Fall 2012

Null Convention Logic applications of asynchronous design in nanotechnology and cryptographic security

Jun Wu

Department: Electrical and Computer Engineering

NULL CONVENTION LOGIC APPLICATIONS OF ASYNCHRONOUS DESIGN IN NANOTECHNOLOGY AND CRYPTOGRAPHIC SECURITY

by

JUN WU

A DISSERTATION

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

COMPUTER ENGINEERING

2012

ABSTRACT

This dissertation presents two Null Convention Logic (NCL) applications of asynchronous logic circuit design in nanotechnology and cryptographic security. The first application is the Asynchronous Nanowire Reconfigurable Crossbar Architecture (ANRCA); the second one is an asynchronous S-Box design for cryptographic system against Side-Channel Attacks (SCA).

The following are the contributions of the first application:

1) Proposed a diode- and resistor- based ANRCA (DR-ANRCA). Three configurable logic block (CLB) structures were designed to efficiently reconfigure a given DR-PGMB as one of the 27 arbitrary NCL threshold gates. A hierarchical architecture was also proposed to implement the higher level logic that requires a large number of DR-PGMBs, such as multiple-bit NCL registers.

2) Proposed a memristor look-up-table based ANRCA (MLUT-ANRCA). An equivalent circuit simulation model has been presented in VHDL and simulated in Quartus II. Meanwhile, the comparison between these two ANRCAs have been analyzed numerically.

3) Presented the defect-tolerance and repair strategies for both DR-ANRCA and MLUT-ANRCA.

The following are the contributions of the second application:

1) Designed an NCL based S-Box for Advanced Encryption Standard (AES). Functional verification has been done using Modelsim and Field-Programmable Gate Array (FPGA).

2) Implemented two different power analysis attacks on both NCL S-Box and conventional synchronous S-Box.

3) Developed a novel approach based on stochastic logics to enhance the resistance against DPA and CPA attacks. The functionality of the proposed design has been verified using an 8-bit AES S-box design. The effects of decision weight, bitstream length, and input repetition times on error rates have been also studied. Experimental results shows that the proposed approach enhances the resistance to against the CPA attack by successfully protecting the hidden key.

ACKNOWLEDGMENT

I would like to thank Dr. Minsu Choi, my advisor and the chairman of my doctoral committee. The present work would not have been completed without his great advice and insightful guidance. It has been a pleasure working in his research group.

I would like to express my sincere gratitude to the remainder of my doctoral committee. Dr. Yiyu Shi and Dr. Daryl G. Beetner provided useful integrated circuit design background for the present work. Dr. Sahra Sedighsarvestani and Dr. Dan Lin has taken time to review this work and has provided useful outside perspectives.

Most of my doctoral research was supported in part by the National Science Foundation under Award Number ECCS-0801362.

Finally, I would like to thank my parents, Zhuanfang Wu and Huimin Su for their continuous encouragement and support. I thank my husband, Hengsi, for his great love.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

Synchronous logic with clocked structures has dominated the digital design over the past decades. As the decrease of feature sizes and the increase of the operating frequency of integrated circuits (IC), clock-related issues become more serious, such as clock skews, increased power at the clock edges, extra area, and layout complexity for clock distribution networks, and glitches. These motivate the research of asynchronous (i.e., clockless) logic design which has benefits of eliminating all the clock-related issues listed above.

Null Convention Logic (NCL) is a delay-insensitive (DI) logic that belongs to the category of asynchronous logic. NCL was first proposed by Karl Fant and Scott Brandt in 1994 [1, 2, 3], and further developed by Dr. Scott Smith's research group [4]. NCL initially aimed at designing Application Specific Integrated Circuit (ASIC) and Very-large-scale Integration (VLSI) circuits with lower power, lower noise, and lower electromagnetic interference (EMI). Various NCL based circuits have shown these characteristics. An NCL based Motorola STAR08 processor [5] shows the power and noise reduction up to 40% and 10dB, respectively, comparing to its synchronous counterpart. In [6], an 8-operation NCL ALUs was designed as a benchmark. The simulation result shows that the dual-rail NCL circuit consumes less power but requires a larger area compared with the conventional Boolean logic version. Other designs like NCL divider [7] and NCL multiply-and-accumulate unit [8] have shown the benefits of speed improvement and reduction in power consumption, noise, and EMI.

Designing an NCL circuit is less complex than designing the traditional asynchronous circuit due to the absence of global clock and the property of DI, which does not need worse-case delay analysis and extensive delay matching to synchronize the datapath and control-path without a clock signal. The current research of NCL mainly focuses on the formal design and optimization of NCL, such as the FPGA implementations of NCL[9], Design For Test (DFT) techniques for NCL [10], speed increase of NCL using cycle reduction techniques [11], and timing/gate optimization methods [12]. However, there is few study at the application level of NCL. And its advantages are not fully investigated.

This dissertation proposes and demonstrates two NCL applications to explore some other benefits of NCL. The first application is the Asynchronous Nanowire Reconfigurable Crossbar Architecture (ANRCA). Potential benefits from ANRCA include enhanced manufacturability, scalability, modularity, and robustness. The second NCL application is an asynchronous S-Box design for cryptographic system to resist side-channel attacks (SCA). This design demonstrates that NCL has the advantage of securing cryptographic devices against various power analysis attacks, including Simple Power Analysis (SPA), Differential Power Analysis (DPA), and Correlation Power Analysis (CPA). In order to enhance security, scholastic bit streams based logics were proposed. The stochastic bit streams method has good scalability and it can be applied to many other devices when they require enhanced security.

1.1. OVERVIEW OF NCL

NCL uses dual-rail or quad-rail signaling methods to achieve the DI [4]. A pair of dual-rail signals A^0 and A^1 could be either 10 (i.e., DATA0), 01 (i.e., DATA1), or 00 (i.e., NULL); as 11 is considered invalid. Same for the quad-rail signaling method but it uses four rails instead of two. The possible sets for quad-rail signaling method would be DATA0 (1000), DATA1 (0100), DATA2 (0010), DATA3 (0001), and NULL (0000). NCL uses two states, DATA (i.e., data representation) and NULL (i.e., control representation) to synchronize itself and control the input and output, eliminating the need of a reference clock signal. To mark the transition between the NULL and DATA states, each NCL combination logic must be bracketed by input and output DI registers, these registers have an input/output acknowledgment signal that alternates between 0s and 1s to provide request-for-NULL (i.e.,RFN) and request-for-DATA (i.e.,RFD), respectively. An example is shown in Figure 1.1. These signals are used to initiate a delay insensitive handshaking protocol that handles timing locally. The four-phase handshaking protocol includes: 1) The proper conditions are met to provide DATA at the output of the registration element; 2) RFN goes back to its previous state; 3) All of the inputs to the registration element are at NULL state; and 4) RFD is generated and it goes back to the previous state. The completion detection component is used to determine whether the corresponding pipeline stage is ready for another DATA/NULL cycle. It consists a cascade of NCL AND gates at which the output is fed back to the previous register. When it detects the current operation is a complete DATA set or a complete NULL set, the output will be asserted to request the next cycle. Therefore, the period of DATA-to-DATA cycle consists of four stages:

1. Time for NULL combinational evaluation ($T_{Ni} \rightarrow T_{RN_{i+1}}$);
2. Time for NULL completion acknowledgement ($T_{N_{i+1}} \rightarrow T_{RDi}$);
3. Time for DATA combinational evaluation ($T_{Di} \rightarrow T_{RD_{i+1}}$);
4. Time for DATA completion acknowledgement ($T_{D_{i+1}} \rightarrow T_{RNi}$);

where T_{D_i} and $T_{D_{i+1}}$ represent the propagation time of DATA in the current stage and next stage, respectively. Similarly, T_{N_i} and $T_{N_{i+1}}$ represent the propagation time of NULL on the current stage and next stage, respectively. T_{RNi}, T_{RDi}, $T_{RN_{i+1}}$ and $T_{RD_{i+1}}$ represent the acknowledge time of request for NULL/DATA on the current or next stage, respectively.

Threshold gates provide the basic building block of NCL designs. There are two types of NCL threshold gates: $THmn$ and $THmnWw_1..w_R$, where n represents the number of inputs and m is the threshold value of the gate [13]. This means that at least m of the n inputs must be asserted before the output becomes asserted. Available $w_1..w_R$ are the integer weights of $input_1..input_R$, respectively. For example, a $TH34w2$ gate has $n = 4$ inputs and its weight of the first input is $w_1 = 2$. In order to assert its output, at least three of the four inputs must be asserted since $m = 3$. Figure 1.2 shows the symbol of a $TH34w2$ gate [13]. The inputs and outputs of a threshold gate can be one of two states, NULL or DATA. For example, the 1-bit NCL register consists of two $TH22n$ gates and a $TH12$ gate. A threshold gate starting with its output in an NULL state will remain in the NULL state until the specified number of inputs are placed in the DATA state. Once the gate reaches the DATA state, it remains in this state until all of the inputs return to the NULL state. The hysteresis in the threshold gate provides the threshold needed to keep from switching during intermediate state when the number of inputs in the DATA state is between zero and the threshold limit.

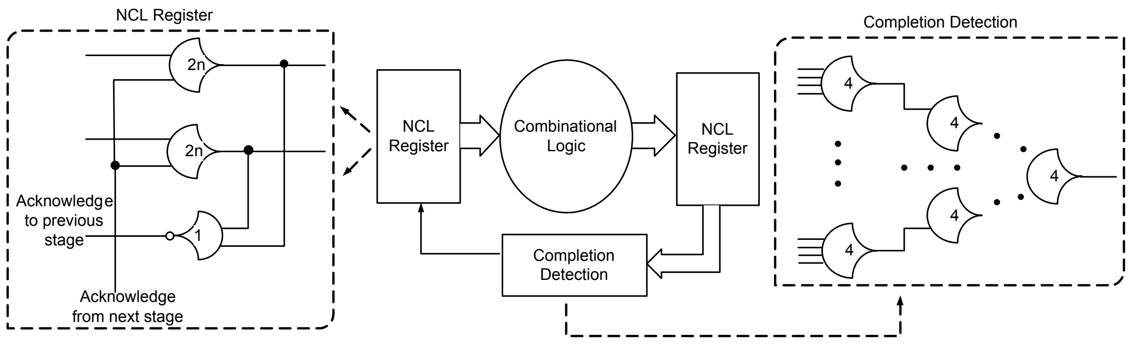


Figure 1.1. Block diagram of NCL combinational logic with two DI registers.

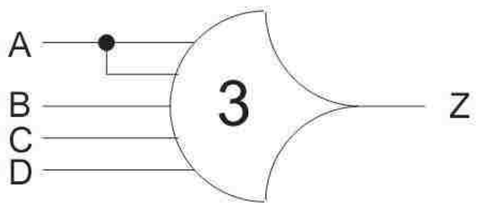


Figure 1.2. Threshold gate TH34w2.

There are twenty-seven fundamental threshold gates (TH gates) with hysteresis capability [13]. Each uses no more than four inputs. The TH gate provides a threshold for the output assertion condition and hysteresis for state-holding behavior. This allows all the inputs to be incorporated before generating the outputs, ensuring a complete transition. Any arbitrary logic(s) can be obtained by using different combinations of these TH gates [4]. For example, the Boolean equation of $TH23$ gate is $Z = AB + BC + CA + (A + B + C)Z^*$ where $AB + BC + CA$ is the threshold term, $A + B + C$ is the hold condition, and Z^* is the previous output. Table 1.1 shows the truth table of this gate. When Z^* is 0, the updated output is the same result of the threshold equation, but once it becomes 1, the output will remain 1 until all inputs (i.e., A, B, C) become 0 eventually. This principle is designed in all TH gates that are shown in Table 1.2.

Table 1.1. Truth table of NCL TH23 gate.

Z^*	A B C	Z	Z^*	A B C	Z
0	0 0 0	0	1	0 0 0	0
0	0 0 1	0	1	0 0 1	1
0	0 1 0	0	1	0 1 0	1
0	0 1 1	1	1	0 1 1	1
0	1 0 0	0	1	1 0 0	1
0	1 0 1	1	1	1 0 1	1
0	1 1 0	1	1	1 1 0	1
0	1 1 1	1	1	1 1 1	1

In order to retain DI, NCL circuits should satisfy the condition of input-completeness and observability [14]. Input-completeness requires that the transition of all outputs in a combinational circuit should wait until all inputs have transitioned from either NULL to DATA or DATA to NULL completely. In another word, all outputs cannot transition before all inputs arrive. The observability condition ensures that every gate transition is observable at the outputs, meaning that there is no orphans could propagate through a gate [13]. For example, an incomplete NCL AND function ($Z = A \bullet B$) can be designed using a $TH12$ gate and a $TH22$ gate, where $Z^1 = A^1 \bullet B^1$ and $Z^0 = A^0 + B^0$. It is incomplete because the output can transfer from NULL to DATA0 without both inputs are DATA, which breaks the condition of input-completeness. To make it complete, the equation for DATA0 $Z^0 = A^0 + B^0$ can be changed to $Z^0 = A^0(B^0 + B^1) + B^0(A^0 + A^1) = A^0B^0 + A^0B^1 + A^1B^0$; Therefore, a complete NCL AND function could be designed using a $THand0$ gate and a $TH22$ gate.

1.2. ADVANTAGES OF USING NCL

NCL is an asynchronous logic, which eliminates the need for a global clock and the clock distribution network. Therefore, timing design is easier than its synchronous counterpart due to the lack of requirement to compensate clock skew, clock jitter, and glitches. Unlike traditional asynchronous design techniques (i.e., Huffman circuits)

Table 1.2. Twenty-seven NCL TH gates.

NCL TH Gate	Boolean Equation	NCL TH Gate	Boolean Equation
TH12	A+B	TH22	AB
TH13	A+B+C	TH23	AB+AC+BC
TH33	ABC	TH23w2	A+BC
TH33w2	AB+AC	TH14	A+B+C+D
TH24	AB+AC+AD+BC+BD+CD	TH34	ABC+ABD+ACD+BCD
TH44	ABCD	TH24w2	A+BC+BD+CD
TH34w2	AB+AC+AD+BCD	TH44w2	ABC+ABD+ACD
TH34w3	A+BCD	TH44w3	AB+AC+AD
TH24w22	A+B+CD	TH34w22	AB+AC+AD+BC+BD
TH44w22	AB+ACD+BCD	TH54w22	ABC+ABD
TH34w32	A+BC+BD	TH54w32	AB+ACD
TH44w322	AB+AC+AD+BC	TH54w322	AB+AC+BCD
THxor0	AB+CD	THand0	AB+BC+AD
TH24comp	AC+BC+AD+BD		

[15] that need timing analysis to achieve the delay matching so as to synchronize the datapath and control path with the absence of a clock. NCL circuits do not need such extensive timing analysis, which makes the design much less complex. In another word, NCL has the potential to process at its maximum frequency due to the fact that the data go through path with minimal delay. This allows a NCL circuit to potentially operate faster than a Boolean asynchronous design.

Another benefit of NCL is the lower power consumption. The demonstration could be found in section 3 that total power consumption of both synchronous S-Box and NCL S-Box is compared based on the measurement results of EDA tools and FPGA simulation. The rational is NCL's monotonic transition between DATA wave and NULL wave, creating an idle power state and eliminating the glitch power [13]. NCL circuits only switch when useful work is being performed, not every clock edge like Boolean circuits. NCL systems also distribute the demand for power over time and area, reducing the occurrence of hot spots, system noise, and peak power demand [16]. According to [2], because of the DI, NCL is insensitive to the changes of physical implementations and parameters, such as the scale changes, variations in propagation delay, aging issues, temperature, manufacturing variations, and so on. Therefore, a NCL circuit is anticipated to operating at a lower voltage with fast speed when the high performance is not required. As discussed in later sections, the NCL's power consumption characteristic has the benefit of increased security for cryptographic devices.

This dissertation have explored more advantages of using NCL in two application areas. They are: 1) NCL based nanowire crossbar architectures, which have the benefit of enhanced manufacturability, scalability, modularity, and robustness. 2) NCL based AES S-Box design that could successfully resist various SCAs, including DPA and CPA. More specific explanations will be presented in Section 2 and Section 3, respectively.

1.3. WORKS SUMMARY

This dissertation consists of three projects.

The NCL-based nanowire crossbar reconfigurable architectures are presented in Section 2. It studies two implementations of ANRCA and discusses the advantages of using asynchronous logic for nanoscale crossbar structure. The proposed ANRCAs are unique for two reasons: 1) It is based on asynchronous NCL, clock-related failures can be removed; 2) It addresses design, test, and manufacturing issues in nanowire crossbar architecture by designing hierarchical structure, introducing function test, and presenting fault-tolerance and repair strategies. Part of Section 2 comes from the following publications:

- Advances in Nanowire-Based Computing Architectures, a book chapter published in Cutting Edge Nanotechnology, 2010.
- Latency/Area Analysis and Optimization of Asynchronous Nanowire Reconfigurable Crossbar System, Nano Communication Network 2010.
- Asynchronous Nanowire Reconfigurable Crossbar Architectures, submitted to IEEE Transaction on Nanotechnology 2011.
- Area and Latency Measurement and Optimization of Clock-Free Nanowire Reconfigurable Crossbar Systems, IEEE I2MTC 2010.
- Memristor Lookup Table (MLUT)-Based Asynchronous Nanowire Crossbar Architecture, IEEE Nanotechnology 2011.

- Post-Configuration Repair Strategy for Asynchronous Nanowire Crossbar System, IEEE MWSCAS 2012.

- Configurable Logic Block (CLB) Design for Asynchronous Nanowire Crossbar System, IEEE MWSCAS 2012.

Section 3 focuses on the development of NCL based AES S-Box. The objective is to demonstrate NCL S-Box could effectively resist various SCAs and has lower total power consumption than its synchronous counterpart. The hardware implementation of NCL S-Box is designed in VHDL and simulated using both EDA tools and a SCA evaluation FPGA board (SASEBO-GII). Part of Section 3 comes from the following publications:

- Measurement and Evaluation of Power Analysis Attacks on Asynchronous S-Box, accepted for publication in IEEE Transaction on Instrumentation and Measurement 2012.

- FPGA-based Measurement and Evaluation of Power Analysis Attack Resistant Asynchronous S-Box, IEEE I2MTC 2011.

- Asynchronous Nanowire Reconfigurable Crossbar Architectures, GLSVLSI 2010.

Section 4 presents a novel implementation of S-Box design that is based on stochastic logic. By involving probabilistic bit streams in logic implementations, power traces become more unpredictable and data independent. This highly randomness property is helpful for cryptographic devices against power analysis attacks. However, such nondeterministic encoding scheme might generate logic errors. Therefore, different factors that would improve the accuracy have been analyzed in this section, including the length of bit streams, the decision weight, and the repetition times of the inputs. The enhanced security of stochastic logic-based AES S-Box has been experimentally verified on the same SASEBO-GII board. Part of section 4 will be submitted to IEICE Electronics Express (ELEX) 2012.

Finally, Section 5 discusses the contribution of this dissertation and suggestions for future works.

2. PROJECT I: ANRCA

2.1. BACKGROUND

Many challenges have arisen with continued scaling of Complementary Metal-Oxide-Silicon (CMOS) technology, including the increase of integrated circuit complexity, increase in frequency, power density, non-recoverable expenses, and so on. These difficulties have made it troublesome to further progress with, leading nanotechnologies to take the forefront of continuing the technological advancement. One of the most promising nanotechnologies is the nanowire crossbar-based architecture: a two-dimensional array formed by the intersection of two orthogonal sets of parallel and uniformly-spaced synthesized nanowires such as carbon nanotubes (CNTs) and silicon nanowires (SiNWs) [17, 18, 19]. These wires can be aligned to construct an array with nanometer-scale spacing and formed crosspoints of nanoscale wires that can be used as functional logic devices. These devices include programmable diodes, field-effect transistors (FETs), and memristors, depending on the nature of nanowires and interlayer material. The resulting structures act like programmable logic arrays to implement conventional logics. The memristors based nanowire crossbar could also be used to build memories due to its non-volatile characteristic. A typical nanowire crossbar structure is shown in Figure 2.1. Nanowire crossbars offer both opportunities and challenges. One of the opportunities is to achieve ultra-high density which has never been achieved by photolithography (a density of 10^{11} crosspoints per square centimeters has been reported in [20]). The most important challenge is to make them reliable enough in computational applications because of the high fabrication defect densities (as high as 10%, are expected [21]).

