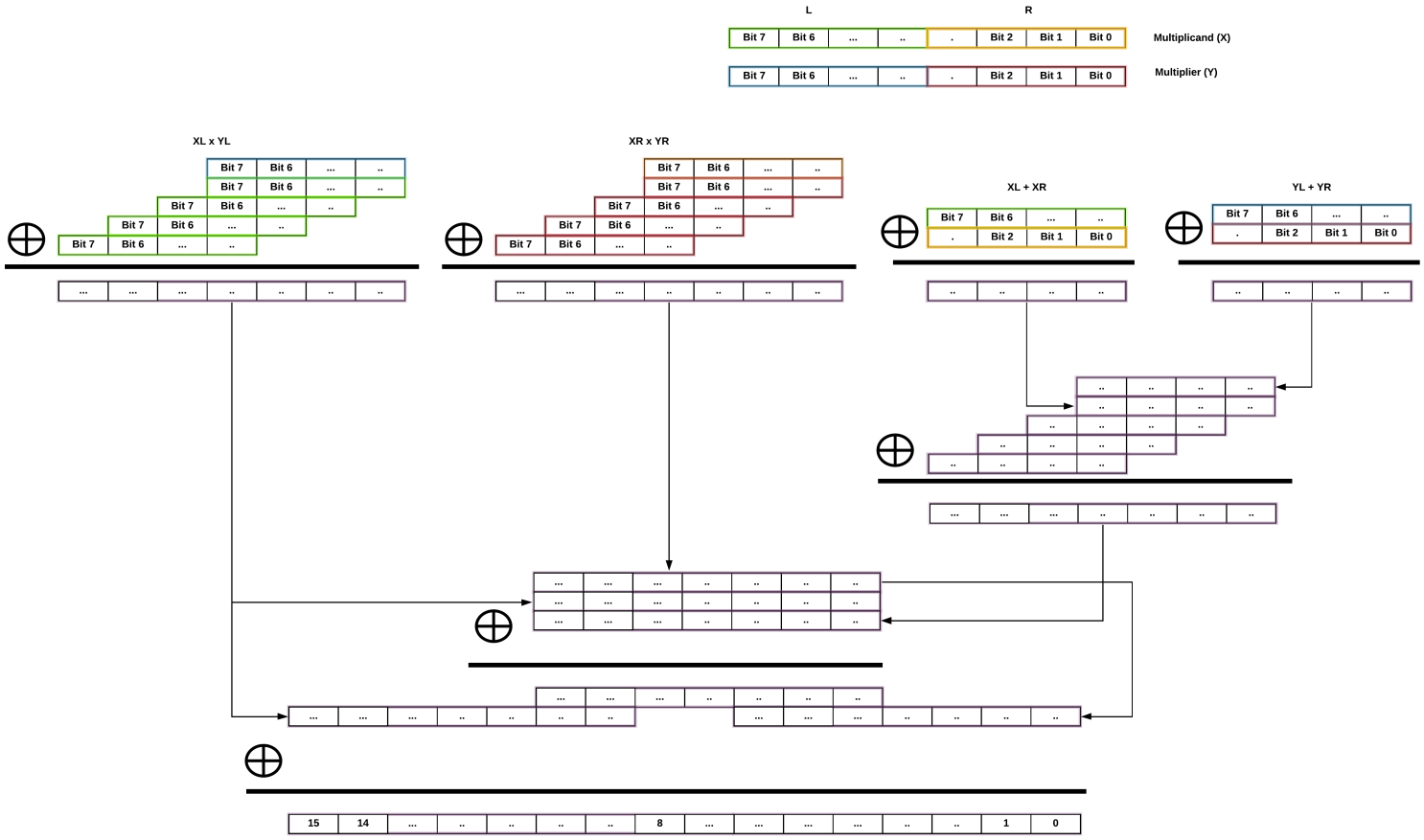

Figure 4.3: Karatsuba multiplier

Arch. Name	Description
LUT V1	Single-cycle LUT
LUT V2	2-stage pipelined LUT
FLT V1	Single-cycle FLT
FLT V2	12-stage pipelined FLT
FLT V3	11-stage increased parallelism FLT

Table 4.1: Descriptions and labels of the 5 S-Box implementation approaches that were evaluated in this work

The extended Euclidean algorithm (EEA) obtains the multiplicative inverse by computing the greatest common divisor [33]. The algorithm however is more complex in branching condition, thereby complicating expected branch prediction if a pipelined implementation is developed. Furthermore, it has a non-constant run time [20], making it susceptible to relatively simple timing-based side-channel attacks. Longer or shorter run time of the algorithm leaks what subset of inputs the plaintext was a part of, thereby simplifying key guess complexity. Extending run times of all inputs to match the worst run time input (to reduce the information leakage) would yield a greater number of rounds needed to compute the inverse than algorithms such as FLT.

To maximize efficiency, a 2^8 finite field FLT was chosen in this work since EEA has non-constant run time exposing itself to timing attacks and a likely larger worst case encryption latency due to its complexity. Two versions of LUT were assessed: a single cycle 256-1 MUX and a 2-stage 256-1 MUX. Three variations of FLT were assessed: A single-cycle 12× multiply data path, a 12-stage pipelined version and a more paralleled 11-stage pipelined version, as illustrated in Fig. 4.4. All of the five designs were assessed in both FPGA and ASIC platforms. These designs are summarized (and labeled) in Table 4.1.