Synthesizing nanowires can be done efficiently through bottom-up fabrication paradigm, meaning that the CNTs and SiNWs are synthesized first, then assembled into functional devices. The traditional top-down lithographic manufacturing would not be practical [22]. Combined with fluidic flow techniques by Langmuir-Blodgett (LB) [23], these wires can be aligned to construct arrays that can be used multiple times to yield complex hierarchically assembled nano-systems. Unfortunately, this

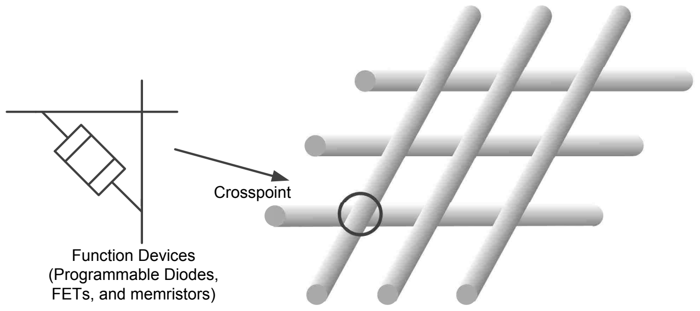


Figure 2.1. Typical nanowire crossbar structure.

technique can lead to random breaks between the ends of the nanowires from rearranging them to scale down the size. Thus, a hierarchical structure is better than a monolithic structure for integrated nano-systems due to the fact that small amount of defects would not cause system failure. Hewlett-Packard (HP)'s lab have successfully fabricated 8×8 (i.e., 64 bits) crossbar memory arrays by using a fabrication technique of nanoscale crossbar called nano-imprint lithography [24, 25]. The fabrication process reveals that the amount of defects increases proportionally to the decrease in the size of the memory arrays. Challenges still arise with the enhancement of lithographic resolutions and defect tolerance even though these fabrication techniques produce results that are quite favorable for synthesizing nanowires efficiently.

Until now, various nanowire crossbar structures have been proposed: Dehon et al. [26] have developed an operational reconfigurable computing system, known as NanoPLA, utilizing these nanowires and molecular-scale devices. HP [27, 28] announced a computational structure with a decoder scheme for addressing nanowires with micro-scale wires. They have also recently demonstrated a nanoscale crossbar-based memristor array [29]. An analysis of diode-resistor based nanoPLA was present in [30]. Other nanoscale reconfigurable homogeneous architectures, such as NanoFabric [31, 32, 33]; follow a similar principle; grouping multiple crosspoints together to serve as a memory device or logic element, and the CMOS/Nano hybrid structure has been used to configure the interface.

2.2. ADVANTAGES OF ANRCA

Unlike those proposed nano-structures that are all based on synchronous operation, the presented work proposes Asynchronous Nanowire Reconfigurable Crossbar Architectures (ANRCA). They have a distinct feature of asynchronous operation, which is based on the NCL, a delay-insensitive data encoding and self-timed logic[14]. Potential benefits of using NCL for nanowire crossbar design include:

1. Easy for manufacturing because of the clock-less characteristic, eliminating all clock-related hardware.

2. ANRCAs are designed in a bottom-up manner and integrated without the trouble of synchronizing each module, which indicates better scalability and modularity than its clocked counterparts because the timing complexity remains the same as the circuit size increases.

3. The complemented primary inputs (e.g., $\overline{A}$, $\overline{B}$, $\overline{C}$ and $\overline{D}$) used in NanoFabric [31] and the Field Effect Transistor (FET) based inverters used in NanoPLA [26, 18] are no longer needed because they can be implemented simply by crossing over the NCL dual-rail signals.

4. Due to non-determinism in the bottom-up self-assembly approach, exhibiting variations in physical parameters in nanowire crossbar structures. These variations would have negative effects on the timing behavior of circuits. ANRCAs are independent of timing issues, thus they are anticipated to have better robustness over the design parameter variations.

5. Stuck-at-1 faults are easy to detect, which relatively reduces testing complexity. Once a fault has occurred, the NCL circuit will be halted because it interferes with the transition from DATA (either 01 or 10) to NULL (00). Also, the dual-rail signal 11 is invalid in NCL; therefore, any permanent or transient fault that results in this invalid state can be easily detected.

This section presents two implementations of ANRCA. One uses crosspoints as programmable diodes that create AND/OR planes to implement logic gates (Section 2.3). The other uses configurable memristors to realize nanoscale lookup tables (LUTs)(Section 2.6). Discussions of defect issues and repair strategies for these two optimization models are comprehensively studied. Comparisons between these two optimization models have been analyzed in terms of area, programming steps, crosspoints utilization rate, and defect tolerance(Section 2.8). The above contributions of this project are particularly beneficial in designing an optimized reconfiguration hardware fabrication and in efficiently mapping any given complex logic.

2.3. DIODE- AND RESISTOR-BASED ANRCA (DR-ANRCA)

The primitive unit of DR-ANRCA is the Programmable Gate Macro Block (PGMB), DR-PGMB for short. A single DR-PGMB is made of six horizontal nanowires that cross over eleven vertical nanowires to form sixty-six crosspoints. These crosspoints are formed by programmable diodes to create an AND/OR logic plane. The use of pull-up resistors on the vertical nanowires create the AND-plane, and pull-down resistors on the horizontal wires create the OR-plane, enabling each DR-PGMB to be programmed to realize any given NCL threshold gate function in sum-of-product (i.e., SOP) form. For example, Figure 2.2 shows a DR-PGMB implementing a $TH23$ NCL gate, whose boolean expression is $Z = AB + BC + CA + (A + B + C)Z^*$, where Z^* represents the previous output value of the $TH23$ gate, which is fed back to the input nanowire.

For the purpose of efficiently reconfiguring the given DR-PGMB as one of the 27 arbitrary NCL threshold gates, three configurable logic block (CLB) structures were designed: CLB-1, CLB-2, and CLB-3. Figure 2.3 shows that the CLB-1 structure consists of four DR-PGMBs which are surrounded by nanowires and demultiplexers. The demultiplexers are used as the interface between microwires and nanowires since they can use a small number of microwires to control a relatively large number of nanowires. The input lines for addressing the demultiplexers are microwires, which can be implemented on a CMOS substrate [34]. The output lines are nanowires that are used to control crosspoints in DR-PGMBs. The number of input microwires (M) and output nanowires (N) is expressed as $N = 2^M$. The demultiplexers must be placed

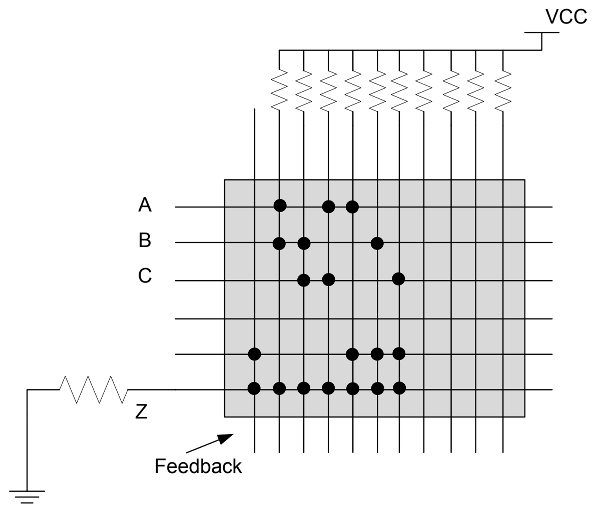


Figure 2.2. TH23 gate realized on a programmable gate macro block.

on the side of the rows and columns to program all the crosspoints to ON/OFF states by applying positive or negative voltages. Assuming that the unselected outputs are driven by ground, then the intersection of driven nanowires would have a voltage drop that is different from other unselected crosspoints. This would allow the configuration of crosspoints sensitivity to voltage to drop across them. For example, driving a positive voltage on one of the column nanowires and a negative voltage on one of the row nanowires, the rest of nanowires would be driven with ground, the selected crosspoint is defined as the intersection of driven nanowires.

The CLB-1 shown in Figure 2.3 is configured to function as a NCL full adder. Two TH23 gates and two TH34w2 gates are implemented in four PGMBs, then, interconnected in the configurable interconnection grid. Three inputs X, Y, Ci and two outputs S, Co are represented by X^0, X^1, Y^0, Y^1, Ci^0, Ci^1, Co^0, Co^1, S^0, S^1 encoded in dual-rail logic. The top demultiplexer is used to decode the input signals from DR-PGMB to determine which nanowires are selected. The other demultiplexers are used to select different DR-PGMBs to receive those input signals. The following steps are used to program logic onto the CLB: 1) Use the top and right demultiplexers to choose input crosspoints; 2) Use the bottom and right demultiplexers to map the selected crosspoints to each DR-PGMB; 3) Program each DR-PGMB to the required threshold gate. 4) Retrieve the generated output from the designated DR-PGMB.

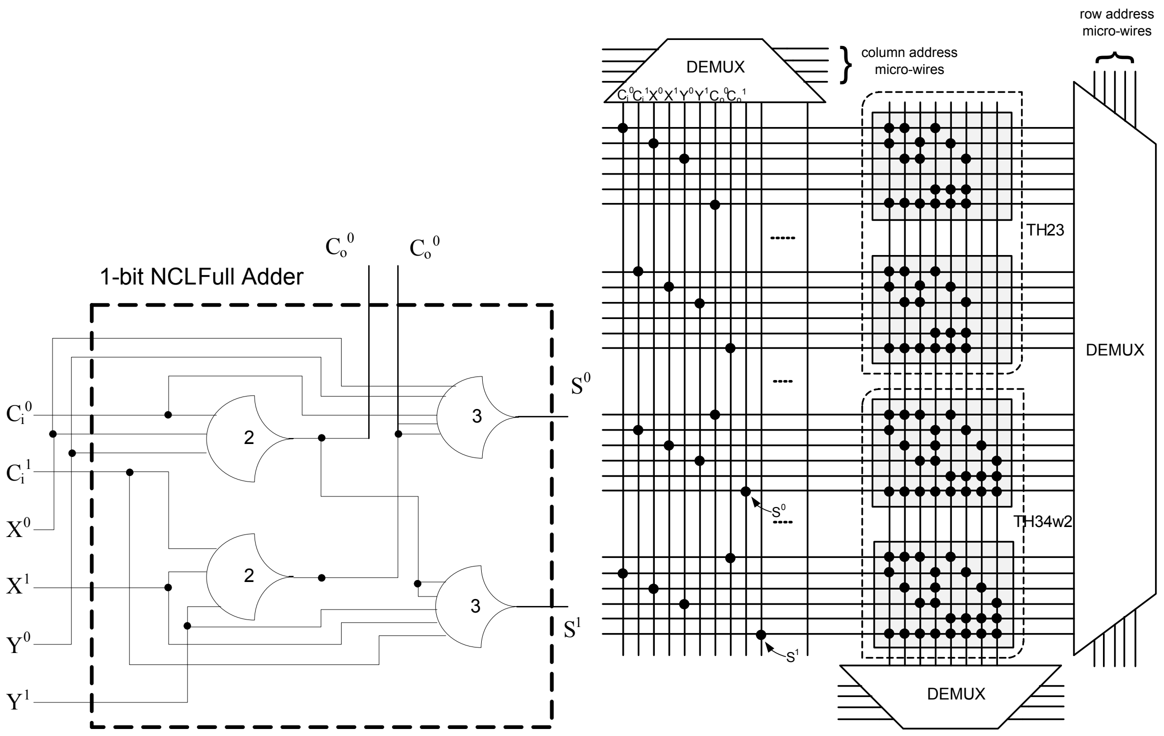


Figure 2.3. NCL full adder implemented in CMOS (left) and CLB-1 (right).

The three CLBs are distinguished by the number of demultiplexers on the right side: CLB-1 has one, while CLB-2 and CLB-3 (Figure.2.4) have two and four, respectively. The microwire address lines may be shared among multiple demultiplexers to allow multiple crosspoints to be simultaneously accessed.

NCL delay-insensitive registers are needed to bracket the combinational logic design to achieve the transition between DATA and NULL state. An 1-bit NCL register can be implemented on a single CLB using two $TH22$ gates and a $TH12$ as shown in Figure 2.5. The schematic of an 1-bit NCL register has been shown in Figure 1.1. I^0, I^1, O^0 and O^1 represent input and output data rails, respectively. K_i and K_o are the handshaking signals. The complemented value of K_o can be designed by crossing over the wires due to its dual-rail property. For the full adder design, an 3-bit input register (i.e., three CLBs) and 2-bit output register (i.e., two CLBs) are needed to provide the appropriate number of input/output signals. Thus, an FPGA-like hierarchical architecture was proposed to implement the higher level logic that requires a large number of DR-PGMBs, such as multiple bits full adder. Figure

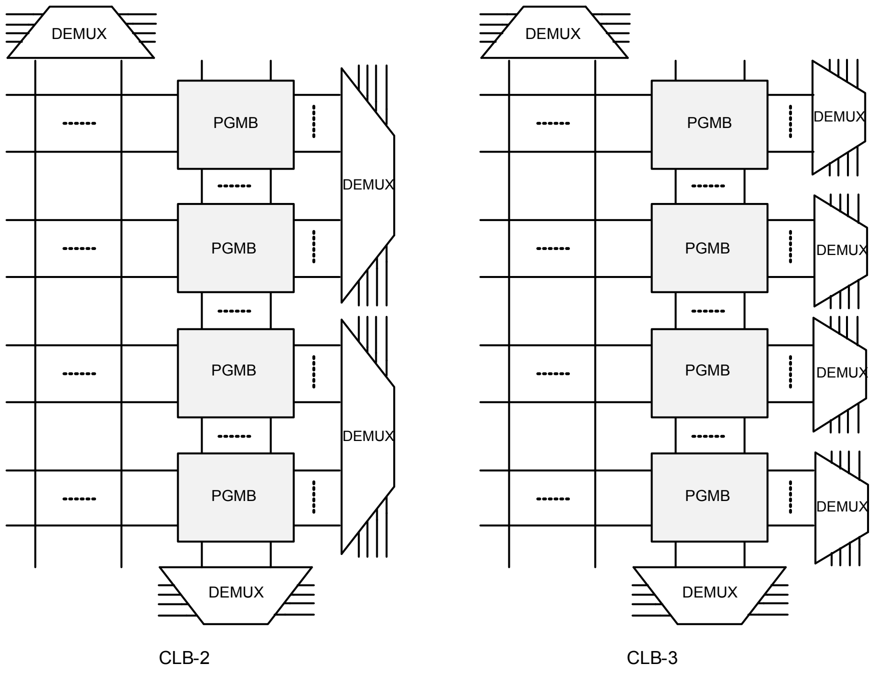


Figure 2.4. The second and third variations of (CLB-2 and CLB-3).

2.6 shows the structure of the two-level hierarchical architecture that can be used for implementing a 4-bit adder. Compared with the nanowire crossbar 3-bit adder design that was introduced in [35], although the proposed DR-ANRCA consumes more area, it is relatively easier to reconfigure and unaffected from any timing variations because of its delay-insensitivity.

2.4. NUMERICAL ANALYSIS OF DR-ANRCA

2.4.1. Area. The area-efficient demultiplexer designs reported in [36] are used in the proposed CLB designs. Thus, the demultiplexers selected by the DR-PGMB on the right are assumed to be allocated on the same column-wise area of the substrate. According to the NanoPLA assembly parameters provided by Dehon[26], the lithographic interconnect pitch can be estimated as $105nm$ for the $45nm$ node. A $10nm$ nanowire pitch is acceptable for assembling the crosspoints between each pair of crossed nanowires. Nanowire with $3nm$ diameters has been demonstrated. Microwires have diameters of around $45nm$ [26]. The pitch between microwires and nanowires is assumed to be a median of P_n and P_m, which is around $60nm$. Thus, as

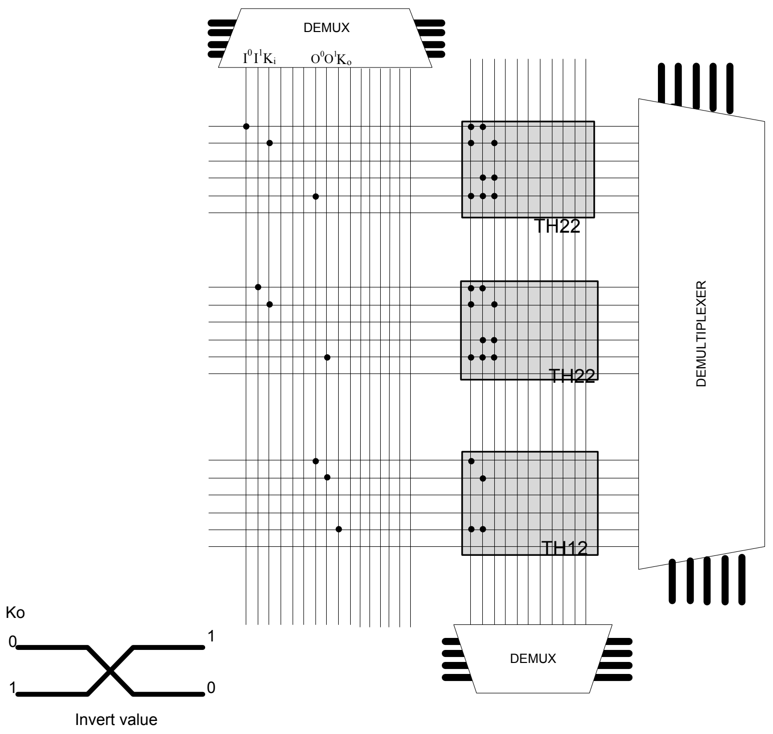


Figure 2.5. 1-bit NCL register implemented in one CLB.

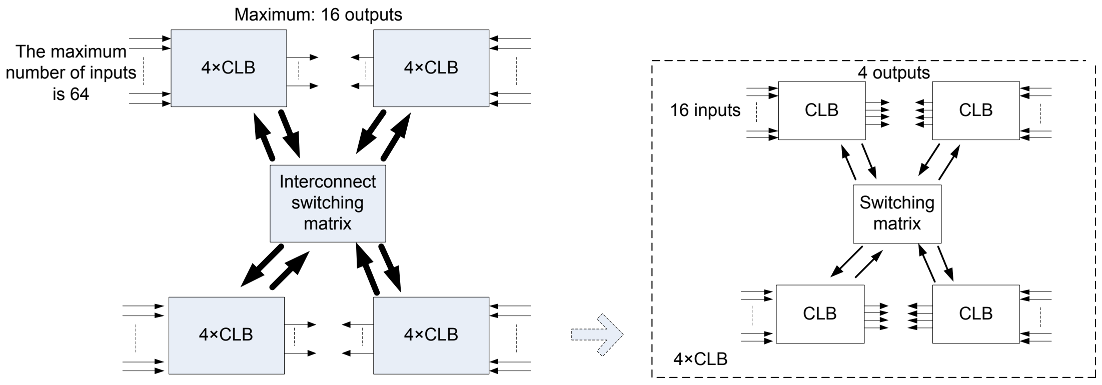


Figure 2.6. Two-level hierarchical architecture for the proposed ANRCA.

shown in Figure 2.3, the overall area of each CLB can be estimated as the product of the width and height ($Area = Width \times Height$):

$$Width = P_n \times (N_v - 1) + P_m \times (M_v - 1) + P_{mn} + N_v \times D_n + M_v \times D_m \quad (1)$$

$$Height = P_n \times (N_h - 1) + P_m \times (M_h - 1) \times 2 + P_{mn} \times 2 + N_h \times D_n + M_h \times D_m \times 2 \quad (2)$$

where: N_v and N_h represent the number of nanowires in vertical and horizontal direction, respectively; M_v and M_h represent the number of microwires for demultiplexers in vertical and horizontal direction, respectively; P_n and P_m, D_n and D_m are the pitch and diameter of the nanowires and microwires, respectively; and P_{mn} is the pitch between microwires and nanowires. Thus, in calculating their areas using the above equations, the estimated areas of the proposed CLB-1, CLB-2, and CLB-3 are $1.477um^2$, $1.265um^2$, and $1.053um^2$, respectively.

2.4.2. Programming Steps. The complexity of the logic mapping operation is another factor that should be considered to optimize the design. A programming step is defined as selecting and programming a specific crosspoint in a given CLB. The average programming step count is defined as the estimated number of steps needed to program a given number (i.e., 2, 4, 8, or 10) of randomly distributed crosspoints among DR-PGMBs. For example, say there are four crosspoints need to be programmed and they are randomly distributing on the crossbar. If they are programmed by the design of CLB-1, each point would be selected one by one since there is only one PGMB select demultiplexer. Thus the programming step count in this case is four. For CLB-2 and CLB-3, actual programming step count depends on the geometric distribution of crosspoints being programmed because different demultiplexers are controlling the selection of rows. One extreme case is that the four crosspoints happen to locate on the same column of each PGMB controlled by the four demultiplexers of CLB-3. The total step of programming these four points is one since four demultiplexers can select them simultaneously in this case. However, other distributions should be considered too. Therefore, a numerical analysis program for calculating the average steps has been implemented in MATLAB. A probabilistic algorithm and a square curve fitting algorithm were developed to derive the average number of steps. Figure 2.7 shows the relationship between the number of crosspoints to be programmed and the average number of steps required to fully program them.

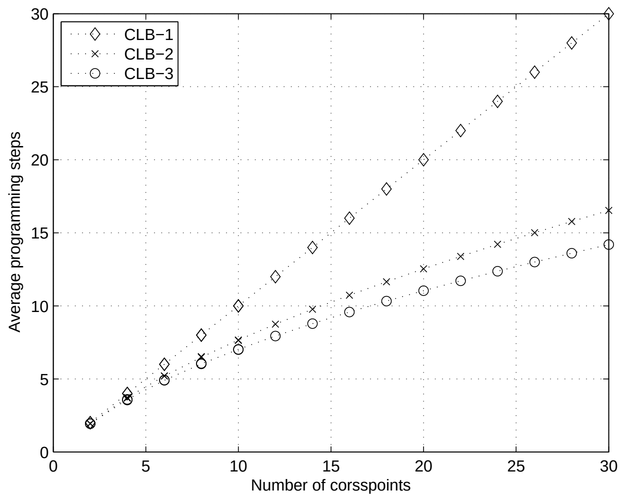


Figure 2.7. Curve fitting results of programming steps for various CLBs.

2.4.3. Latency. Latency is defined as the total time required to process input to generate output. System latency for the proposed ANRCA consists of two parts: the cycle of combinational logic implemented by the CLBs and the cycle of NULL and DATA handshaking feedback signal propagation. A single cycle in one DR-PGMB of CLB evaluates both input (i.e., in AND-plane) and output (i.e., in OR-plane). The total time for a single input/output plane is estimated based on the following equations: the latency for the input plane and output plane can be estimated as

$$T_{inPlane} = N_{ci} \times T_d \tag{3}$$

$$T_{outPlane} = N_{co} \times T_d \tag{4}$$

and the latency for a demultiplexer is estimated as

$$T_{propagation} = N_p \times T_p \tag{5}$$

thus the overall latency is

$$\begin{aligned} T_{cycle} &= T_{inPlane} + T_{outPlane} + T_{propagation} \\ &= (N_{ci} + N_{co}) \times T_d + N_p \times T_p \end{aligned} \tag{6}$$

where: N_{ci} and N_{co} represent the number of input and output crosspoints of an DR-PGMB, respectively; T_d is the processing time of demultiplexers; T_p is the propagation delay of programming one crosspoint on its DR-PGMB. N_p represents the total number of crosspoints on an DR-PGMB. The comparison for the latency of different CLBs, is dependent on the logic that is implemented on it. A full adder design is used as a benchmark to make a simple comparison of the latency among three CLBs: CLB-1, CLB-2, and CLB-3 is $18T_d + 90T_p$, $10T_d + 54T_p$, and $5T_d + 37T_p$, respectively.

Generally, a single NCL cycle comprises the propagation delay of NULL/DATA and the acknowledge time of a request for DATA/NULL [37] as Figure 2.8 shows.

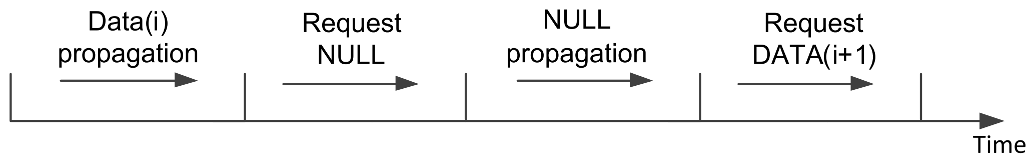


Figure 2.8. A single NCL cycle of DATA-to-DATA latency.

The proposed ANRCA also follows this rule since it is based on NCL. Therefore, the DATA-to-DATA latency is given by:

$$T_{D_i to D_{i+1}} = T_{D_i} + T_{RNi} + T_{N_i} + T_{RD_{i+1}} \tag{7}$$

where T_{D_i} and T_{N_i} represent the propagation time of DATA and NULL in the current stage, respectively; T_{RNi} is the acknowledge time of request for NULL in the current stage; $T_{RD_{i+1}}$ is the acknowledge time of request for DATA in the next stage.

2.5. DEFECT/FAULT ISSUES OF DR-ANRCA

One of the main problems with nanoscale crossbar architecture is the high inherent defect density that is caused by the bottom-up self-assembly fabrication technique[38]. The conventional fault-tolerance and reliable design are not adequate in nanoscale integration because of the increased defect and fault rates. Therefore, fault-tolerance techniques for nanowire crossbar structure is critically needed. Several test algorithms have been developed by our research group, including defect-unaware, defect-aware, and function test algorithm (FTA) [39, 40]. This dissertation uses FTA to detect defective crosspoint locations, focuses on the development of post-configuration repair technique, and fault-tolerance techniques to increase the reliability of the proposed ANRCAs. The fault-tolerance approaches include permutation, commutative method, and redundancy. There are three ways to tolerate defects in a DR-PGMB:

1. Reconfigure the order of primary inputs utilizing the demultiplexer that is designated to select inputs. The number of combinations for rearranging the order of inputs can be determined from factorial of the number of inputs.

2. Rearrange the order of columns of the given DR-PGMB base on the commutative law since the product terms in the sum-of-product (SOP) of TH gates can be rearranged.

3. Include redundant rows and columns in each DR-PGMB to increase the chances of generating the correct output by forcing the desired logic.

Using a $TH23$ gate (i.e., $Z = AB + BC + AC + (A + B + C)Z^*$) as an example, the above defect tolerance methods are demonstrated in Figure 2.9. It shows a side-by-side comparison of a defective DR-PGMB with a corrected DR-PGMB that is based upon the respective methods described above. Figure 2.9(a) shows the matrix implementation of a $TH23$ gate on a DR-PGMB with defects on the intersection of

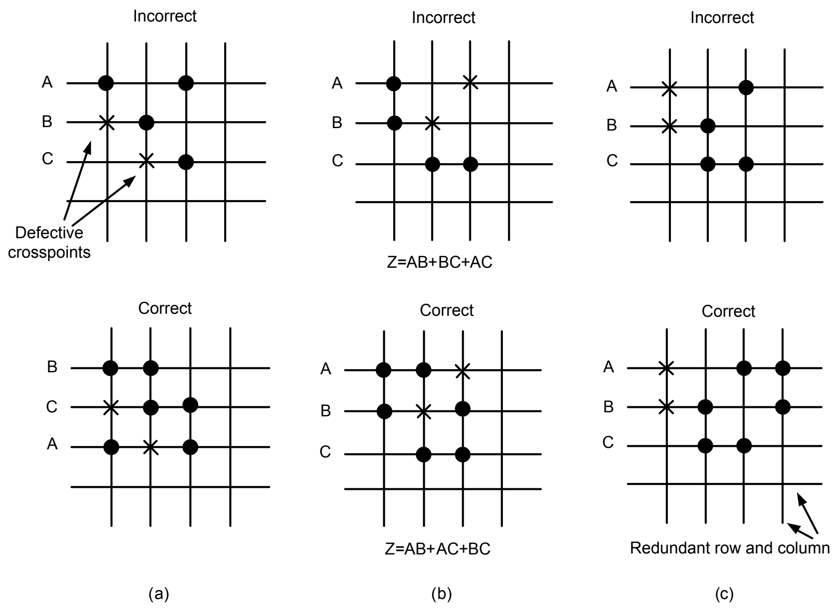


Figure 2.9. Demonstration of defect tolerance methods.

(2,1) and (3,2). To avoid these defective crosspoints, the input order can be simply changed from A,B,C to B,C,A. Figure 2.9(b) shows the defective crosspoints located at (2,2) and (1,3); in this case, the SOP can be rearranged from $Z = AB + BC + AC + (A + B + C)Z^*$ to $Z = AB + AC + BC + (A + B + C)Z^*$. Figure 2.9(c) shows the defective crosspoints located at (1,1) and (2,1). The product term of the $TH23$ gate can not be mapped on the first column without using extra rows. Therefore, the solution is to use extra rows or columns. If the number of defective crosspoints are too large to be tolerated by the selected DR-PGMB, the worst-case scenario is to discard the selected DR-PGMB and use another one to reprogram the given TH gate function.

FTA uses a test tuple that combines the input patterns and the previously asserted output patterns to list all possible faults for the mapped TH gate. Based on FTA, a new post-configuration repair technique was derived to provide a balanced combination of tolerable repair time and acceptable repair performance [39, 40]. Test tuples having one-to-one correspondence with defective crosspoints are applied first to detect and then isolate DR-PGMB rows and columns that are needed to be repaired.

Consider a TH34w2 gate shown in Figure 2.10. The black dots represent programmed-ON crosspoints and crosses represent defective crosspoints. The combination of both symbols represents defective crosspoints that affect the functionality of the gate. To implement a TH34w2 gate on a DR-PGMB, a total of 25 crosspoints must be programmed on the specific locations. In this case, there are 7 defective crosspoints among a total of 66 crosspoints. The defect rate is approximately 10% and five of the defective crosspoints, (2,4), (4,8), (6,1), (6,5), (6,8) affect the functionality of the TH34w2 gate and three of them are located on the OR plane. Thus, according to the functionality of TH34w2 (i.e., $Z = AB + AC + AD + BCD + (A + B + C + D)Z^{*}$) and its specific coordinates shown in Figure 2.10 part (a), a test tuple table (see Table 2.1) was generated for TH34w2 to sort the test vectors for related defective crosspoints. The OR plane takes the priority because it reflects the various results from the AND plane [40]. The initial state of input is assumed to be 0000 so that the output Z will be 0 as well. Any stuck-at-1 crosspoints in the OR plane could be tested using this test vector. Although test tuple 0000 could be used to test all the required crosspoints, it takes a long time to go through all the required crosspoints (i.e., 25 in this example). Therefore, a combinations of various inputs could make the testing more efficient. For example, the input 0001 with $Z^{*} = 0$ (i.e., Z^{*} is the previous output) could be used to test the defective crosspoints with coordinates (1,3), and (5,8). According to the functionality of $TH34w2$, the correct result would be 0 when input is 0001 and Z^{*} is equal to 0. However, it becomes 1 since the crosspoints (1,3) is defective. A similar testing algorithm could be used for other crosspoints required to implement a $TH34w2$ gate. Thus the fault location has been detected whenever the observed output does not match the desired one. Using this method, all 25 required crosspoints could be covered within 20 steps.

Once the location of the defective crosspoints have been confirmed, the repair strategy could be developed to avoid the defective crosspoints by either rearranging the mapping or using extra rows or columns. As Figure 2.10 part (b) shows, with column 4 moved to column 10 and column 8 moved to column 9, the remaining two defective crosspoints with initial locations (6,1) and (6,5) are moved to (7,1) and (7,5), respectively. The existing defective crosspoints could be avoided. However, another round of FTA should be used to test the new location of these crosspoints to ensure

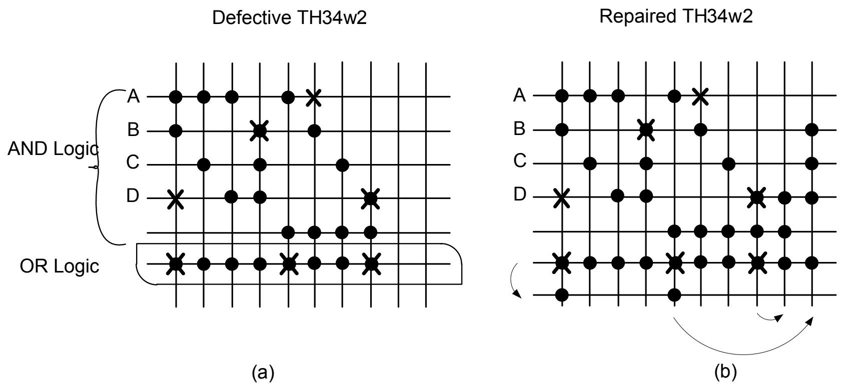


Figure 2.10. A case study of successfully realized on a defective DR-PGMB of TH34w2 gate.

the functionality of $TH34w2$ would work property. The algorithm could be modified to test other TH gates on the DR-PGMB.

2.6. MEMRISTOR LOOK-UP-TABLE BASED ANRCA

Memristor, another promising technology for nanoscale computation systems, is considered as the fourth fundamental circuit element [41]. It is expected to have advantages in building nanoelectronic memories, computer logic, and neuromorphic computer architectures [42]. It is also capable of replacing programmable resistors or rectifying devices to yield configurable crossbar junctions (i.e., crosspoints). There are two important properties of memristors[43] : (a) as a memory storage element, each memristor crosspoint can be programmed independently into a low-conductance (Logic 0) or a high-conductance (Logic 1) as normal resistive switching elements. (b) as a switche, low-conductance (OFF-state) and high-conductance (ON-state) refer to the unconnected state and the connected state, respectively. Various research papers describe the way to write/read memristor based nanowire crossbar[44, 45, 46]. A tutorial of using memristor-based crossbars has been presented in [47]. Using the properties of memristors, this work proposes another implementation of ANRCA that relies on configurable memristors to realize nanoscale look-up-tables (LUTs).

Table 2.1. Test Tuples for TH34w2 gate.

Test Tuples (input bits)	Required Crosspoints Coordinates
0000	OR Plane and F(1,5), (2,6), (3,7), (4,8)
0001	F(1,3), F(5,8)
0010	F(1,2), F(5,7)
0011	F(2,4), F(1,3), F(1,2)
0100	F(1,1), F(5,6)
0101	F(1,1), F(1,3), F(3,4)
0110	F(1,1), F(1,2), F(4,4)
0111	F(1,1), F(1,2), F(1,3)
1000	F(2,1), F(3,2), F(4,3), (5,5)
1001	F(2,1), F(3,2)
1010	F(2,1), F(4,3)
1011	F(2,1), F(2,4)
1100	F(3,2), F(4,3)
1101	F(3,2), F(3,4)
1110	F(4,4), F(4,3)
1111	N/A

The basic unit of MLUT-ANRCA is the MLUT based PGMBs, MLUT-PGMB for short. A single MLUT-PGMB consists of eight horizontal nanowires crossing over four vertical nanowires that are surrounded by column/row demultiplexers and a multiplexer (see Figure 2.11). It can be programmed to realize any given NCL gate by directly implementing the truth table of the given gate function using MLUT with the hysteresis (i.e., state-holding behavior) that is required to achieve the proposed delay-insensitivity via a feedback interconnect.

Figure 2.11 shows the implementation of a MLUT-PGMB programmed to function as an NCL $TH23$ gate. The demultiplexers located at the row and column

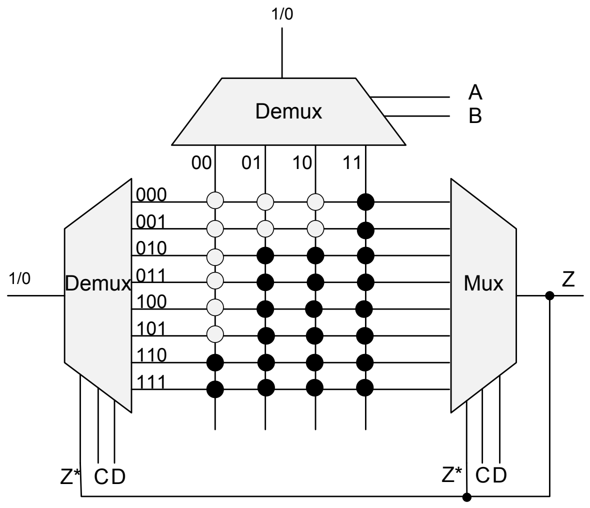


Figure 2.11. TH23 gate realized on a MLUT-PGMB.

of the MLUT-PGMB are utilized to select a programmable memristor junction. Light-colored dots represent the memristor crosspoints programmed as 0, while the dark-colored dots are 1. A feedback signal is sent to provide Z^* to the row demultiplexers for hysteresis behavior. If Z^* is equal to 0, then the top half of the LUT is selected to provide the output setting logic. If Z^* is equal to 1, the bottom half of the LUT is selected to provide the output resetting logic. An equivalent circuit simulation has been presented in [48]. As shown in this example, a single MLUT-PGMB can be programmed to function as a TH gate. Likewise, multiple MLUT-PGMBs can be programmed and interconnected via a reconfigurable interconnection network to form a higher level circuit in nanoscale. This is similar to the hierarchal architecture presented in DR-ANRCA section. Another benefit of the MLUT-PGMB design is that it simplifies the reading procedure by assuring that the memristors at each crosspoints are required to be programmed either ON or OFF state (i.e., 1 or 0 in Boolean) so that the LUT could be setup. This eliminated the need of using the adaptive measurement algorithm proposed in [43]. Because the comparison of two different states is only necessary. Also, for any TH gate, the initial crosspoint (1,1) (i.e., all inputs are 0) is always programmed to 0 and the last crosspoint (8,4) (i.e., all inputs are 1) is always programmed to 1, therefore they are used as references to determine the value of a selected output.

The functionality of the proposed MLUT-PGMB's threshold and hysteresis behaviors were verified by implementing a structural model in VHDL. A timing simulation is also performed using the design automation tool. The simulation waveform is shown in Figure 2.12. Output F becomes '1' when ABC becomes '011' (i.e., threshold behavior), then maintains this value until ABC resets back to '000' (i.e., hysteresis behavior). The results demonstrate that the structural model's functionality accurately matches the logic of the $TH23$ gate.

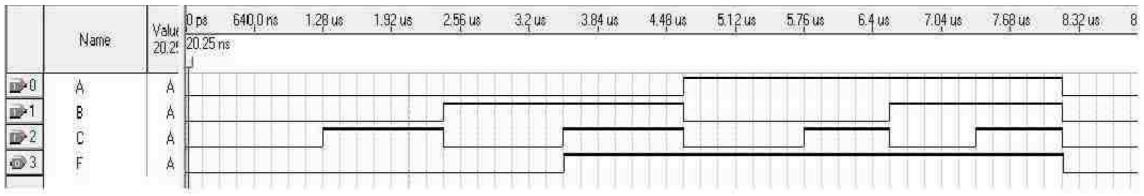


Figure 2.12. Waveform of TH23 gate.

2.7. DEFECT/FAULT ISSUES OF DR-MLUT

To address the high inherent defect density in the context of MLUT-ANRCA design, this study focuses on a MLUT-PGMB as a basic unit of MLUT-ANRCA to realize the proposed defect-tolerance methods. There are three states in which a defective crosspoint can be detected: nonprogrammable, stuck-at-1, or stuck-at-0. A crosspoint in a non-programmable state can not be changed to an ON or OFF state. A comparison method is used to detect whether or not a crosspoint is in this a state. This method utilizes the initial crosspoint (1,1) and the last crosspoint (8,4) as references for comparing a selected crosspoint. If the selected crosspoint does not match either of the reference values, it can be considered as a nonprogrammable crosspoint. This method is based on the assumption that the reference crosspoints are programmed properly. The crosspoints with stuck-at-0 or stuck-at-1 faults can be simply detected by programming opposite values on them and then utilize the proposed comparison method to determine whether it is reconfigurable.