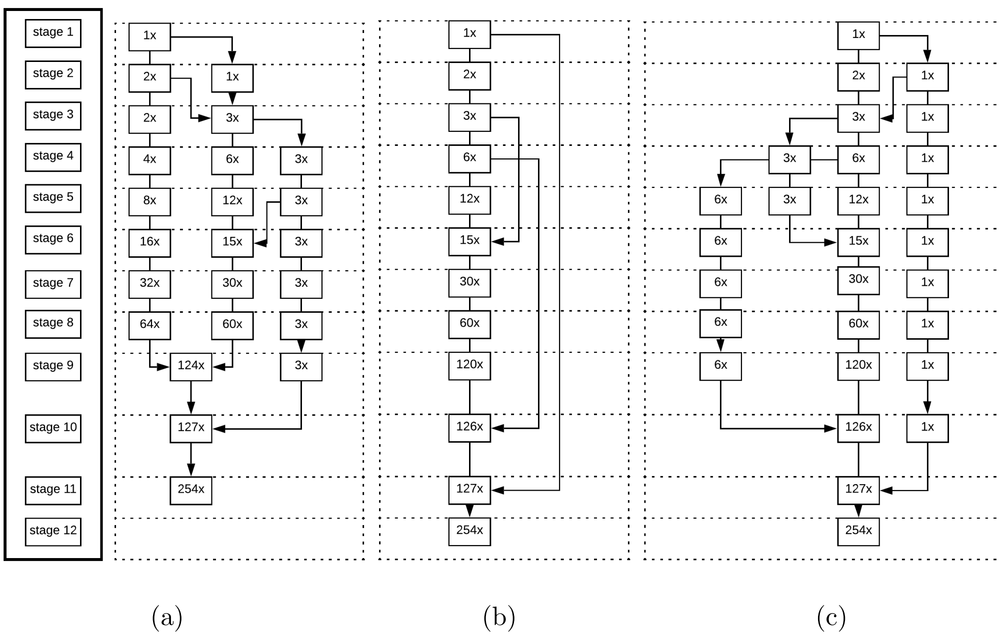

Figure 4.4: Three variations of FLT that were evaluated in this work: (a) 11-stage pipeline-parallel, (b) single-cycle, and (c) 12-stage pipeline

4.2 ASIC Synthesis Results

All of the five design approaches were implemented in System Verilog and synthesized at the gate level using Nangate Open Cell Library in a 45nm technology node [36]. Correct functionality for each design was demonstrated. In terms of performance, the FLT and LUT single cycle are comparable in area, power and delay, but have limited maximum operating frequency. Pipelined variations of FLT achieve higher operating frequencies than the LUT (see Fig. 4.5), but suffer from latency and increased dynamic and static power dissipation due to the number of switching elements and total number of transistors (see Fig. 4.6). Largest power is consumed by memory elements such as flip-flops, increasing substantially with each of its many pipeline stages (see Fig. 4.7). The power-delay product (see Fig. 4.8) shows that FLT implementations consume much higher power and are slower than their LUT counterparts.

For high performance applications that can tolerate larger power and area, pipelined FLT circuits offer 30% higher maximum operating speed and throughput at the expense of 12 clocks of latency. In low power applications, the single-cycle FLT offers a 7.8% smaller circuit footprint (see Fig. 4.9) and a comparable power efficiency. The FLT variations use much higher number of combinational cells. Furthermore, pipelining substantially increases the number of sequential cells (see Fig. 4.10). The number of ports for each design (see Fig. 4.11) is greater for the FLT-V2, FLT-V3 and LUT-V2 pipelined designs. At the top level, FLT based designs use a larger number of nets and total cells compared to the LUT implementations (see Fig. 4.12).

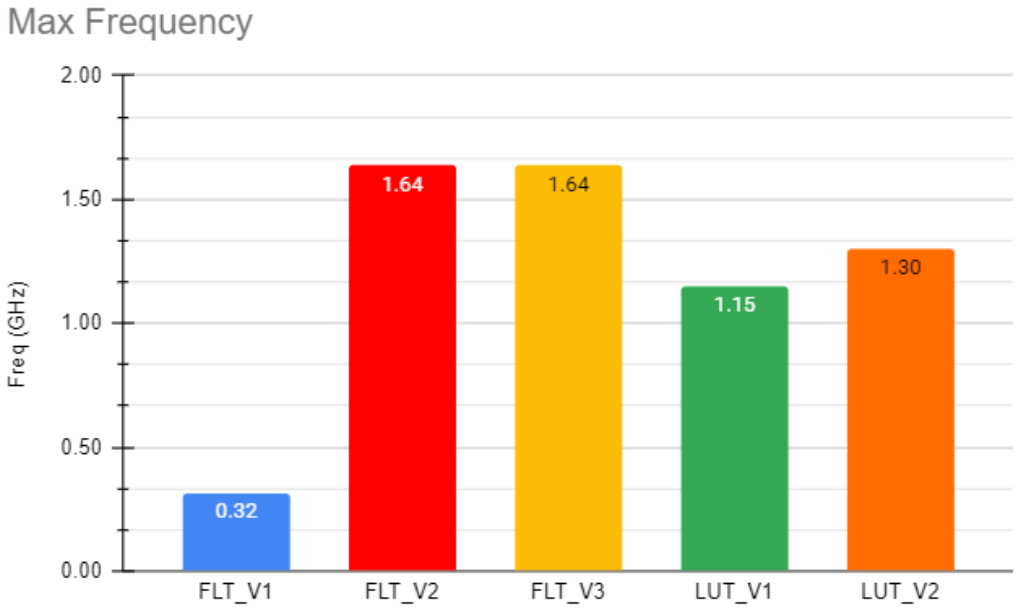

Figure 4.5: Maximum operating frequency for different approaches summarized in Table 4.1 for ASIC implementation

4.3 FPGA Results

Using the same System Verilog code described above for ASIC, a programmable logic layer is implemented on a Zync 7Z007S SoC MiniZED development board and controlled by an ARM core CPU layer [37]. The programmable logic layer utilizes an AXI interface to receive a batch of data and compute the inverse. For this design, latency is defined by the size of the block of data.

Limitations related to the FPGA prevented practical maximum operating frequency measurements since critical paths are located within the AXI-interface of the design, as shown in Fig. 4.13. Results listed in Table 4.2 and shown in Fig. 4.14 demonstrate that the single cycle FLT-V1 has a comparable size to the single cycle LUT. Pipelined variations of the designs exhibit similar results as the ASIC implementation. For example, the FLT implementation