Tolerance for these defective crosspoints can be attained by the use of redundancy, replacement, or a passive approach. Redundancy is the basic method in the fault-tolerance system, it involves the use of spare wires to reprogram copies of selected logic. The area provided by the MLUT-PGMB allows crosspoints on specific rows to be available for redundancy based on the number of inputs for the mapped TH gate. For example, consider the worst case scenario of a 10% defect rate. The left part of Figure 2.13 shows a defective MLUT-PGMB implementing a given $TH23$ function. The crosspoints (1,3), (3,4), and (5,2) are crossed out to indicate that there are defects. The right part of Figure 2.13 shows a repaired version of the same gate. Crosspoints (2,3), (4,4), (6,2) have been used to replace those defective crosspoints. The same method can be applied to other four inputs TH gates by adding extra wires to increase the number of rows/columns. Relatively high defect rates can be tolerated using this method, especially for the three/two inputs TH gates which require a lower number crosspoints to be programmed on a 8×4 MLUT-PGMB. Defect rates of 50% or even 75% can be tolerated without increasing the size of the MLUT-PGMB.

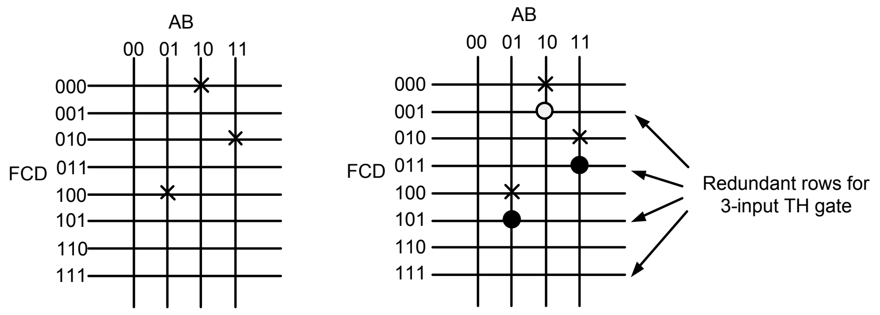


Figure 2.13. A defective MLUT-PGMB and a version repaired using the redundancy method.

The hysteresis function of NCL determines that the number of crosspoints programmed to 1 is more than the number of crosspoints programmed to 0, because the asserted output will become 0 only when all inputs become 0. Based on this observation, we can use a passive approach to deal with stuck-at-1 faults. If these faults occur at the crosspoints that are originally going to be programmed to 1, they can be ignored because the desired functionality can still be achieved. Figure 2.14 shows a bar graph that indicates the percentage of crosspoints that are programmed to 1 for each TH gate. It shows that more than 50% of the crosspoints on TH gates are programmed to 1. Thus, this method can be used to tolerate most of stuck-at-1 faults on MLUT-PGMBs. The same method can be applied to stuck-at-0 faults, however, the number of crosspoints that are programmed to 0 for each TH gate is relatively low. To deal with the faults that cannot be ignored, the wires can be rearranged by changing the switching function of the demultiplexer to tolerate them. This approach rearranges the programmed crosspoints to a position where the faults do not interfere with the functionality of the desired TH gate. Alternatively, redundant rows can be used to replace the defective row.

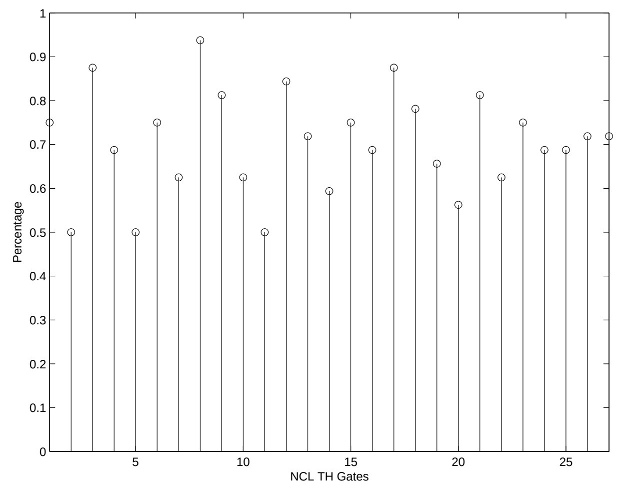


Figure 2.14. Percentage of crosspoints that are programmed to 1 for each TH gate.

2.8. COMPARISON BETWEEN DR-ANRCA AND MLUT-ANRCA

The two designs introduced in this dissertation present novel implementations of ANRCA that achieve the goal of eliminating the dependency of a global clock by implementing the NCL on nanowire crossbar architectures. The DR-ANRCA utilizes crosspoints as programmable diodes that create an AND/OR plane to implement TH gates. MLUT-ANRCA, on the other hand, uses configurable memristors to realize nanoscale lookup tables (LUTs) to represent each TH gate. The comparison of both in terms of structure, control method, and defect-tolerance analysis are introduced in this section to realize the pros and cons of each method.

The structure of a DR-PGMB consists of 6×11 nanowires to create a total of 66 crosspoints to implement any TH gate, whereas a MLUT-PGMB consists of 8×4 nanowires to create a total of 32 crosspoints to perform the same functionality. Clearly, MLUT-PGMB requires fewer crosspoints to provide the same functionality. Another benefit of the MLUT-PGMB structure is that it has a better utilization rate of crosspoints than DR-PGMB when implementing the same TH gate function. For example, to program a $TH23$ gate on both designs, a DR-PGMB requires 18 out of 66 crosspoints, whereas a MLUT-PGMB requires 16 out of 32. The utilization rate is 27.3% versus 50%. Figure 2.15 shows the comparison of crosspoints utilization rate between these two designs for all TH gates. This work also compares the two ANRCAs in terms of area, programming latency, and defect tolerance. The structure of these two designs are different. DR-ANRCA utilizes a combination of three or more demultiplexers to effectively control and program DR-PGMBs in a CLB. Each MLUT-PGMB needs two demultiplexers and one multiplexer to program a TH gate. Similar to DR-ANRCA, it can be managed by a switching matrix to communicate with other MLUT-PGMBs in a CLB. Therefore, a DR-ANRCA uses fewer demultiplexers to implement a CLB, making it relatively easier to be reconfigured than MLUT-CLBs do. According to the NanoPLA assembly parameters provided in [26], $10nm$ nanowire pitch is acceptable for assembling the crosspoints between each crossed nanowire and $3nm$ nanowire diameters have also been demonstrated. The area of a single DR-PGMB nano-array and a MLUT-PGMB nano-array is $9,044nm^2$ and $7,276nm^2$, respectively. As Figure 2.15 shows, for most of the TH gates that are implemented in MLUT-PGMB, the crosspoints utilization rate is 100% since all 32 crosspoints will be programmed to achieve its functionality, where the same TH gates require fewer crossponits (up to 50%) to be implemented on a DR-PGMB. Therefore, assuming the latency of programming a crosspoint is the same for both PGMBs (i.e., ts), the total time required to program a MLUT-PGMB would be considerably greater than that needed to program a DR-PGMB. Figure 2.16 demonstrates a scatter graph that plots the propagation time of programming each TH gate on both PGMBs. The order of TH gates on the x-axis are done in ascending order according to the number of inputs.

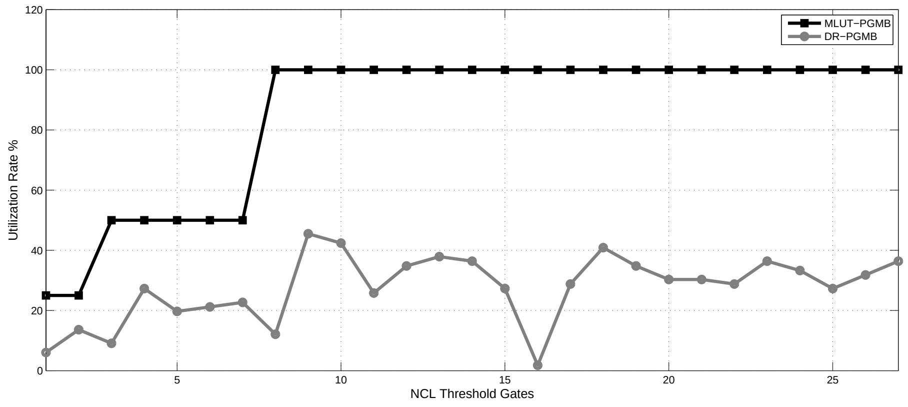


Figure 2.15. Crosspoints utilization comparison.

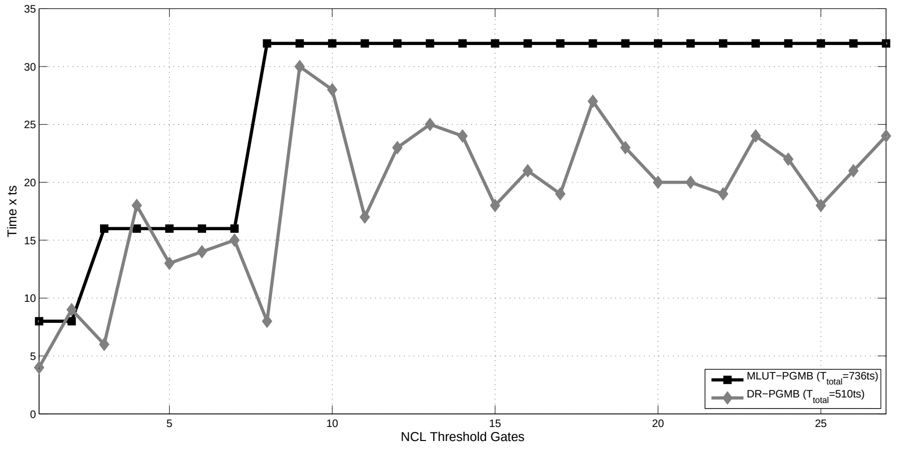


Figure 2.16. Programming time comparison.

Introducing redundancy is a common method that has been widely used in the field of fault-tolerance and reliability. Usually there are two types of redundancy methods: spatial redundancy and temporal redundancy [27]. This work utilized the spatial redundancy method for tolerating these defective crosspoints in the proposed ANRCAs. The cost of spatial redundancy is that larger areas will be consumed when the crossbar scales up. In this regard, MLUT-PGMBs are more flexible in the replacement of a defective crosspoints than DR-PGMBs are because the latter must rely on the AND/OR functions when using the redundancy crosspoints, and the defective crosspoints cannot be fully replaced without increasing the size of each PGMB. However, the defective crosspoints in MLUT-PGMB, especially when it is implementing a two- or three-input TH gate, can be ignored or replaced by adjacent crosspoints, as explained in section 2.7. The comparison made here are based on theoretical designs; a more detailed comparison can be made by evaluating the performance of diodes and memristors.

2.9. CONCLUSIONS

This section has presented two different implementations of ANRCA: DR-ANRCA and MLUT-ANRCA. Both of them are based on NCL that intrinsically eliminate many clock-related issues caused by complex clock distribution networks. Synthesizing nanowires is based on bottom-up fabrication paradigm. The integrated nano-systems prefer using a hierarchical structure over monolithic structure. The studies of ANRCAs start with its primitive configurable logic unit (PGMB), then three versions of CLB for reconfiguration, and finally an FPGA-like hierachical architecture. With all these features, ANRCAs are anticipated to have the advantages of improved manufacturability, scalability, modularity, and robustness.

Comparisons between different designs are discussed in terms of of area, programming steps, and latency. Due to the high inherent defect density in nanowire crossbar caused by the self-assembly fabrication techniques, the conventional fault-tolerance methods are not adequate for ANRCAs. This research have proposed some defect-tolerance and repair strategies for both ANRCAs, including FTA based post-configuration, permutation and commutative method, spatial redundancy, replacement, and passive strategy.

3. PROJECT II: NCL S-BOX DESIGN

3.1. BACKGROUND

In relation to the market of digital information security, crypto-hardware devices that have enhanced security measures while being energy efficient are in high demand. The growth of innovation for these devices can be seen in today's mobile phones and portable devices and computer/network security in industrial control systems [49]. In order to reach this demand of low powered devices with high security features, researchers generally focus around the actually cryptographic algorithm implemented in the hardware itself to encrypt and decrypt information. Thus, securing cryptographic devices against various side channel attacks (SCA) has become a very attractive research topic in recent years along with the developments of information technologies. SCAs explore the security information (i.e., secret key) by monitoring the emitted outputs from physical cryptosystems. These outputs include execution timing, power consumptions, electromagnetic leaks, and even thermal/acoustic emanations [50]. Accurate measurement and estimation of these outputs is the key point of a successful attack. The measurement should be based on the hardware gate-level approach rather than the software instruction-level estimation [51, 52, 53, 54]. Also, for the power consumption measurement, the focus would be the dynamic power consumption that is dissipated during the transistors switching rather than static leakage power consumption [55]. Advanced Encryption Standard (AES) was announced with the intention of being a faster and more secure encryption algorithm over others since its algorithm is comprised of multiple processes used to encrypt information with supports of up to 256 bits key and block sizes, making an exhaustive search impossible to check all 2^{256} possibilities. Usually the hardware AES implementation has a higher reliability than software since it is difficult to be read or modified by attackers and less prone to reverse engineering. Unfortunately, AES is still vulnerable to SCA [56, 57]. The published SCA include simple power analysis (SPA), differential power analysis (DPA), correlation power analysis (CPA) [50, 58], collision attack[59], and leakage power analysis (LPA) [60]. Among them, DPA and CPA are the most popular and effective attack that has been reviewed by numerous researchers on various crypto-systems during these years [50]. In the meantime, many countermeasures for resisting SCA attacks were proposed as well. Most of the countermeasures designed for hardware implementations of AES are based on securing the logics cells to balance the power consumption of the system and make it independent of the processing data. This process of adjusting the basic units of the system makes the overall design less vulnerable to attacks. The hardware implementation of AES essentially has higher reliability than software because it is difficult to be read or modified by the attackers and less prone to reverse engineering [61]. These countermeasures can be separated into two categories based on the framework of the circuit they are implemented on, synchronous and asynchronous.

3.2. EXISTING COUNTERMEASURES

The countermeasures for synchronous circuits include Sense Amplifier Basic Logic (SABL) [62], an improved two-spacer alternating dual rail circuit [63], Wave Dynamic Differential Logic (WDDL) [64], a dynamic voltage and frequency switching approach [65], masked logic styles [66, 67], using Fourier Transform [68], Random Switching Logic (RSL) [69] with its simplified version Dual-rail random-switching logic [70], and recently proposed Masked Dual-Rail Pre-charged Logic (MDPL) and its improved version [71, 72]. These works are centered around resisting DPA attacks and introduce methods on how to effectively reduce the impact of DPA attack. However, they are fundamentally based on synchronized circuits, which either requires a precise control of timing or suffer from some timing related issues, such as glitches [71, 73], hazard, and early propagation [73, 72, 74], which still could leak some side-channel information to the attackers.

Asynchronous circuits, on the other hand, have natural advantages in terms of SCA resistance. The clock-related information leakage can be either eliminated or significantly reduced, which extensively increases the difficulties of attack due to the lack of timing references. The countermeasures based on asynchronous circuits include Balanced delay insensitive method [75], GALS module [76], and 1-of-n data encoded speed independent (SI) circuit [77, 78]. However, the increased security does not come for free. The area required to implement them are potentially larger than their synchronized counterparts. The benefits in terms of total power consumption and speed are still questionable. In addition, some of the countermeasures are based on the EDA tool simulation results or theoretical analysis, which may not effectively prove that these methods could resist real SCA attacks experimentally.

From these existing countermeasures, the dual-rail encoding[63, 79], with the pre-charge method, spacers, or return to zero (RTZ) protocols are frequently used in both synchronous and asynchronous designs. The dual-rail encoding provides better data independence with the power consumption since the Hamming weights of each data set are the same. A RTZ, spacer, or pre-charge method is used to achieve the monotonic transition to enhance the security. Our proposed Null Conventional Logic (NCL) based S-Box design essentially matches all these important security properties: asynchronous, dual-rail encoding, and an intermediate state (i.e., NULL). Unlike other asynchronous designs, NCL adheres to the monotonic transitions between DATA (i.e., data representation) and NULL (i.e., control representation), which utilizes dual-rail and quad-rail signaling methods to achieve the delay-insensitivity [9]. This would significantly reduce the design complexity. With the absence of a clock, the NCL system is proved to reduce the power consumption, noise, and electromagnetic interference [80, 81]. Furthermore, this work has demonstrated that NCL could also resist SCA without worrying about the glitches and power supply variations [82]. This work provides an extension to what has been presented in [82]. Besides the DPA attack, a CPA attack has been also applied to both synchronous and NCL S-Box design to demonstrated that the proposed NCL S-Box is capable of resisting CPA attack as well.

Usually, there are four methods to conduct the power measurement experiments: 1) use computer-aided design (CAD) tools [61], 2) use regular FPGA board [64], 3) use the SASEBO-GII FPGA board, and 4) use a taped-out ASIC chip. The procedures of taping out a chip include the front-end verification using CAD tools and FPGA board. They are complicated, time consuming, and expensive. Therefore, in order to prove the proposed idea in a more effective way, the first three methods have been tried and the experimental results show that the third method is the most effective one among these three methods. The rational behind lies as follows: 1) while

CAD-tool-based simulation shows the synchronous S-Box design is indeed vulnerable to the DPA attacks, the DPA attacks could not be successfully implemented on the NCL S-Box by such simulation due to too much regularity in the simulated power traces. The reason is that the CAD tools approximate the simulation results. 2) There are many constraints on using regular FPGA board for this experiment. All the decoupling capacitors related to the core power supply should be removed to increase the chance of successful attacks. Also, a current probe is needed to measure the current consumed by the core of the FPGA chip. However, the bandwidth of current probes is usually lower than that of voltage probes, which might not be able to capture the high-frequency AC current variations caused by data transients. 3) A stable power supply is critically important for power analysis experiments. Therefore, using the SASEBO-GII FPGA board is the most effective way because it is designed for the purpose of SCA experiment and it solves all the issues that a regular FPGA board has. In summary, this work provides a general testing procedure to do SCA attacks on the SASEBO-GII FPGA board.

3.3. NCL AES S-BOX DESIGN

AES algorithm consists of a number of rounds that are dependent on the key size. For both cipher and decipher of AES algorithm, each round consists of linear operation (i.e., AddRoundKey, ShiftRows, and MixColumns steps) and non-linear operation (i.e., SubBytes step). SubBytes step is the first step of AES round. Each byte in the array is updated by an 8-bit substitution box (S-Box), derived from the multiplicative inverse over $GF(2^8)$. AES S-Box is constructed by combining the inverse function with an invertible affine transformation in order to avoid attacks based on mathematics. The S-Box is one the of most critical components in the implementation of AES hardwares. It consumes the majority of power and is also the most vulnerable component to SCAs. A block diagram of AES S-Box is shown in Figure 3.1. The block diagram of multiplicative inversion over $GF(2^8)$ component where MM is modular multiplication, and XOR is exclusive-or operation [83].