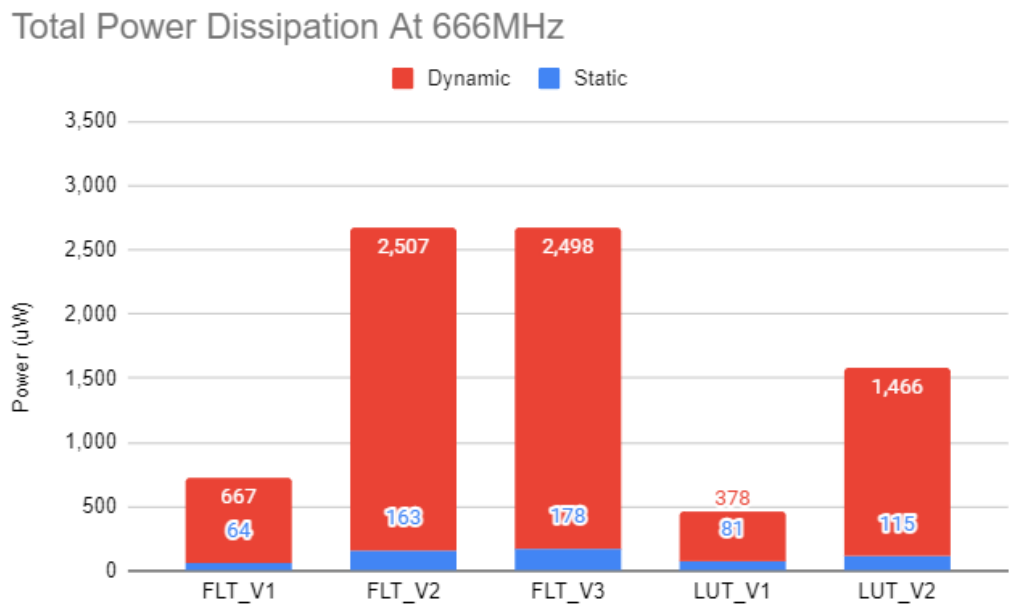

Figure 4.6: Power breakdown for different approaches summarized in Table 4.1 for ASIC implementation

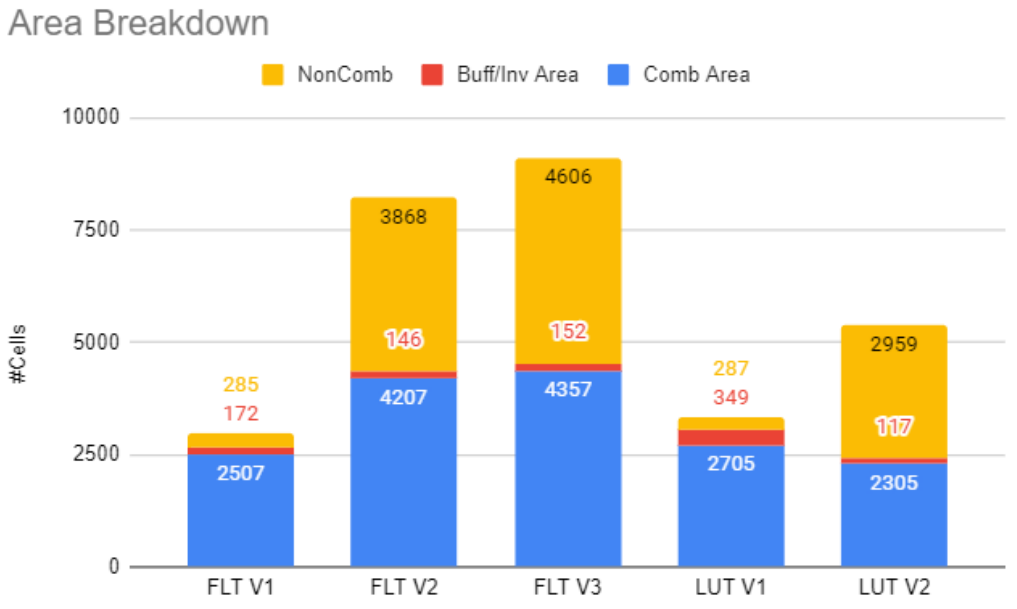

Figure 4.7: Area breakdown for different approaches summarized in Table 4.1 for ASIC implementation

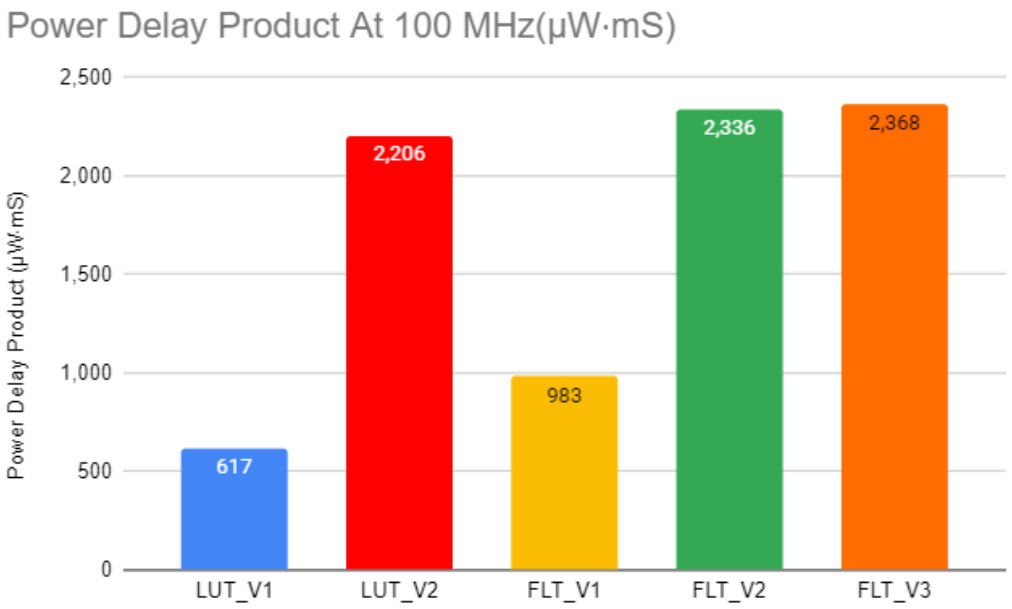

Figure 4.8: Power-delay product results for different approaches summarized in Table 4.1 for ASIC implementation

consumes more area and operates at higher clock speeds. When encrypting a smaller payload of 256 bytes (see Fig. 4.15) and larger payload of 4,096 bytes (see Fig. 4.16), the FLT variations have higher throughput at the expense of a larger area.