The hardware implementation of AES S-Box adapted in this research follows the combinational logic circuit architecture presented in [83], but uses NCL gates

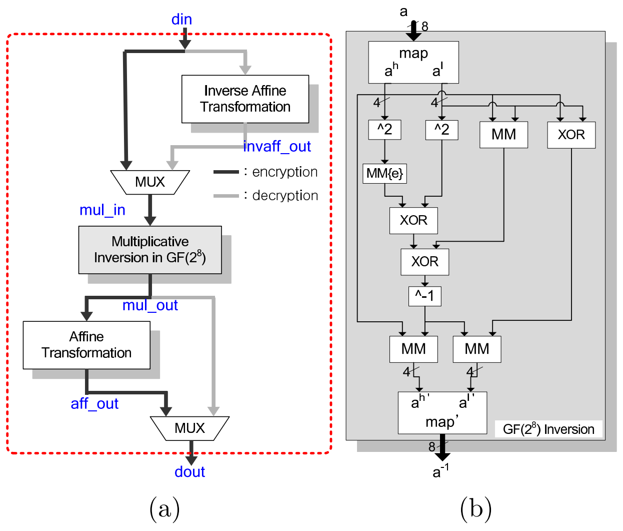


Figure 3.1. Block diagram of a combinational S-Box with encryption and decryption datapaths.

instead of Boolean logic gates. The affine transformation and inverse affine transformation components follow a series of Boolean equations given in Table 3.1, where i and q represents the 8-bit input and output, respectively. Both transformations require many XOR gates. The multiplicative inversion in $GF(2^8)$ follows the procedure shown in Figure 3.1 part (b). 1) Map operation converts the 8-bit input into elements of $GF(2^4)$ (i.e. a_h and a_l); 2) Calculate the square of a_h and a_l. It should be noticed that multiplication in $GF(2^4)$ is done by multiplying the polynomial $a_h(x)a_h(x)$ followed by a modular reduction; 3) A series of multiplication and XOR operations were implemented to extend the field $GF(2^4)$ to the field $GF(2^8)$. To implement this conventional S-Box using NCL, the XOR, AND, and MUX operation in dual-rail NCL gates are required [61].

NCL has a total of 27 threshold gates to realize various logic functions. In order to achieve the input-completeness and observability, it is important to choose appropriate threshold gates. For example, in the design of a two-to-one multiplexer, according to the Karnaugh map in Figure 3.2(a), the sum-of-product (SOP) functions can be simplified as follows:

Table 3.1. Boolean equations for affine transformation and inverse affine transformation components.

$q = aff\_trans(i)$	$q = aff\_trans^{-1}(i)$
$q_0 = (i_0 \oplus i_4) \oplus (i_5 \oplus i_6) \oplus (i_7 \oplus 1)$	$q_0 = i_2 \oplus i_5 \oplus i_7 \oplus 1$
$q_1 = i_1 \oplus i_5 \oplus i_6 \oplus i_7 \oplus i_0 \oplus 1$	$q_1 = i_0 \oplus i_3 \oplus i_6$
$q_2 = i_2 \oplus i_6 \oplus i_7 \oplus i_0 \oplus i_1$	$q_2 = i_1 \oplus i_4 \oplus i_7 \oplus 1$
$q_3 = i_3 \oplus i_7 \oplus i_0 \oplus i_1 \oplus i_2$	$q_3 = i_2 \oplus i_5 \oplus i_0$
$q_4 = i_4 \oplus i_0 \oplus i_1 \oplus i_2 \oplus i_3$	$q_4 = i_1 \oplus i_3 \oplus i_6$
$q_5 = i_1 \oplus i_5 \oplus i_2 \oplus i_3 \oplus i_4 \oplus 1$	$q_5 = i_2 \oplus i_4 \oplus i_7$
$q_6 = i_6 \oplus i_2 \oplus i_3 \oplus i_4 \oplus i_5 \oplus 1$	$q_6 = i_0 \oplus i_3 \oplus i_5 \oplus 1$
$q_7 = i_7 \oplus i_3 \oplus i_4 \oplus i_5 \oplus i_6$	$q_7 = i_1 \oplus i_4 \oplus i_6$

$$Z^0 = A^0S^0 + S^1B^0; \tag{8}$$

$$Z^1 = A^1S^0 + S^1B^1; \tag{9}$$

After modifying both functions for input-completeness, new SOP functions are obtained:

$$Z^0 = A^0S^0(A^0 + A^1)(B^0 + B^1) + S^1B^0(A^0 + A^1)(B^0 + B^1); \tag{10}$$

$$Z^1 = A^1S^0(A^0 + A^1)(B^0 + B^1) + S^1B^1(A^0 + A^1)(B^0 + B^1); \tag{11}$$

and both of them can be mapped to a NCL circuit with a TH24comp gate, a THand0 gate, and a TH22 gate. The finalized NCL MUX logic diagram is shown in Figure 3.2(b).

Likewise, two TH24comp gates can be used to implement an XOR logic function. A THand0 and a TH22 gate are used to implement an AND logic function. The logic diagrams are shown in Figure 3.3.

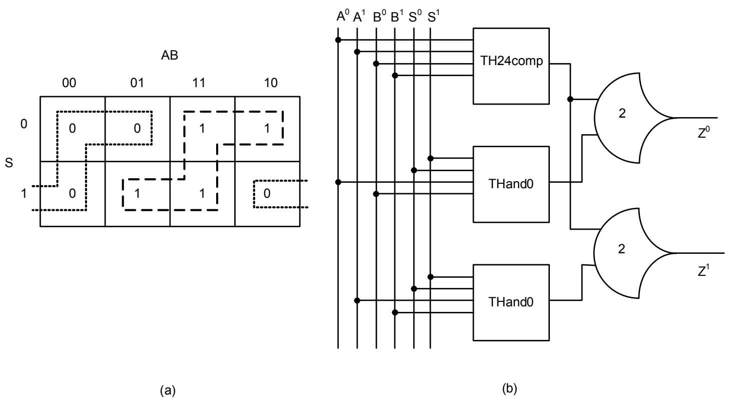


Figure 3.2. An input-complete NCL multiplexer design.

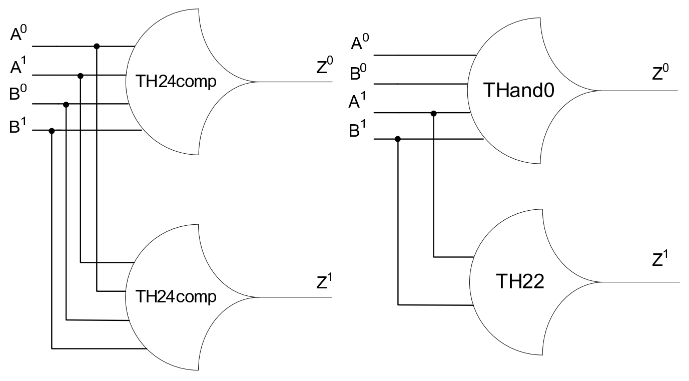


Figure 3.3. Input-complete NCL XOR (left) and NCL AND (right) functions for the proposed NCL S-Box.

3.4. FUNCTIONAL VERIFICATION

The proposed NCL S-box has been implemented in VHDL and simulated with ModelSim by Mentor Graphics. By referring to the waveform shown on Figure 3.4, the initial value of the input and output is NULL and DATA0, respectively, as previous input registers are reset to NULL and output registers are reset to DATA0. As soon as the resets fall down to 0, K_o from the output register becomes 1 and K_i from the input register connected to K_o becomes 1. As K_i rises, the input is changed to the waiting input signal, 0101010101010101 in dual-rail signaling which means 00000000 in binary

(0x00 in hexadecimal). The output arrives later due to the propagation delay, the output becomes 0110100101011010 in NCL which means 01100011 in binary and 0x63 in hexadecimal. The input signals are cumulative from 0 to 255, increment by 1 in each cycle. As shown in Figure 3.4, the input signal increases from 0x00 to 0x02 and the corresponding output signals are 0x63, 0$x7C$, and 0x77, respectively. The results are matching with the standard S-Box announced by the NIST [84]. As every bit of the output signal changes from NULL to DATA , K_o falls to 0, which means that the output register has received the proper output DATA signal. Every single component (i.e. affine and inverse affine transformation, multiplicative inversion) has been verified separately. All the input/output date were extracted using the VHDL textio package, then, a scripting program was written to verify each of the output date, ensuring they function correctly. Table 3.2 shows the encryption and decryption simulation results for both synchronous S-Box and NCL S-Box using 10 arbitrary sample inputs, 5 for encryption and 5 for decryption, respectively. On the NCL S-Box output column, the results are shown as 16 bits, which are the extended dual-rail signals. For example, for input 158, the NCL S-Box output is 0101010110011010, and this dual-rail encoded data word is equivalent to 00001011 in binary which is equal to the output of the conventional synchronous S-Box.

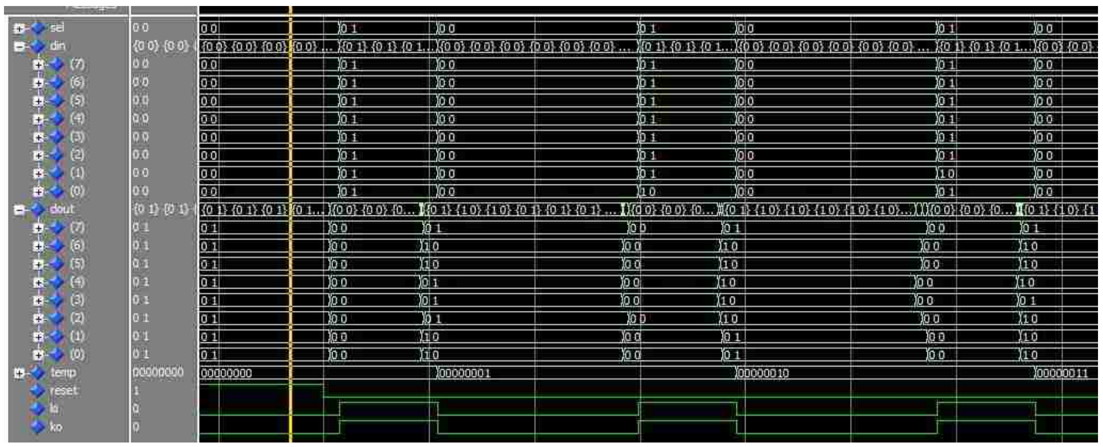


Figure 3.4. ModelSim waveform for the proposed NCL S-Box with input signals changing from 0d to 3d.

Table 3.2. Simulation results for 10 arbitrary samples from conventional synchronous S-Box and the proposed NCL S-Box.

Simulation Results			
Mode	Input	Output	
		S-Box	NCL S-Box
Encrypt	9	00000001	0101010101010110
	26	10100010	1001100101011001
	106	00000010	0101010101011001
	122	11011010	1001101001101001
	158	00001011	0101010110011010
Decrypt	32	01010100	0110011001100101
	51	01100110	0110100101101001
	156	00011100	0101011010100101
	185	11011011	1010011010011010
	203	01011001	0110011010010110

Besides ModelSim simulation, the Field-Programmable Gate Array (FPGA) based verification has also been done on both synchronous S-Box and NCL S-Box designs. Figure 3.5 is an oscilloscope screen image of the NCL S-Box hardware implementation with the embedded key 11010100. The 16 digital signals represent the dual-rail outputs of NCL S-Box. The blue waveform (Channel 1) is the voltage across a shunt resistor of the FPGA core. It is used to measure the current of the FPGA, which represents the power consumption of the FPGA core. The green (Channel 2) and purple (Channel 3) waveforms are used to set the triggers to ease data alignment for SCA programming. The falling edge of the green signal means that all 256 input data have been processed and the last one would be Binary 11111111. Since the key is 11010100, after the bit-wise XOR function, the actual input goes to the S-Box would be 00101011. According to the standard S-Box table[84], the corresponding output is 11110001, which is 1010101001010110 in NCL or $0xF1$ in hexadecimal as shown in the Figure 3.5. Following that, the input signal is incremented to 00000000 and the S-Box input becomes $00000000 \bigoplus 11010100 = 11010100$, which generates the corresponding output 01001000 (i.e. $0x48$). Similarly, hexadecimal numbers $0x03$ and $0xF6$ shown in the Figure 3.5 can be derived as well. All 256 inputs with different

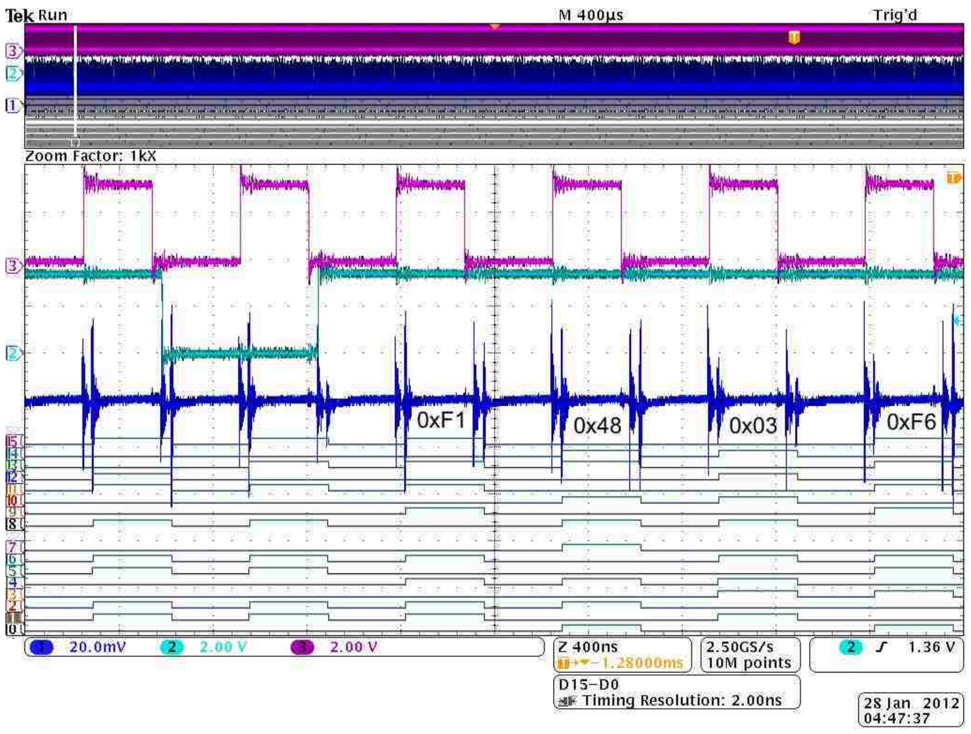


Figure 3.5. Power waveform of NCL S-Box design.

keys have been verified during the power analysis programming using Matlab. The correct behavior of the function is the prerequisite for a successful power attack.

3.5. POWER MEASUREMENT EXPERIMENT

FPGA provides an effective platforms for fast and highly reconfigurable prototyping of integrated circuits. Various instrumentation and measurements methods were implemented or demonstrated on FPGAs [85, 86]. A power measurement methodology for FPGA was also presented to break down the power consumption of different elements inside the logic [87], allowing attackers to get detailed information inside the circuit. This section goes into detail on the procedures used for the author's experiments and the results obtained from the power measurement associated to the comparison of the synchronous AES S-Box. The total power consumption has been measured using the MG tools in [61] and Table 3.3 presents the measurement results on both designs. It shows the proposed NCL S-Box has 22% to 26% lower total power consumption than synchronous S-Box. Figure 3.6 shows the SASEBO-GII board [88] that is used as the basic platform in this experiment. There are two FPGA cores in this board that can be utilized; the AES S-Box circuit is implemented in the cryptographic circuit and the configuration circuit is programmed into the configuration FPGA. The purpose of separating these two circuits is to prevent the power trace of the configuration circuit from interfering with the power trace of the cryptographic circuit, so that the measurements of making/resisting power analysis attacks can be done fairly. Two different power analysis attacks were conducted on both synchronous and NCL designs: Differential Power Analysis (DPA) and Correlation Power Analysis (CPA).

Table 3.3. Power simulation results for synchronous AES S-Box and NCL AES S-Box using Accusim and AdvanceMS.

Temperature: 27°C VDD: 1.8V	Synch S-Box	NCL S-Box
Total Power Dissipation (Watts) - Accusim+Eldo	2.474E-08	1.934E-08
Total Power Dissipation (Watts) - AdvanceMS	2.686E-08	1.981E-08

3.6. DIFFERENTIAL POWER ANALYSIS

DPA is a traditional power analysis attack that was first proposed by Kocher et al. [89] in 1999. DPA is a more effective and sophisticated form of a power analysis attack than SPA is. DPA utilizes a statistical method to analyze the relationship between the key and the power consumption, even when the power consumption variations are so small that SPA could not deal with [89]. Once the logic circuits were programmed into the FPGAs, measurements on the power waveform were performed using an oscilloscope (i.e.,Tektronix MSO 4054 for this study). Figure 3.5 indicates that most of the power consumption from the S-box occurs during the transition between the DATA and NULL process in NCL logic. A trigger signal was designed

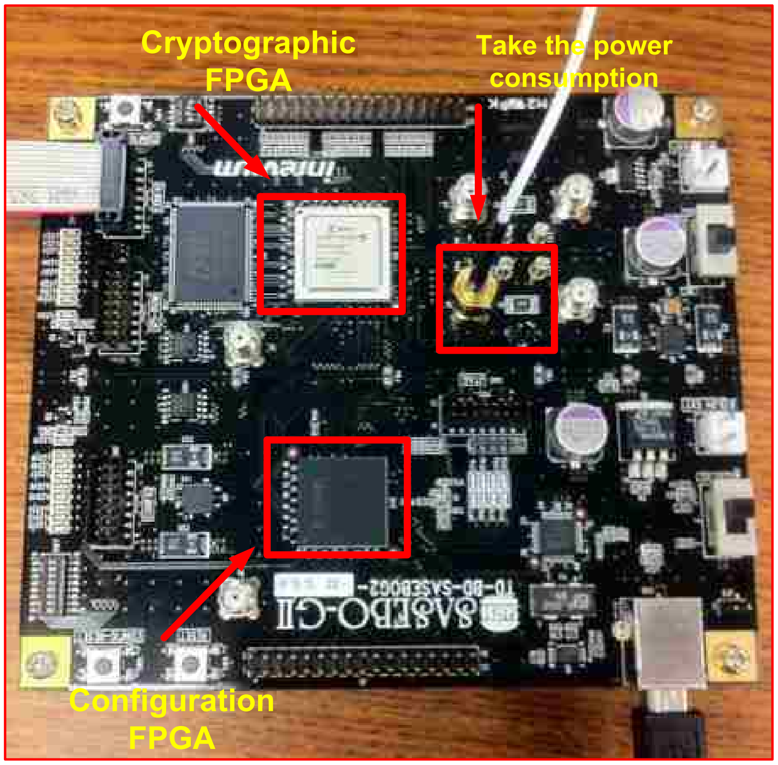


Figure 3.6. Specified side channel attack standard evaluation FPGA board (SASEBO-GII).

to check the cycle of operation encompassing 256 ascending inputs. Following that, experimental data were collected between two trigger pulses. The experimental data consisted of 1 million points of the FPGA core current (representing power consumption) and the digital output data along a time span of $400 \mu s$. The resolution time is thereby 4ns. Thus, 165 points have been sampled for each input data.

The sampled data are then processed in Matlab to implement a DPA attack. There are 256 outputs of data, each of which is related to a specific input datum. Figure 3.7 illustrates the steps to this statistical analysis. In this figure, the solid lines represent the path of the security key traveling through the system, while the dash lines represents the hypothetical keys moving through the system. The process of DPA consist of the following steps:

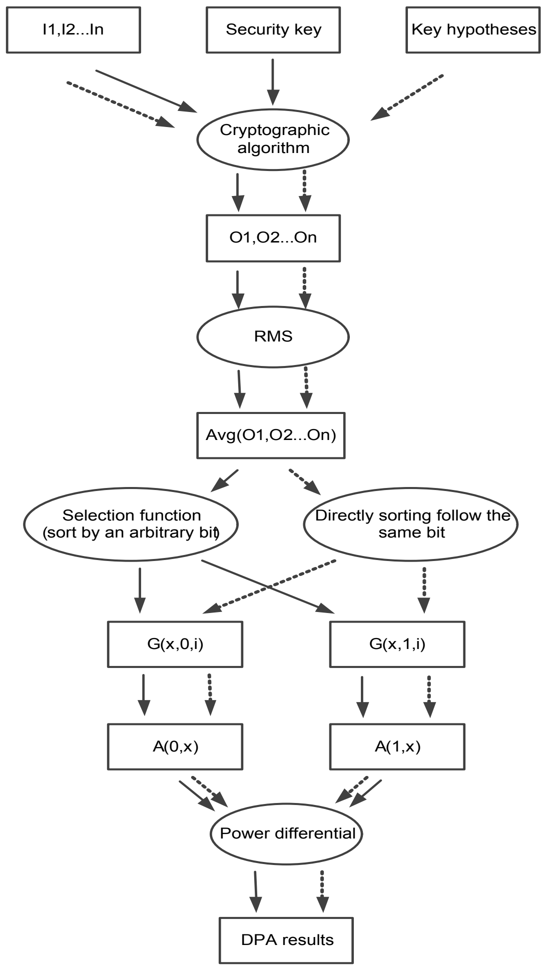


Figure 3.7. Block diagram illustrating the DPA attack steps.