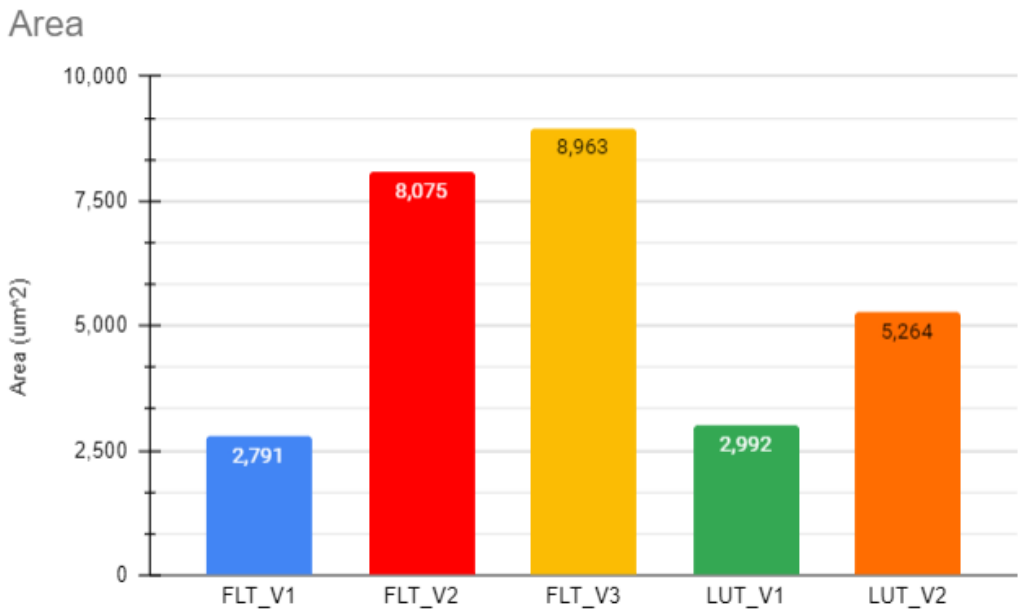

Figure 4.9: Area breakdown for different approaches summarized in Table 4.1 for ASIC implementation

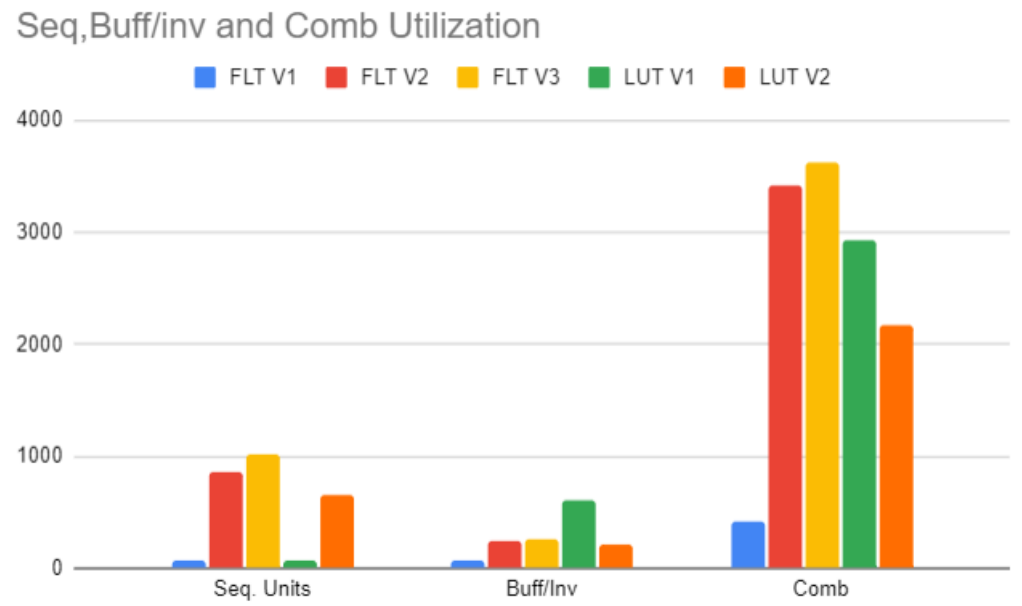

Figure 4.10: Sequential, buffer and combinational cell utilization for different approaches summarized in Table 4.1 for ASIC implementation

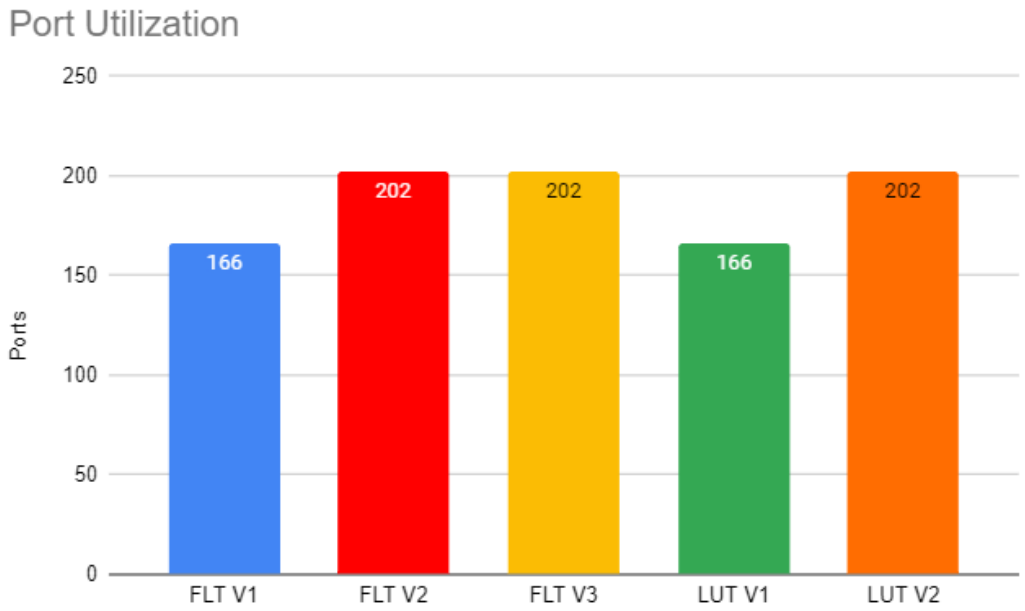

Figure 4.11: Port utilization for different approaches summarized in Table 4.1 for ASIC implementation

	LUT V1	**LUT V2**	**FLT V1**	**FLT V2**	**FLT V3**
Latency (Clks)	1	2	1	11	12
Encrypt 4096(μs)	12.96	7.36	45.80	7.08	7.26
Encrypt 256 (μs)	1.75	1.78	6.23	1.69	1.75
Resource Usage (Cells used)					
BRAM	4	8	4	8	8
FPGA-LUT	39.15	46.98	44.22	49.68	49.56
LUTRAM	13.7	15.87	13.70	16.42	16.95
FF	19.80	25.11	19.80	25.44	25.11
Max Freq (MHz)	172.00	175.00	48.48	177.78	177.78
Slack (ns)	0.01	0.08	0.35	0.14	0.02