1. The average current corresponding to each output data are calculated in Matlab using root mean square (RMS) algorithm.

$$A_p = \sqrt{\frac{1}{N_{sample}} \sum_{k=0}^{N_{sample}} x_i^2} \tag{12}$$

A_p is the average of N_{sample} sampled current for output data i;

2. Two groups are defined according to the select function when the secret key is used (Detailed explanation could be found in[89]).

$$A_{0,x} = \frac{1}{N_0} \sum^{G_{x,0,i}} A_{p,i} \tag{13}$$

$$A_{1,x} = \frac{1}{N_1} \sum^{G_{x,1,j}} A_{p,j} \tag{14}$$

$G_{x,0,i}$ and $G_{x,1,j}$ are two sets of average current, respectively, and they are grouped according to an arbitrary bit in the output.

3. The average power in each group is calculated using the results of the first two steps and then the difference in average power between these two groups is the result of the DPA process.

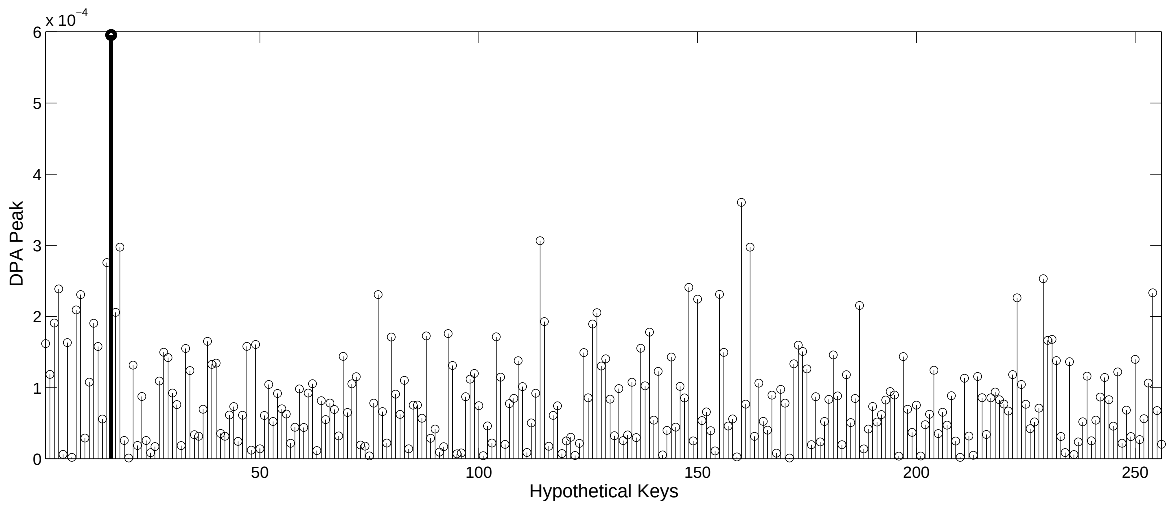
$$d_x = |A_{0,x} - A_{1,x}| \tag{15}$$

$A_{0,x}$ and $A_{1,x}$ are the average currents of 0-group and 1-group, respectively, for a hypothetical key x; d_x is the difference between the average currents.

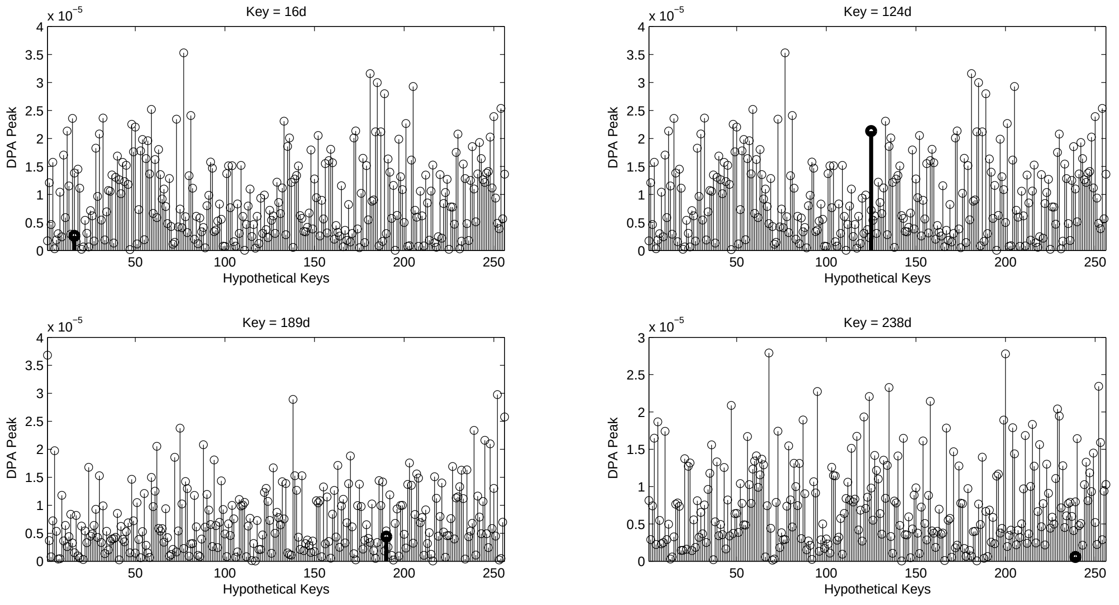
This DPA process has been implemented on both synchronous S-Box and NCL S-Box with 256 keys. Figure 3.8 part (a) is the result of the DPA attack on the synchronous S-Box design; the correct key (16d) is clearly identifiable since the power peak is much higher than other hypothesized keys. Figure 3.8 part (b) shows the result of the DPA attack on the same S-Box design using NCL. The attacker can not identify the correct key since its power peak is not prominent compared to others. Due to the limited amount of space, only three other randomly selected keys are shown here. The DPA attack results show that the selected keys can not be identified from other assumption keys. Therefore, the proposed NCL S-Box design significantly improves security against DPA attacks.

3.7. CORRELATION POWER ANALYSIS

A more effective way to find the secret key of a cryptographic device is to analyze the correlative relationship between the plain-text/cipher-text and instantaneous



(a)



(b)

Figure 3.8. Power peaks for various key assumptions for a DPA attack on synchronous S-Box design and NCL S-Box design.

power consumption of the device. Pearson correlation coefficient is the most familiar measure of dependence between two quantities. Correlation power analysis (CPA) is an improvement of DPA [90, 72, 91, 92]. In CPA, a power model is used to predict the power consumption in terms of hypothetical keys and various input/output data; the predicted power is then compared with the measured power using a correlation coefficient algorithm [93, 94, 95]. If the hypothetical key is the secret key, its correlation coefficient with the measured power will be significantly higher than other incorrect hypothetical keys.

Before introducing the detailed steps of CPA, an experimental setup (hardware configuration) is described below, which is important for obtaining a sufficient number of power traces to execute a successful attack. Similar to the DPA experiment, CPA was implemented in the S-Box design, which is downloaded into the SASEBO-GII FPGA board. The core current of the cryptographic FPGA was measured through a shunt resistor in the core power supply and it represents the instantaneous power consumption.

The output of the S-box was sampled for the attacks attempts. A one bit digital signal from the configuration FPGA is also sampled. Its falling edge represents the instant when an input datum is fed to the cryptographic FPGA. In this way the current measurement (analog signal) can be easily aligned with the output of the S-box (digital signals). Such alignment is crucial for successful CPA attacks[93]. All analog and digital signals were measured using a Tektronix 500 MHz-bandwidth oscilloscope MSO4054. The total number of data for one set of measurement is 10 million samples. The main clock for the FPGA board is 24 MHz and an input data are sent to the cryptographic FPGA every 16 clock cycles. Therefore, the rate of data is $24/16 = 1.5$ MHz and the time span for each data is $\frac{1}{1.5\text{ MHz}} = 666.67$ ns. The sampling rate of the oscilloscope is 2.5 GHz for analog signals and 500 MHz for digital signals, which are the maximum settings for MSO4054 oscilloscope. Therefore, the length of instantaneous power measurement is $\frac{2500\text{ GHz}}{24\text{ MHz}} \times 16 = 1666$ samples for each of datum

Unlike the DPA experiment, where the power consumption of all possible keys (including the hypothetical keys and the secret key) are required to collect from the FPGA board, CPA only requires the power consumption of the cryptographic device to be collected while the secret key is embedded within it. The power consumption of other hypothetical keys could be predicted using a power model; a Hamming Weight

(HW) model is used in this experiment [93]. The steps for revealing a secret key using CPA is described as follows:

1. Retrieve a set of expected cipher-texts (**O**) (the output of the S-Box in this case) that are generated by a set of plain texts (**I**) and a set of hypothetical keys (**K**). Let I_i be an element in **I** ($i \in [1, nd]$ and nd is the number of plain texts) and K_j be an element in **K** ($j \in [1, nk]$] and $nk = 256$ is the number of possible key for an 8-bit S-box). Define the S-box as a function name $O(\cdot) = \text{SBox}(\cdot)$. Then an element in **O** matrix is calculated as

$$O_{i,j} = \text{SBox}(I_i \text{xor} K_j). \tag{16}$$

By applying (16), the **O** matrix is given by

$$Output_{nd,nk} = \begin{bmatrix} O_{1,1} & O_{1,2} & \cdots & O_{1,nk} \\ O_{2,1} & O_{2,2} & \cdots & O_{2,nk} \\ \vdots & \vdots & \ddots & \vdots \\ O_{nd,1} & O_{nd,2} & \cdots & O_{nd,nk}. \end{bmatrix} \tag{17}$$

2. Apply a power consumption mode and build a hypothetical power consumption matrix **HW** from the cipher-text matrix **O**. In this work, the Hamming weight of a given cipher-text (the output of the S-box) is used as the power consumption model.

$$HW_{i,j} = \text{HammingWeight}(O_{i,j}). \tag{18}$$

By applying (18), the **HW** matrix is given by

$$HW_{nd,nk} = \begin{bmatrix} HW_{1,1} & HW_{1,2} & \cdots & HW_{1,nk} \\ HW_{2,1} & HW_{2,2} & \cdots & HW_{2,nk} \\ \vdots & \vdots & \ddots & \vdots \\ HW_{nd,1} & HW_{nd,2} & \cdots & HW_{nd,nk}. \end{bmatrix} \tag{19}$$

3. Measure the real power consumption of the cryptographic device for nd number of inputs. As explained above, the instantaneous current consumption during DATA transitions is the most important information for CPA. Therefore, this experiment needs the transients power between the outputs rather than the steady power. The alignment between the measured power and its corresponding data is another critical fact for a successful CPA attack. As a result, each data window is defined as from 100 samples before a transient to 1500 sample points after a transient (a total of 1600 samples per data window). The transient signal is the falling edge from the configuration FPGA, as discussed before. In this way the digital sampling data can be accurately aligned with the measured analog power. According to previous discussion, each datum has 1666 sampled real power consumption points. There are 66 points discarded because they are all in steady-state so that they play an insignificant role in CPA attacks. Therefore, the windows length for each datum $nt = 1600$. A total number of $nd \times nt$ sampled points are re-organized into a real power consumption matrix **RP**. At the end of this step, a real power consumption matrix $RP_{nd,nt}$ of S-Box should be generated, which is given by

$$RP_{nd,nt} = \begin{bmatrix} RP_{1,1} & RP_{1,2} & \cdots & RP_{1,nt} \\ RP_{2,1} & RP_{2,2} & \cdots & RP_{2,nt} \\ \vdots & \vdots & \ddots & \vdots \\ RP_{nd,1} & RP_{nd,2} & \cdots & RP_{nd,nt}. \end{bmatrix} \tag{20}$$

4. Compare the **HW** matrix with the **RP** matrix using the correlation coefficient formula to find out the correct key that has the highest correlation value. There are a total number of $nk * nt$ correlation coefficients to be calculated. In each round of calculation, take one column X_j ($j \in [1, nk]$) of the **HW** matrix

$$X_j = \begin{bmatrix} HW_{1,j} & HW_{2,j} & \cdots & HW_{nd,j} \end{bmatrix}^T \tag{21}$$

take one column Y_t ($t \in [1, nt]$) of the **RP** matrix

$$Y_t = \begin{bmatrix} RP_{1,t} & RP_{2,t} & \cdots & RP_{nd,t} \end{bmatrix}^T \tag{22}$$

and calculate an element of the correlation coefficient matrix **Corr** as

$$Corr(X_j, Y_t) = \frac{Cov(X_j, Y_t)}{\sqrt{Var(X_j) \cdot Var(Y_t)}}. \tag{23}$$

The correlation coefficient is defined as equation (23). It is used to estimate the relationship between two vectors X_j and Y_t. $Cov(X_j, Y_t)$ represents the covariance between X_j and Y_t. $Var(X_j)$ and $Var(Y_t)$ represent the standard deviations of X_j and Y_t, respectively. The Pearson correlation coefficient estimator r between X_j and Y_t could be written in equation (24) for a series of n measurements, where $\bar{x}$ and $\bar{y}$ represent the mean value of X_j and Y_t, respectively.

$$r = \frac{\sum_{i=1}^{n} (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 \cdot \sum_{i=1}^{n} (y_i - \bar{y})^2}} \tag{24}$$

By applying (23), the correlation coefficient matrix is give by

$$Corr_{nk,nt} = \begin{bmatrix} Corr_{1,1} & Corr_{1,2} & \cdots & Corr_{1,nt} \\ Corr_{2,1} & Corr_{2,2} & \cdots & Corr_{2,nt} \\ \vdots & \vdots & \ddots & \vdots \\ Corr_{nk,1} & Corr_{nk,2} & \cdots & Corr_{nk,nt}. \end{bmatrix} \tag{25}$$

Figure 3.9 shows the results of CPA attacks on the synchronous S-Box design and NCL S-Box design, respectively. The X-axis represents the length of a data window and Y-axis represents the correlation value for each key hypothesis. There would be total $2^8 = 256$ possibilities from 0 to 255 for a 8-bit key. Thus, there are 256 correlation values in Y-axis for every sampled point in the data window. Correct Keys are plotted in black, while the other key assumptions are plotted in grey. The highest correlation peaks happened during the time of data transition. Therefore, for synchronous S-Box design, the highest correlation value occurs at $200 \div 2.5e9 = 80(ns)$, which is around the $\frac{80 \text{ ns}}{666.67 \text{ ns}} \times nt = 191$th data point for a data window (as shown in Figure 3.9(a)). This result shows that the CPA attack on the synchronous

S-Box is successful. For the NCL S-Box design, there are two significant correlation peaks in Figure 3.9(b) due to the transition between two data units being processed as DATA1 to NULL, then NULL to DATA2. Using the similar method mentioned previously, the two data transition points are around the 300th data point and the 1200th data point for a data window. The correct key (247d) is plotted in black, while other key hypothesis are plotted in grey. Figure 3.9(a) shows that the correct key (247d) is clearly revealed by observing the highest correlation value, while it can't be observed in Figure 3.9(b) because it is buried by other keys. Similarly, three other random keys are selected for the CPA attacks, but none of the keys have the highest correlation coefficient. Therefore, the NCL S-Box design is resistant to CPA attacks.

3.8. NUMBER OF TRACES

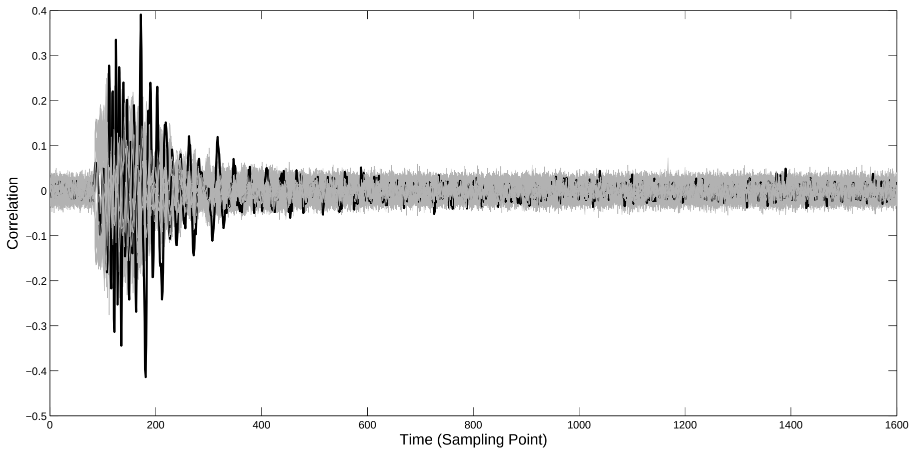
Besides proper data alignment, another important factor is to get enough number of power traces to conduct a successful DPA or CPA attack. The more traces are measured, the more elements are in the **RP** matrix, the measurement noises would cause less affect, and the correlation between the measurement power and power model would be more precise. However, too many power traces would increase the time to gather data and the time to process the data. Therefore, it is interesting to analyze the effect of number of power traces on the success of CPA attacks. For a given set of number of trace $ntr \in [1, 500]$ and a given key $k_j \in K$, In the **HW** matrix, take the column corresponding to the key k_j for ntr number of traces,

$$X_j^{'} = \begin{bmatrix} HW_{1,j} & HW_{2,j} & \cdots & HW_{ntr,j}. \end{bmatrix}^T \tag{26}$$

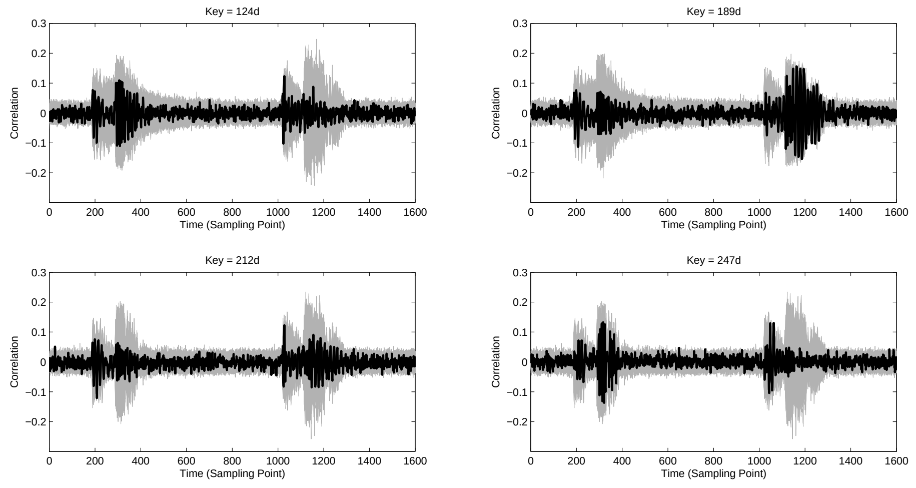
In the **RP** matrix, take the column when transients happen (column 191 for synchronous S-box and column 300 for NCL S-box, respectively, as discussed previously)

$$Y_t^{'} = \begin{bmatrix} RP_{1,tran} & RP_{2,tran} & \cdots & RP_{ntr,tran}. \end{bmatrix}^T \tag{27}$$

And calculate the correlation coefficient between vectors $X_j^{'}$ and $Y_t^{'}$. Repeat the above process for all possible combinations of ntr and k_j, the results are shown in Figure 3.10 part (a) and part (b), for synchronous S-box and NCL S-box, respectively.