Table 4.2: Latency and resource utilization for different approaches summarized in Table 4.1 for FPGA implementation

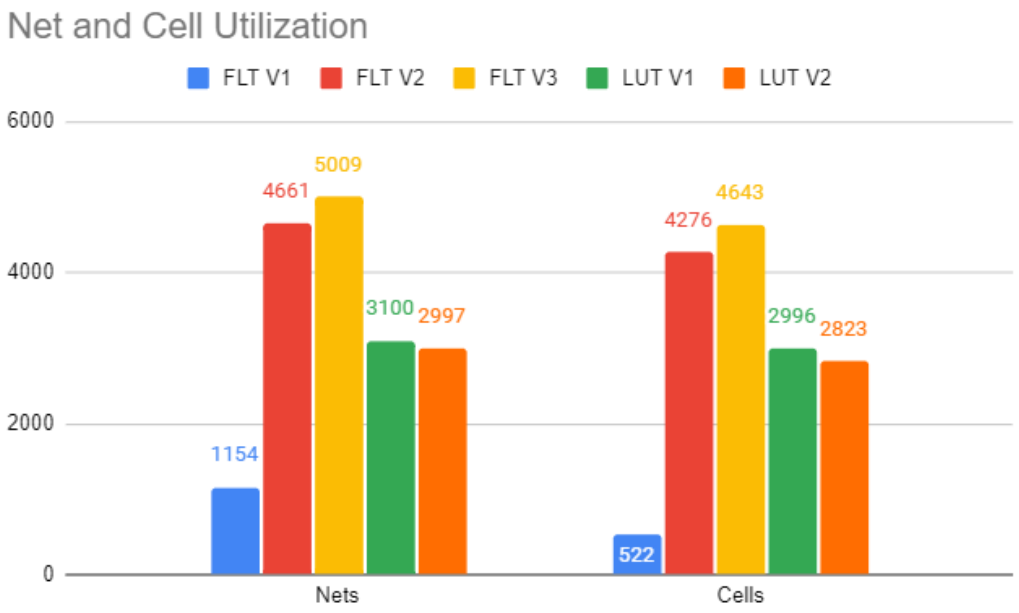

Figure 4.12: Net and cell utilization for different approaches summarized in Table 4.1 for ASIC implementation

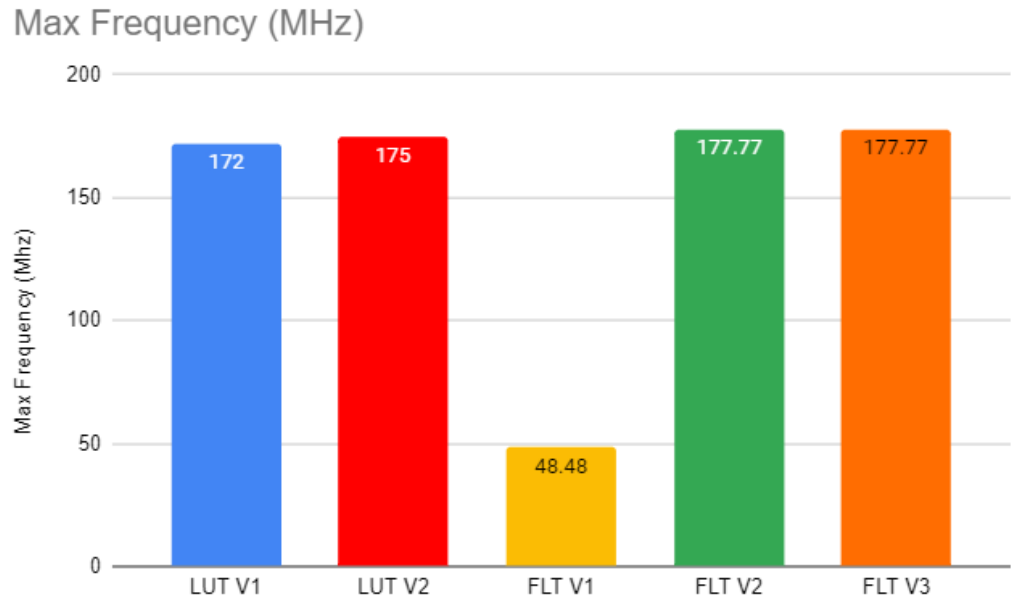

Figure 4.13: Maximum frequencies for different approaches summarized in Table 4.1 for FPGA implementation

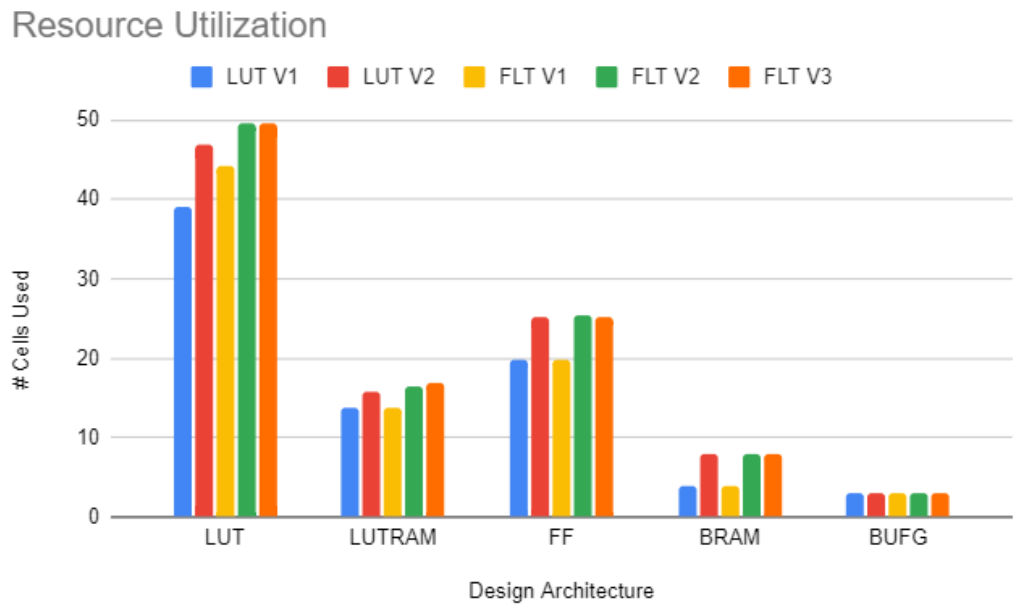

Figure 4.14: Resource utilization for different approaches summarized in Table 4.1 for FPGA implementation