(a)

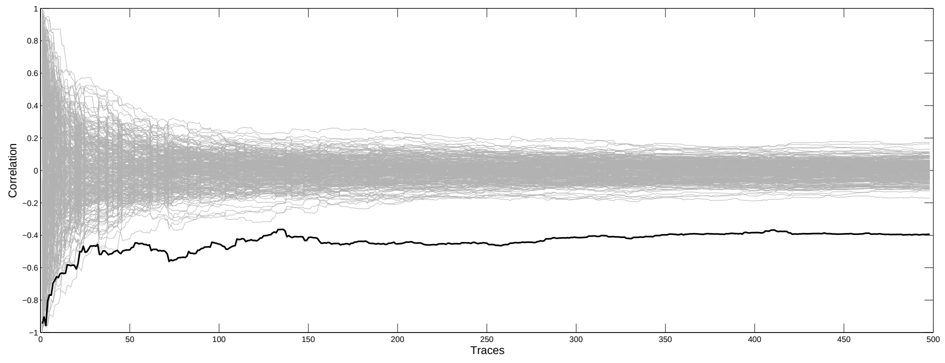


(b)

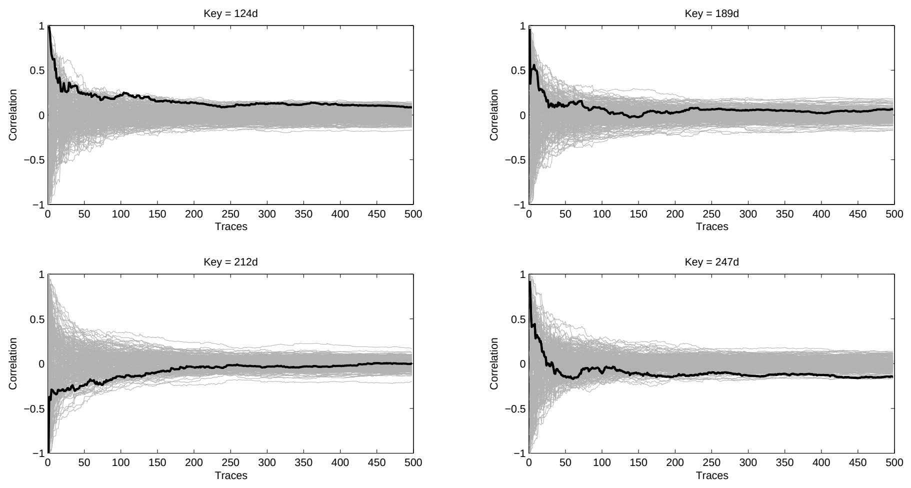
Figure 3.9. CPA result for synchronous S-Box design and NCL S-Box design.

Figure 3.10 provides the threshold of how many traces are needed for attackers on both designs at the instant of data transients.

The X-axis represents the number of traces and Y-axis represents the correlation value at the time of data transients. The correct key (247d) is plotted in black, and other keys are plotted in grey. For the synchronous S-Box, when the number of power traces is small (ie. less than 25), the correlation coefficient of the correct key



(a)



(b)

Figure 3.10. The number of traces needed for a successful CPA attack.

is buried by the correlation coefficient of other keys. This means that CPA would fail when the number of power traces is too small. When there is enough number of power traces (ie, more than 50), the correlation coefficient of the correct key begins to stand out from others. CPA attack in this case will be successful. However, for the NCL S-Box design, the correct key does not stand out from other keys even the number of traces pass 500. Therefore, it demonstrated that NCL makes the attack more difficult.

3.9. CONCLUSIONS

In this section, a hardware implementation of the proposed low power SCA resistant asynchronous S-Box design for the AES crypto-system is revealed to be successfully resisting DPA and CPA attacks. The asynchronous S-Box design is based on a self-timing logic referred to as NCL, which supports beneficial properties for resisting DPA/CPA: clock-free, dual-rail signal, and monotonic transitions. These beneficial properties make it difficult for attackers to decipher secret keys embedded within the cryptographic circuit of the FPGA board. Utilizing the two FPGAs included in the SASEBO-GII board, the configuration and cryptographic functions are able to be performed separately to ensure that the power trace measurements for the analysis attacks do not interfere with each other. Experimental results of the original design against the proposed S-Box revealed that the asynchronous design decreased the amount of information leaked from both DPA and CPA attack. Results also revealed that the proposed design showed benefits of flatter power peaks and 22% to 26% lower total power consumption during regular operation.

The proposed DPA and CPA attacks procedure based on power measurement is comprehensive and general, not limited to SASEBO-GII board. It can be revised and used for studying SCAs on other devices.

4. PROJECT III: STOCHASTIC BITSTREAMS COMPUTATION

4.1. BACKGROUND

Unlike most digital circuits that are designed to transfer definite inputs into definite outputs. (i.e., Boolean, integer, and real value). Stochastic logic operates on probabilistic bitstreams, where the signal is encoded by the probability of obtaining a one or a zero from a given input signal [96]. An illustration of a stochastic bitstream is shown in Figure 4.1 [97]. A real value x = 3/8 ($0 \leq x \leq 1$) is represented as a bitstream with three bits are one.

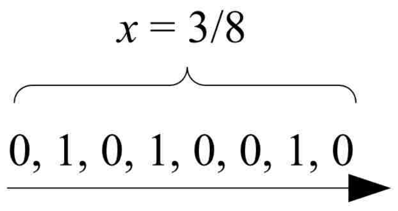


Figure 4.1. A stochastic bitstream with value 3/8.

Stochastic logic can be adapted to any combinational circuits, with stochastic bitstreams as inputs and/or outputs. Therefore, logical computation in the deterministic Boolean number domain is transformed into probabilistic computation in the real number domain. Figure 4.2 [97] shows an AND gate with bitstreams as inputs and outputs, the probabilities of obtaining a one in the input streams are 0.8 and 0.5. The probability of obtaining a one in the output stream is 0.4, which is 0.8×0.5.

The benefits of using stochastic logic are:

1. A complex operations in Boolean logic can be designed in a simple way using stochastic logic. For example, the multiplication operation can be implemented using a single AND gate as shown in Figure 4.2[97]. In contrast, conventional digital multiplier design needs a much complex design. More complex functions

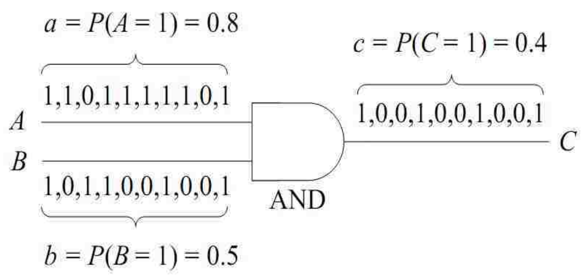


Figure 4.2. An AND gate to implement multiplication.

such as the Taylor expansion of an exponential function and the square root function can also be implemented using stochastic methodology [98, 99].

2. Stochastic logic circuits are highly tolerant to errors in signal processing where small fluctuations can be tolerated but large errors are catastrophic [100]. For example, if a single bit occasionally flipped into its complementary value (i.e., logic one becomes logic zero or vice-versa), the probability of occurrence of zeros and ones are not substantially changed. But if the same thing happened to a conventional circuit, bit flips can be a serious problem.

3. Stochastic representation of data could be either serial streaming on a single wire or in parallel on a bundle of wires. The parallel bitstreams can be synthesized into nanowire crossbar arrays [101]. The high fault-tolerance is also very important for nanoscale computation.

4. The highly randomness property of stochastic logic is helpful for cryptographic devices against power analysis attacks and thus, to enhance hardware security. This work has addressed this advantage by implementing an 8-bit AES S-Box with stochastic bitstreams.

However, things always come with tradeoffs. Comparing with conventional binary radix encoding, the stochastic logic encoding generates a large delay in obtaining the computation results [96]. A binary encoding can represent 2^n numbers with only n bits, while to present real numbers with a resolution of 2^{-n}, numbers of the form $\frac{x}{2^n}$ for integers x between 0 and 2^n, a stochastic encoding requires a stream of 2^n

bits. Stochastic logic uses probabilistic bitstreams to represent the binary logic state. Thus, the random fluctuation will generate some errors during the logic computation. In Figure 4.2, if the input stream A changes to 0,1,0,1,1,1,1,1,1 (a=0.8), while input stream B remain the same, the output bitstream will become 0,0,0,1,0,0,1,0,0,1, which is 0.3, not 0.4 any more. Therefore, the 100% accuracy in stochastic bitstreams is difficult to achieve.

4.2. STOCHASTIC CRYPTOGRAPHIC DEVICES

Utilizing the concept of stochastic bitstreams, a novel approach to enhance the security of cryptographic devices against power analysis attacks has been presented in this work. The proposed structure has good compatibility that could be used in different combinational logic core of encryption standards. To demonstrate the methodology, a MATLAB simulink based simulation was designed to show the idea. Following that, a VHDL hardware implementation of an 8-bit AES S-Box was built to verify the resistance of power analysis attacks.

4.2.1. MATLAB Simulink Structure. For a given combinational logic (i.e., a two-input AND gate followed by a two-input OR gate in this case), an encoder with a random number generator is needed to generate the bitstreams. The output of the combinational logic is bitstreams, therefore a decoder is needed to convert the bitstreams to binary numbers for logic verification. The system structure of this MATLAB Simulink based simulation is shown in Figure 4.3.

The simulation model includes three encoders for each input signal, a given combinational logic, and a decoder for output. The block diagram of an encoder is shown in Figure 4.4. It consists of a build-in random number generator (RNG) with sampling rate of 100 KHz, a decision function, and a comparator. RNG generates a random number (RN) in the range of 0 to 1. The expression of decision function is $P = 0.25 + 0.5 \times x$, where x represents the input. P is the design weight; $P = 0.25$ when $x = 0$, meaning that the probability of 0s in the bitstreams is 25%; similarly, the probability of 1s in the bitstreams is 75%. The output becomes 1 if $RN \leq P$, and 0 if $RN > P$. The generated bitstreams can be seen in Figure 4.5. The top two signals (i.e., and_{in1}, and_{in2}) represent the two input bitstreams of the AND gate, and the bottom signal (i.e., and_{out}) is the output bitstreams of the AND gate.

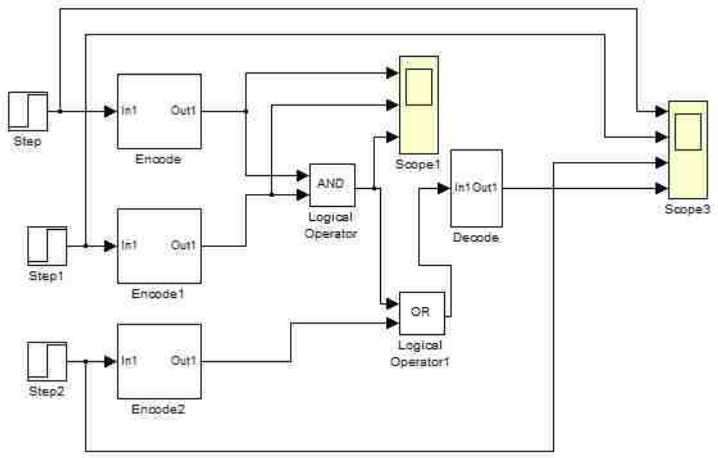


Figure 4.3. Stochastic implementation of a combinational logic in simulink.

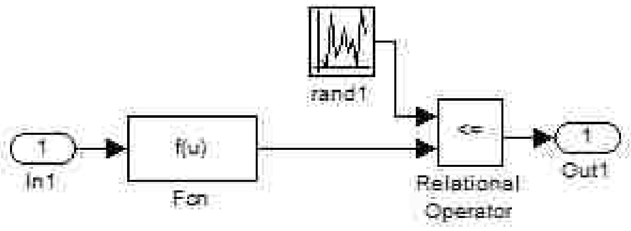


Figure 4.4. Encoder block.

Figure 4.6 shows the block diagram of a decoder. The input of the decoder is the output of stochastic logic blocks, which is in the form of stochastic bitstreams. The up-counter counts the number of 1s in the bitstream and the probability of 1s is calculated by dividing the counter value by the length of the bitstream window. The comparator determines the output state, by comparing the calculated possibility with 0.5. If the calculated probability is larger than 0.5, the output is considered as 1 in conventional logic. On the other hand, it is considered as 0 in conventional logic if the probability is less than 0.5.

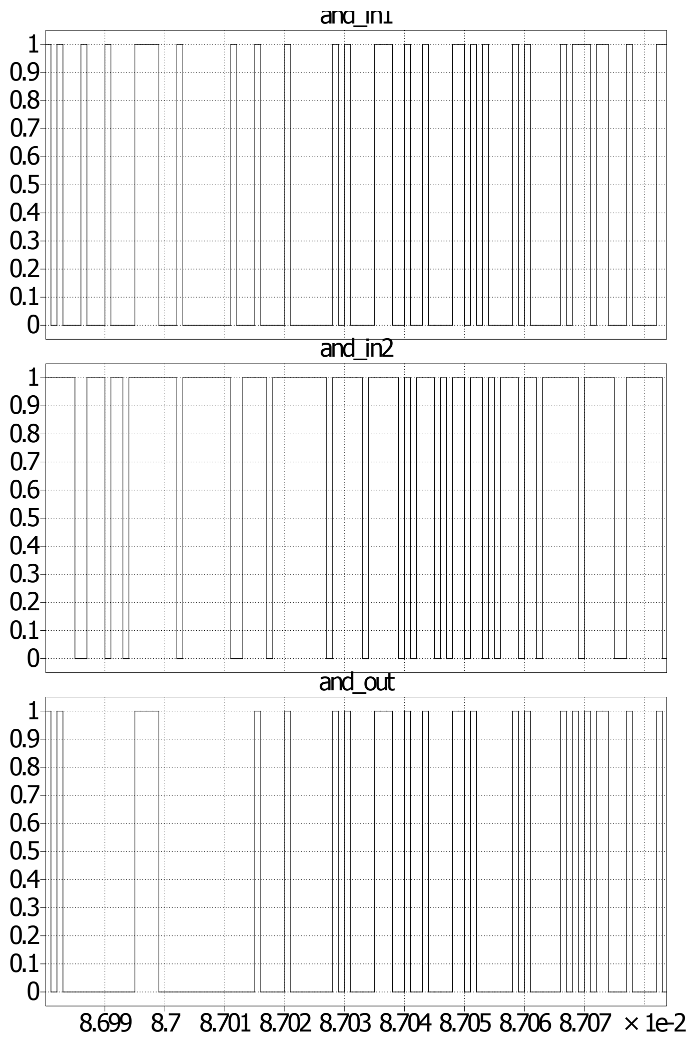


Figure 4.5. Generated bitstreams waveform of AND function.

The final output of the MATLAB simulink based simulation is shown in Figure 4.7. It shows that output is exactly matching with this combinational logic function $OUT \Leftarrow (In1 \bullet In2) + In3$;

This simulation model has demonstrated the idea of utilizing encoder and decoder to convert signals between stochastic bitstreams and conventional binary signals for a given combinational logic. However, in order to do the power analysis attack, a synthesizable hardware model of a cryptosystem need to be designed, so the power traces can be measured using the SASEBO-GII FPGA board.

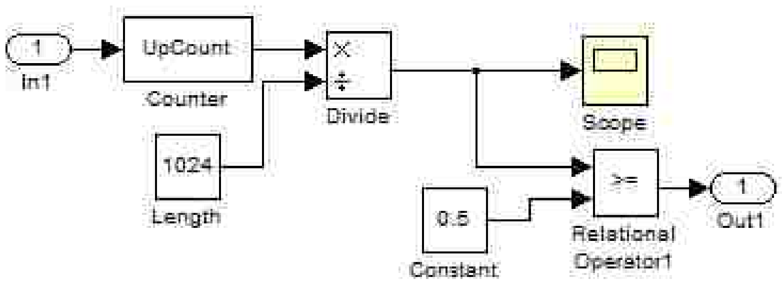


Figure 4.6. Decoder block.

4.2.2. Hardware Implementation of Bitstream Based AES S-Box. This subsection presents the hardware implementation of an 8-bit AES S-Box that is based on stochastic bitstreams. As mentioned earlier, S-Box is the most critical component of AES cryptographic devices because it is the most vulnerable component to SCA. Its Boolean logic implementation has been presented in Section 3.3. In a stochastic logic system, encoders and decoders are needed. The system diagram of the proposed structure is shown in Figure 4.8. In this work, the author defines HIGH and LOW to represent two possible states in a binary encoded bit in the input and output, while 1s and 0s to represent bits in the stochastic bitstreams. For clarity, in Figure 4.8, the solid lines represent the conventional logic states, while the dash lines represent the stochastic bitstreams.

The input (i.e., plaintext) and output (i.e., ciphertext) of the proposed design are in standard boolean logic. As Figure 4.8 shows, there are eight encoders, which generate stochastic bitstreams for each bit of the input data. The inputs of encoders are one bit from the input data and a 32-bit fixed-point random number (RN), which is ranging from 0 to 1. The random numbers are generated by combining the design of linear feedback shift register (LFSR) and cellular automata shift register (CASR) based pseudo-random number generators. This combined Random Number Generator (RNG) has better bit independence, longer cycle length, and enhanced unpredictable randomness [102]. The equation of encoders is

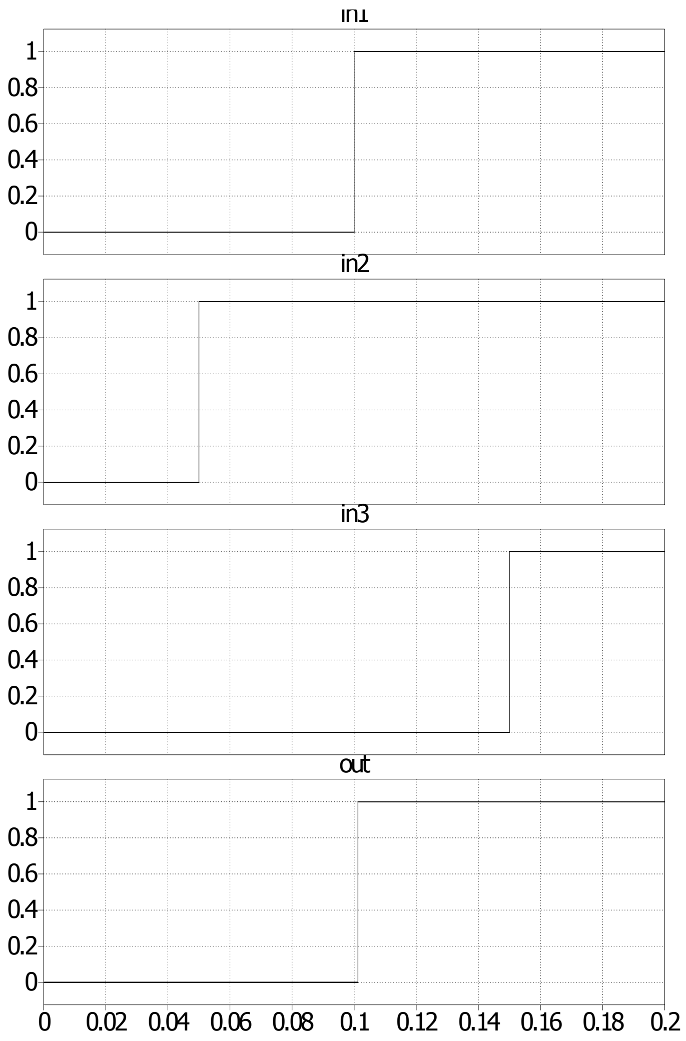


Figure 4.7. Final output waveform.

$$E_o(i) = \begin{cases} 1, & (x = HIGH \land P_1 \geq RN) \lor (x = LOW \land P_0 \geq RN) \\ 0, & (x = HIGH \land P_1 < RN) \lor (x = LOW \land P_0 < RN) \end{cases} \tag{28}$$

where x represents the input bit, P_1 and P_0 are the decision weight of the encoders, which dictate the probability of 1s and 0s in the bitstreams, respectively, when input is HIGH. Note that $P_1 + P_0 = 1$ ($P_1 > 0$, $P_0 > 0$). The distribution of P_1 and P_0 defines the decision weight of encoders, which affects the performance of the proposed

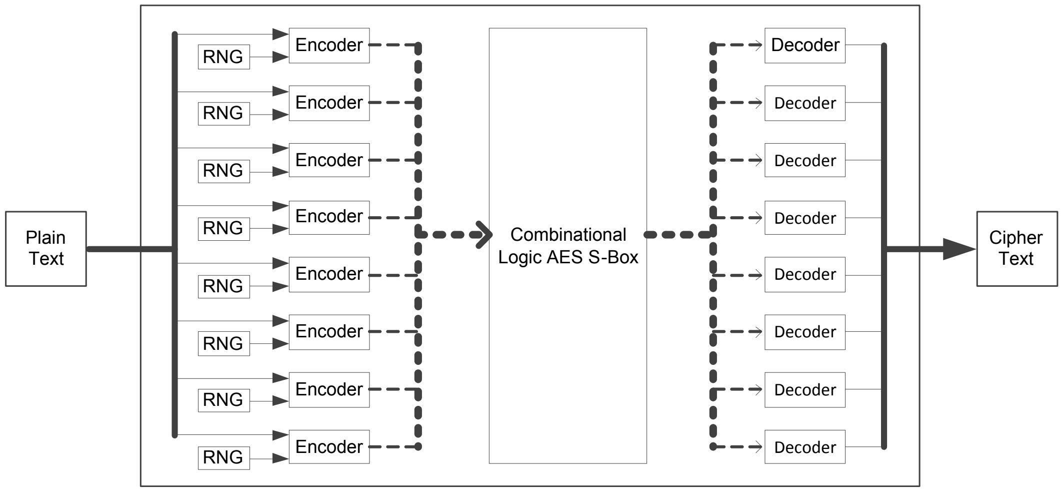


Figure 4.8. System diagram.

structure. The outputs of each encoder (E_o) are stochastic bitstreams that has the probability of P_1 to be '1' when its input is HIGH and the probability of P_0 to be '1' when its input is LOW. RN represents the random number that is generated by the RNGs.