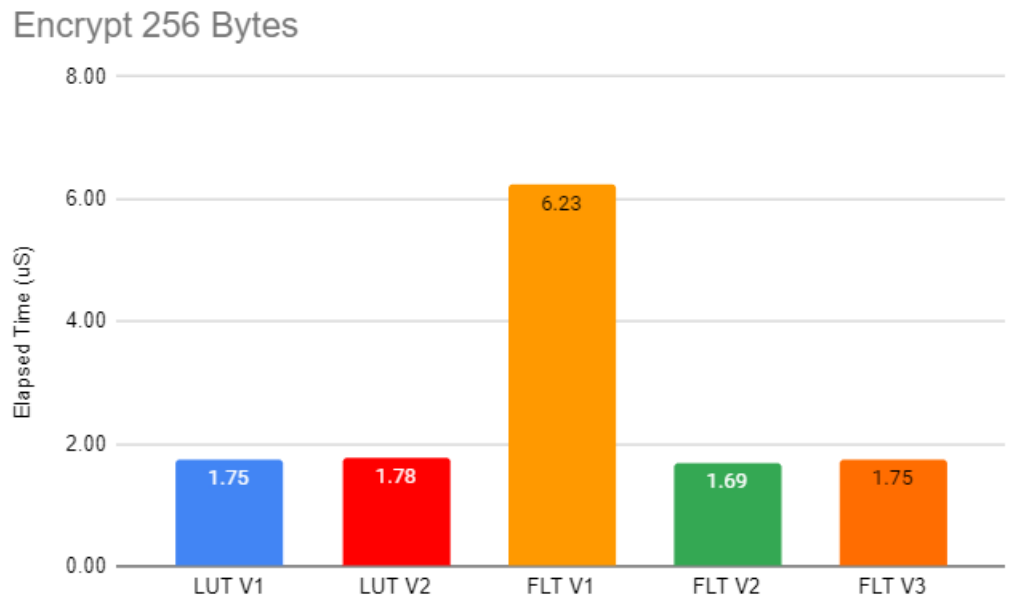

Figure 4.15: Latency to encrypt 256 bytes for different approaches summarized in Table 4.1 for FPGA implementation

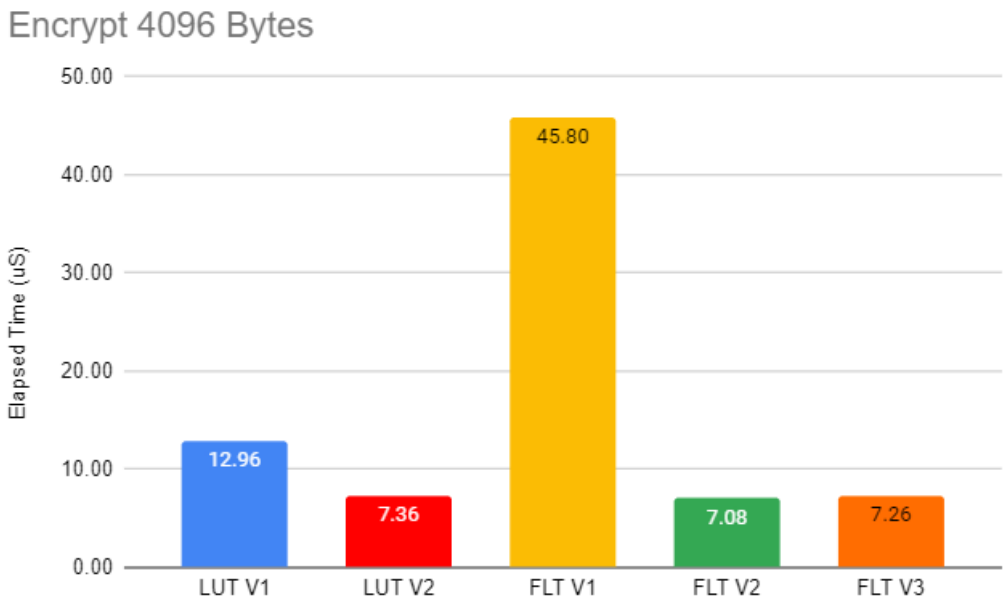

Figure 4.16: Latency to encrypt 4,096 bytes for different approaches summarized in Table 4.1 for FPGA implementation

Chapter 5

Conclusions

A lightweight SIMON cipher was developed for resource-constrained applications. The cipher exhibited strong resistance against early attempts of power-side channel attacks. The system uses a round-parallel architecture to accelerate computation while maintaining the same number of memory elements as a bit-serial implementation. All of the simulations were performed using a 3.3V supply voltage at the 0.5μm ON Semiconductor AMI technology node for a frequency of 13.56MHz (RFID HF). Average power dissipation during an encryption is 1.03mW. The design was also fabricated and test results were obtained. The experimental results sufficiently match simulation results where the measured encryption efficiency is 3.29kbits/sec/μW.

In the second part of the thesis, a Rijndael S-Box, a primary building block of AES hardware, was implemented on both FPGA and ASIC using Synopsys synthesis tools. The primary objective was to compare hardware performance of alternative means of computing the S-Box operation in the AES algorithm. Fermat's Little Theorem (FLT) based implementations consumed similar power as LUT implementations in low frequency applications with a

7.2% improvement in area cost. The high performance versions of FLT S-Box achieved 30% higher maximum frequencies as compared to a simple 256-1 MUX lookup table.

5.1 Future Scope

At the design-level, certain techniques can enhance performance such as the usage of falling edge triggered flip-flops to mitigate switching noise. Decoupling capacitors internal to the die can mitigate power supply fluctuations at the higher frequencies. At the embedded PCB level, the design can be shrunk to further reduce parasitic impedances. These techniques would enhance overall signal integrity.

Power based side-channel attack methodology can be improved with higher sampling rates, reduced supply rail noise, and amplification of the small voltage drops that indicate instantaneous power during clock edges.

AES circuits can use FLT to enhance efficiency in both high performance and low power implementations of AES. An extension to these results could be the analysis of FLT at different levels of folding, from bit-serialized to fully parallel circuits of the multiplier cell and modulus cell system. Increased amount of pipelining can also allow higher frequencies of operation and greater speedup.

Bibliography

[1] *Protecting Against Side-Channel Attacks with an Ultra-Low Power Processor*. URL: https://www.synopsys.com/designware-ip/technical-bulletin/protecting-against-side-channel.html.

[2] Ray Beaulieu et al. "The SIMON and SPECK lightweight block ciphers". In: (June 2015), 1–13, 21–27.

[3] Tutu Wan, Yasha Karimi, Milutin Stanacevic, and Emre Salman. "Perspective Paper—Can AC Computing Be an Alternative for Wirelessly Powered IoT Devices?" In: *IEEE Embedded Systems Letters* 9.1 (2017), 13–16. DOI: 10.1109/les.2017.2653058.

[4] Tutu Wan, Yasha Karimi, Milutin Stanacevic, and Emre Salman. "Energy efficient AC computing methodology for wirelessly powered IoT devices". In: *Proceedings of the IEEE International Symposium on Circuits and Systems* (May 2017), 509–512.

[5] Tutu Wan, Yasha Karimi, Milutin Stanacevic, and Emre Salman. "Energy efficient AC computing methodology for wirelessly powered IoT devices". In: *Government Microcircuit Applications & Critical Technology Conference* (May 2018), 939–944.

[6] Tutu Wan, Yasha Karimi, Milutin Stanacevic, and Emre Salman. "AC Computing Methodology for RF-Powered IoT Devices". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 27.5 (Apr. 2019).

[7] Tutu Wan, Emre Salman, and Milutin Stanacevic. "A New Circuit Design Framework for IoT Devices: Charge Recycling with Wireless Power

Harvesting". In: *Proceedings of the IEEE International Symposium on Circuits and Systems* (May 2016), 2046–2049.

[8] Tutu Wan and Emre Salman. "Ultra Low Power SIMON Core for Lightweight Encryption". In: *2018 IEEE International Symposium on Circuits and Systems (ISCAS)* (2018). DOI: 10.1109/iscas.2018.8351163.

[9] Ivan Miketic and Emre Salman. "Power and Data Integrity in Monolithic 3D Integrated SIMON Core". In: *2019 IEEE International Symposium on Circuits and Systems (ISCAS)* (2019). DOI: 10.1109/iscas.2019.8702438.

[10] Ivan Miketic and Emre Salman. "Power and Data Integrity in Monolithic 3D Integrated SIMON Core". In: *Government Microcircuit Applications & Critical Technology Conference* (Mar. 2019). DOI: 10.1109/iscas.2019.8702438.

[11] C. Yan and E. Salman. "Mono3D: Open Source Cell Library for Monolithic 3-D Integrated Circuits". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 65.3 (2018), pp. 1075–1085.

[12] C. Yan, S. Kontak, H. Wang, and E. Salman. "Open source cell library Mono3D to develop large-scale monolithic 3D integrated circuits". In: *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2017, pp. 1–4.

[13] Hailang Wang, Mohammad H. Asgari, and Emre Salman. "Compact model to efficiently characterize TSV-to-transistor noise coupling in 3D ICs". In: *Integration* 47.3 (2014). Special issue: VLSI for the new era, pp. 296 –306.

[14] Emily Gray-Fow. *A Brief Peek Into the Fascinating World of Side Channel Attacks*. July 2019. URL: https://medium.com/swlh/a-brief-peek-into-the-fascinating-world-of-side-channel-attacks-809f96eabea1.

[15] C Ashokkumar, Bholanath Roy, M Bhargav Sri Venkatesh, and Bernard L Menezes. ""S-Box" Implementation of AES is NOT side channel resistant". In: (May 2020).

[16] Joan Daemen and Vincent Rijmen. *The design of Rijndael: AES — the Advanced Encryption Standard*. Springer-Verlag, 2002, p. 238. ISBN: 3-540-42580-2.

[17] Frank Gürkaynak, Stephan Oetiker, Hubert Kaeslin, Norbert Felber, and Wolfgang Fichtner. "Improving DPA Security by Using Globally-Asynchronous Locally-Synchronous Systems". In: *Proceedings of ESSCIRC* (2005).

[18] Ali Akbar PAmmu, Kwen-Siong Chong, Kyaw Zwa Lwin Ne, and Bah-Hwee Gwee. "High Secured Low Power Multiplexer-LUT Based AES S-Box Implementation". In: (2016).

[19] M M Wong, M L.D. Wong, A K Nandi, and I I Hijazin. "AES S-box using Fermat's Little Theorem for the highly constrained embedded devices". In: *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)* (Aug. 2012).

[20] Yeong Chee Mei and Siti Zarina Md Naziri. "The FPGA Implementation of Multiplicative Inverse Value of GF(28) Generator using Extended Euclid Algorithm (EEA) Method for Advanced Encryption Standard (AES) Algorithm". In: (2011).

[21] Ali Akbar Pammu, Kwen-Siong Chong, and Bah-Hwee Gwee. "Secured Low Power Overhead Compensator Look-Up-Table (LUT) Substitution Box (S-Box) Architecture". In: *2016 IEEE International Conference on Networking, Architecture and Storage (NAS)* (2016). DOI: 10.1109/nas.2016.7549420.

[22] Nikhil Chawla, Arvind Singh, Monodeep Kar, and Saibal Mukhopadhyay. "Extracting Side-Channel Leakage from Round Unrolled Implementations of Lightweight Ciphers". In: (2019).

[23] Wojciech Wodo and Lucjan Hanzlik. "Thermal Imaging Attacks on Keypad Security Systems". In: *Proceedings of the 13th International Joint Conference on e-Business and Telecommunications* (2016). DOI: 10.5220/0005998404580464.

[24] Hagai Bar-El. "Introduction to Side Channel Attacks". In: ().

[25] Steven S. Skiena. *Algorithm Design Manual*. 2nd ed. Springer International Publishing AG, 2008, p. 38.

[26] Arvind Singh, Nikhil Chawla, Jong Hwan ko, Monodeep Kar, and Sailbal Mukhopadhyay. "Energy Efficient and Side-Channel Secure Cryptographic Hardware for IoT-Edge Nodes". In: (2018).

[27] Shivam Bhasin, Tarik Graba, Jean-Luc Danger, and Zakaria Najm. "A look into SIMON from a side-channel perspective". In: *2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)* (2014). DOI: 10.1109/hst.2014.6855568.

[28] Aria Shahverdi, Mostafa Taha, and Thomas Eisenbarth. "Lightweight Side Channel Resistance: Threshold Implementations of SIMON". In: *IEEE Transactions on Computers* 66.4 (Apr. 2017).

[29] Xinmiao Zhang and K.k. Parhi. "High-speed VLSI architectures for the AES algorithm". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 12.9 (2004), 957–967. DOI: 10.1109/tvlsi.2004.832943.

[30] Kean Hong Boey, Philip Hodgers, Yingxi Lu, Maire Oneill, and Roger Woods. "Security of AES Sbox designs to power analysis". In: *2010 17th IEEE International Conference on Electronics, Circuits and Systems* (2010). DOI: 10.1109/icecs.2010.5724741.

[31] Cheng-Hua Duan, Jun Jiang, Xing-Ming Wang, and Wen-Yuan Xu. "Fast S-Box Substitution Instructions and Their Hardware Implementation for Accelerating Symmetric Cryptographic Processing". In: *2009 International Conference on E-Business and Information System Security* (2009). DOI: 10.1109/ebiss.2009.5137980.

[32] Craig Teegarden, Mudit Bhargava, and Ken MAi. "Side-Channel Attack Resistant ROM-Based AES S-Box". In: (2010).

[33] David M. Burton. *Elementary number theory*. McGraw-Hill Education, 2016, pp. 26–31,87–92.

[34] Sumio Morioka and Akashi Satoh. "An Optimized S-Box Circuit Architecture for Low Power AES Design". In: *Cryptographic Hardware and Embedded Systems - CHES 2002 Lecture Notes in Computer Science* (2003), 172–186. DOI: 10.1007/3-540-36400-5_14.

[35] F Rodriguez-Henriquez and C K Koc. "On fully parallel Karatsuba Multipliers for GF(2^m)". In: (2002).

[36] *Open-Cell Library: Silicon Integration Initiative*. Aug. 2019. URL: https://si2.org/open-cell-library/.

[37] *MiniZed*. URL: http://zedboard.org/product/minized.

[38] Aydin Aysu, Ege Gulcan, and Patrick Schaumont. "SIMON Says: Break Area Records of Block Ciphers on FPGAs". In: *IEEE EMbedded Systems Letters* 6.2 (June 2014).

[39] Joan Boyar and René Peralta. "A new combinational logic minimization technique with applications to cryptology." In: (2010).

[40] D. Canright. "A Very Compact Rijndael S-box". In: (2005). DOI: 10.21236/ada434781.

[41] William Diehl, Abubakr Abdulgadir, Jens-Peter Kaps, and Kris Gah. "Comparing the Cost of Protecting Selected Lightweight Block Ciphers Against Differential Power Analysis in Low-Cost FPGAs". In: (Apr. 2018), 1–2,6–8,10–26.

[42] Hasindu Gamaarachchi, Harsha Ganegoda, and Roshan Ragel. "Breaking Speck cryptosystem using correlation power analysis attack". In: *Journal of the National Science Foundation of Sri Lanka* (July 2017).

[43] Hasindu Gamaarachchi and Harsha Ganegoda. "Power Analysis Based Side Channel Attack". In: (Jan. 2018), 1–29,50–65.

[44] Gilbert Goodwil, Benjamin Jun, Josh Jaffe, and Pankag Rohatgi. "A testing methodology for sidechannel resistance validation". In: (2011).

[45] Ege Gulcan, Aydin Aysu, and Patrick Schaumont. "A Flexible and Compact Hardware Architecture for the SIMON Block Cipher". In: *International Workshop on Lightweight Cryptography for Security and Privacy* (Mar. 2015).

[46] David Harris. *CMOS VLSI design: a circuits and systems perspective*. Pearson/Addison Wesley, 2004.

[47] Ravi Kishore Kodali, Chandana N. Amanchi, Shubham Kumar, and Lakshmi Boppana. "FPGA implementation of Itoh-Tsujii inversion algorithm". In: *International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014)* (2014). DOI: 10.1109/icraie.2014.6909308.

[48] Abhishek Kumar and Sokat Tejani. "S-BOX Architecture". In: *Futuristic Trends in Network and Communication Technologies*. Springer, 2018, 17–27.

[49] Abhishek Kumar and Sokat Tejani. "S-BOX Architecture". In: (2019).

[50] Emilia Käsper and Peter Schwabe. "Faster and Timing-Attack Resistant AES-GCM". In: ().

[51] Ahmad Sghaier Omar and Otman Basir. "SIMON 32/64 and 64/128 Block Cipher: Study of Cross-Correlation and Linear Span Attack Immunity". In: (2017).

[52] Ali Akbar Pammu, Kwen-Siong Chong, and Bah-Hwee Gwee. "Highly secured arithmetic hiding based S-Box on AES-128 implementation". In: *2016 International Symposium on Integrated Circuits (ISIC)* (2016). DOI: 10.1109/isicir.2016.7829736.

[53] Taewan Park, Hwajeong Seo, and Howon Kim. "Parallel Implementations of SIMON and SPECK". In: (2016).

[54] L Parrilla, A Lloris, E Castillo, and A Garcia. "Minimum-clock-cycle Itoh-Tsujii algorithm hardware implementation for cryptography applications over GF(2m) fields". In: *Electronics Letters* (Aug. 2012).

[55] Aditya Pradeep, Vishal Mohanty, Adarsh Muthuveeru Subramaniam, and Chester Rebeiro. "Revisiting AES SBox Composite Field Implementations for FPGAs". In: *IEEE Embedded Systems Letters* 11.3 (2019), 85–88. DOI: 10.1109/les.2019.2899113.

[56] Morteza Safaeipour and Mahmoud Salmasizadeh. "A new CPA resistant software implementation for symmetric ciphers with smoothed power consumption". In: *2016 13th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC)* (Sept. 2016). DOI: 10.1109/iscisc.2016.7736449.

[57] Emre Salman and Eby G. Friedman. *High performance integrated circuit design*. McGraw-Hill, 2012.

[58] Arvind Singh, Nikhil Chawla, Monodeep Kar, and Saibal Mukhopadhyay. "Energy Efficient and Side-Channel Secure Hardware Architecture for Lightweight Cipher SIMON". In: (2018).

[59] Arvind Singh et al. "Enhanced Power and Electromagnetic SCA Resistance of Encryption Engines via a Security-Aware Integrated All-Digital LDO". In: (2019).

[60] Jos Wetzels and Wouter Bokslag. "Simple SIMON FPGA implementations of the / Block Cipher". In: (2015).