The outputs of all encoders combine together to form an 8-bit bitstream (i.e., $E_o(0)$ to $E_o(7)$), contain information of the plaintext and are fed into a combinational logic S-box. The outputs of the S-box are the encrypted information, which is also in bitstreams. Therefore, decoders are needed to obtain the ciphertext, which count the number of 1s to determine whether the present bitstream represents HIGH or LOW logic. The equation of decoder is

$$D_o(i) = \begin{cases} HIGH, & \frac{N_{a=1}}{N_{clk}} \geq 0.5, \\ LOW, & \frac{N_{a=1}}{N_{clk}} < 0.5, \end{cases} \tag{29}$$

where $N_{a=1}$ is the number of 1s in the input bitstreams of a decoder and N_{clk} is the number of bitstream bits for an input logic state. D_o is the output of decoder. The following sections will discuss two aspects of the proposed stochastic logic S-box: (1) functionality and (2) resistance to power analysis attacks.

4.3. LOGIC VALIDATION

Stochastic logic injects randomness and uncertainty into the data flow, which helps in resisting power analysis attacks, but errors are inherently inevitable due to probabilistic nature of it. For cryptographic devices, 100% encryption/decryption accuracy is important. Thus, the accuracy of the implemented combinational logic should be analyzed and verified. In general, three factors may affect the accuracy of stochastic logic devices: the length of bitstreams for a given logic state, the decision weight of the encoders, and the repetition times of an input.

The proposed design has been implemented in VHDL and simulated using ModelSim. The output data are exported to a text file using the VHDL textio package, and then, compared with the correct output responses using a script. In this way, errors can be located and the error rate can be calculated as well. An intuitive method to decrease the likelihood of error is to repeat the input data. A given input datum is repeated by N_r times. The decoder side takes N_r number of output data and takes the one that appears most the times as the output. Given a total number of M input data, the error rate E_r is defined as $E_r = \frac{M_{wrong}}{M}$, where M_{wrong} is the number of error outputs, which are defined as the output of the decoder is different from the expected output value, among M input data. Figure 4.9 shows numerous error rate curves obtained by varying decision weight, bitstream length and repetition times. For each bitstream length, the error rate increases as P_0 increases. Increase in the difference between P_0 and P_1 means the resulting bitstreams exhibit less randomness. $P_0 = P_1 = 0.5$ means complete randomness (i.e., the bitstreams contain no information), while $P_0 = 0$, $P_1 = 1$ means no randomness (same as the conventional binary logic). In Figure 4.9, more randomness results in a higher error rate. The longer the bitstream for an input state is, the lower the error rate is. The more repetition times for an input, the lower the error rate is.

Based on the simulation results, there are several patterns lead the proposed structure to achieve near 100% accuracy. For example, two possible patterns are: 1)

Decision weight of $P_0 = 0.1$, $P_1 = 0.9$ with bitstream length of 512 or more, and 2) $P_0 = 0.05$, $P_1 = 0.95$ with bitstream length of 256 or more.

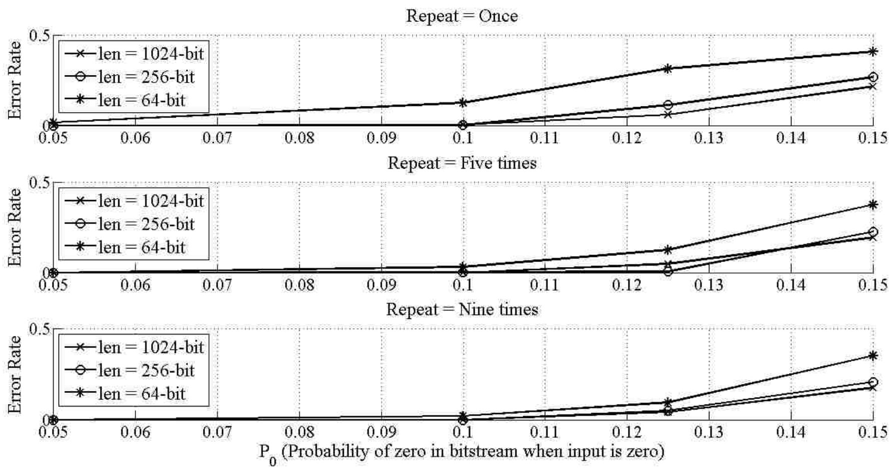
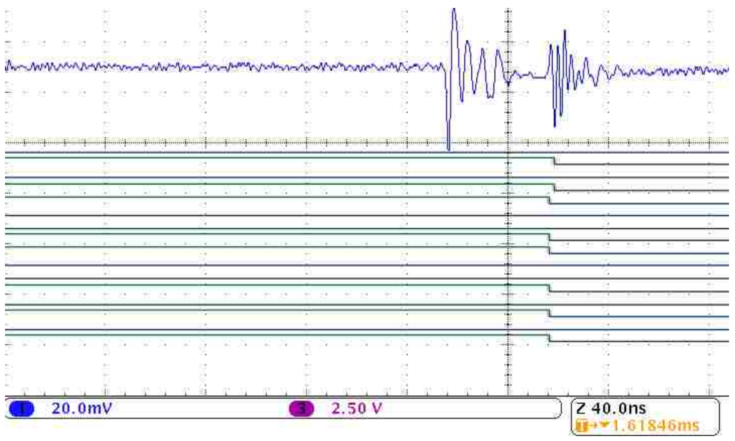


Figure 4.9. Error rate of stochastic bitstream S-box.

4.4. CPA RESULTS

Besides ModelSim simulation, the proposed stochastic bitstream S-Box has been also verified using a specified SCA standard evaluation FPGA board (SASEBO-GII). It includes two FPGA cores to implement cryptographic logic and configuration logic separately. So the power traces from cryptographic core would not be affected by configuration circuit. Figure 4.10 compares the power traces between regular S-box (Top) and stochastic S-box (Bottom) design. It shows that switching power is much less in the stochastic S-box design.

Among various SCA methods, CPA is considered to be more advanced and effective. CPA calculates the correlation coefficients between an estimated power model and the real measured power consumption. Since the power consumption of an integrated circuit can be estimated by the Hamming Distance (HD) between the current state and its consequential state [73]. The estimated power model is the Hamming Weight (HW) matrix of all outputs for all possible hypothesis keys.



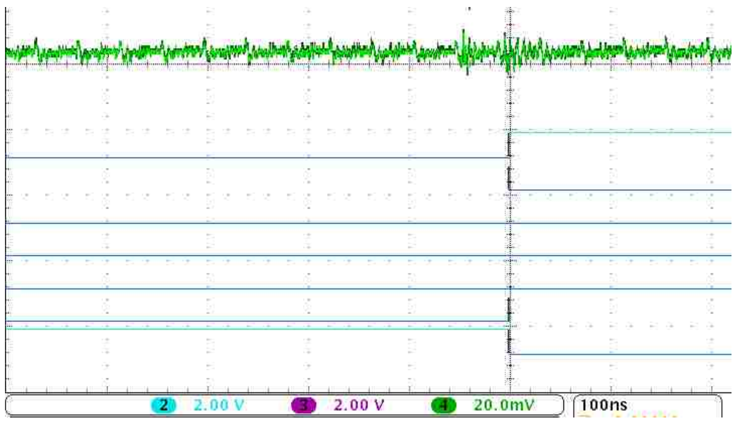


Figure 4.10. Power traces comparison between regular S-box (Top) and stochastic S-box (Bottom) design.

Once the correlation coefficients between the HW matrix and the measured power consumption matrix are calculated. The trace with the highest correlation coefficient is chosen to be the correct secret key. A detailed procedure of carrying a CPA attack has been reported in [103].

Figure 4.11 shows the result of an attempted CPA attack on the proposed design. The X-axis represents the length of a data window and Y-axis represents the correlation value for each key hypothesis. There would be total of 256 possibilities for an 8-bit key. The correct key (63d) is plotted in black, while other hypothesis keys are plotted in grey. The correct key is buried by the other keys. There is no correlation coefficient trace that is significantly higher than the others. Therefore, it is statistically infeasible to reveal the correct secret key in the given case. This indicates that the stochastic bitstream S-box design is resistant to the CPA attacks. The same CPA has conducted to a regular Boolean S-box design in [103], the attack was found to be successful. The proposed stochastic bitstream method can enhance security against power analysis attacks without the need to completely re-design the core logic functional blocks. Therefore, the method in this work can be applied to many other devices when they require enhanced security.

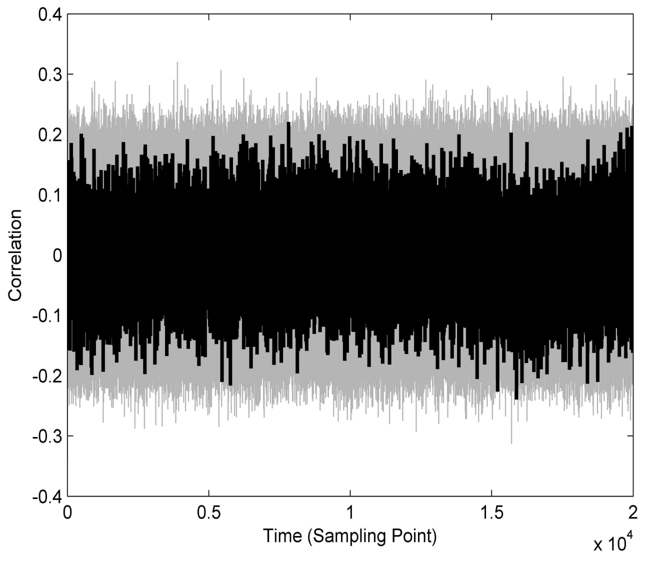


Figure 4.11. CPA results of stochastic bitstream S-box.

4.5. CONCLUSIONS

This section presents a novel approach to enhance the security of cryptographic devices against the power analysis attacks. The proposed approach uses stochastic logic, which utilizes probabilistic bitstreams to represent cryptographic information. The stochastic bitstream is generated by encoders with combination of CASR and LFSR based random number generators. The input and output signals of the core cryptographic component are the stochastic bitstreams and their randomness makes it difficult to carry out power analysis attacks. The output bitstreams are converted back to the conventional binary logic signals by decoders to ensure compatibility with other logic functions. The functionality of the proposed design has been verified using an 8-bit AES S-box design. The effects of decision weight, bitstream length, and input repetition times on error rates have been also studied. Experimental results obtained from a SASEBO-GII crypto-hardware evaluation board shows that the proposed approach enhances the resistance to against the CPA attack by successfully protecting the hidden key.

5. SUMMARY AND CONCLUSIONS

5.1. SUMMARY OF CONTRIBUTIONS

The objective of this dissertation is to study and analyze two applications of NCL and to explore more benefits of asynchronous NCL circuits and related system. NCL is an asynchronous logic with the feature of delay-insensitivity and dual-rail logic representation. Various NCL circuits have been designed with the purpose of improving power consumption and reducing noise and EMI. However, NCL designs require a much larger area than conventional Boolean logic designs, which might not be optimal for the applications like potable digital devices. Therefore, in order to fully demonstrate the merits of NCL, and to locate the appropriate applications for it, this dissertation introduces two applications.

The first one is NCL-based ANRCAs for nano-scale computing. With the absent of clock distribution network, ANRCAs are anticipated to make the nanowire crossbar design much more flexible and fault tolerant for manufacturing. The author has proposed two implementations of ANRCA, DR-ANRCA and MLUT-ANRCA, both of which have the advantages of eliminating clock-related issues, good scalability, and high reliability. Starting with primitive configurable logic unit (PGMB), a number of different reconfigurable CLBs, the fundamental building blocks of FPGA-like hierarchical architectures, have been designed and analyzed. The author also has compared different design schemes in terms of area, programming steps, and latency. Finally, defect-tolerance and repair strategies have been presented and discussed. All these together provide the backbones for the future development of nano-scale asynchronous logic, especially reconfigurable devices.

The second contribution of this dissertation is NCL-based cryptographic devices for enhanced security against SCAs. An 8-bit AES S-Box, the core components of AES cryptographic devices, is used as demonstration. The NCL-based asynchronous S-Box design has a number of features that help improve the resistance to SCAs, such as clock-free, dual-rail signals, and monotonic transitions. Such novel

S-Box designs has been implemented in VHDL and tested in both software simulation and hardware experiment. The detail procedure to carry out two popular SCAs, DPA and CPA, is provided. It has been demonstrated that the proposed NCL-based S-Box has the ability to resist various SCAs, and to have lower total power consumption, by comparing with its synchronous counterparts. As more people become concerned with cyber-security recently, the contribution in this section presents a security enhancement approach in the circuit design level.

Security of cryptographic devices would also be improved by using stochastic logic, which uses probabilistic bit streams to represent cryptographic information. This dissertation proposes a encoder/decoder scheme to use probabilistic bit streams without re-writing everything from scratch. The core component, in stochastic bit streams, has the feature of randomness to make SCAs very difficult. Meanwhile, the proposed design ensures compatibility with other logic functions. The author has studied the effects of a number of parameters on the performance of the proposed design scheme. Hardware demonstration on an 8-bit AES S-box confirms the enhancement against CPA attacks.

5.2. FUTURE WORK

The future work includes but is not limited to the following topics:

1) One topic is to develop an experimental prototype of the proposed ANRCAs using existing nanowire fabrication strategies. As discussed in Section 2, two approaches are possible, including bottom-up or hybrid bottom-up/top-down approaches [104, 105]. DR-ANRCA, MLUT-ANRCA, and some other design techniques developed in this dissertation have the potential in the application of the prototype nanowire-based integrated circuit. Silicon Nanowires (SiNWs) has been extensively studied by many researchers [106] because silicon is the dominant material of the existing semiconductor industry. On the other hand, Germanium Nanowires (GeNWs) has also been studied to implement high performance tunnel diodes recently [107], which could be a good candidate to implement the proposed DR-ANRCAs. The studies of memristor materials are updated frequently with different features [47]. The state-of-the-art in memristors uses titanium dioxide (TiO2) as resistive material, which is sandwiched between platinum electrodes. A silicon-based memristor

was proposed in [108]. ANRCAs experimental prototype can be developed using different materials. Then, the device characterization data would be collected for the performance comparison.

2) Nanowire-based integrated circuit, because of its bottom-up structure, requires major innovation in Computer-Aid Design (CAD) tools. An automated design optimization tool could be developed for the FPGA-like hierarchical ANRCAs. The tool should be able to maximize the utility of PGMBs, optimize programming steps, and take care of the inherent fabrication defect issues.

3) The NCL-based S-Box design can be extended to a full 14 rounds of 256-bit AES design. The first step would be to test the full NCL AES on the FPGA board. And then a NCL-based AES core processor prototype could also be in the process of taping out. Various SCAs are applied to the ASIC to verify its resistance against SCAs. It would be interesting to study and to analyze the practical issues to do successful SCAs on ASIC-based chips. It is also necessary to study the effect of ASIC layout on the chance of successful SCAs.

4) Another enhanced security strategy is to involve a Spatial and Temporal Random Dynamic Voltage Scaling (STRDVS) technology, which can provide additional resistance to SCAs and to further reduce the power consumption [109]. STRDVS can be implemented inside the NCL-base crypto-processor with different voltage levels. Scaling down the supply voltage will have impacts on the chip performance, such as speed and fault tolerance. Therefore, a series of numerical analysis is needed to determine the boundary of voltage scaling range. And the issue of fault-tolerance in high-speed low-voltage digital integrated circuits has become attractive recently. It is interesting to see how fault-tolerance calculation is related to NCL-based crypto-processors.

5) The proposed scholastic bit streams based encoder/decoder scheme needs further optimization to eliminate the errors generated by randomness. The regular error detection and repair strategy is no longer applicable to cryptographic applications due to the fact that the errors in the ciphertext are unpredictable. An automatic scholastic analysis tool could be developed to find the proper distribution between scholastic logic and conventional logic functions.

BIBLIOGRAPHY

[1] K. Fant and S. Brandt, “Null convention logic^{TM}: a complete and consistent logic for asynchronous digital circuit synthesis,” *International Conference on Application Specific Systems, Architectures and Processors*, pp. 261–273, 1996.

[2] K. M. Fant and S. A. Brandt, “Null convention logic: A complete and consistent logic for asynchronous digital circuit synthesis,” in *Proceedings of the IEEE International Conference on Application-Specific Systems, Architectures, and Processors*, p. 261, 1996.

[3] K. M. Fant and S. A. Brandt, “Null convention logic system,” *US Patent (US 5305463)*, 04 1994.

[4] S. C. Smith, “Speedup of null convention digital circuits using null cycle reduction,” *Journal of System Architecture*, vol. 52, no. 7, pp. 411–422, 2006.

[5] J. McCardle and D. Chester, “Measuring an asynchronous processor’s power and noise,” *Proceedings of the Synopsys User Group Conference*, 2001.

[6] S. K. Bandapati and S. C. Smith, “Design and characterization of null convention arithmetic logic units,” *Microelectron. Eng.*, vol. 84, pp. 280–287, Feb. 2007.

[7] S. C. Smith, “Design of a null convention self-timed divider,” in *The International Conference on VLSI*, vol. 1, pp. 447–453, 2004.

[8] L. Zhou and S. Smith, “Speedup of a large word-width high-speed asynchronous multiply and accumulate unit,” in *52nd IEEE International Midwest Symposium on Circuits and Systems*, pp. 499 –502, Aug. 2009.

[9] S. Smith, “Design of an FPGA logic element for implementing asynchronous null convention logic circuits,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 6, pp. 672 –683, 2007.

[10] V. Satagopan, B. Bhaskaran, W. K. Al-Assadi, S. C. Smith, and S. Kakarla, “DFT techniques and automation for asynchronous null conventional logic circuits,” *IEEE Transactions on Very Large Scale Integr. Syst.*, vol. 15, pp. 1155–1159, October 2007.

[11] S. C. Smith, “Speedup of null convention digital circuits using null cycle reduction,” *Journal of System Architecture*, vol. 52, pp. 411–422, July 2006.

[12] S. C. Smith, R. F. DeMara, J. S. Yuan, D. Ferguson, and D. Lamb, “Optimization of null convention self-timed circuits,” *Elsevier’s Integration, The VLSI Journal*, vol. 37, pp. 135–165, Aug. 2004.

[13] S. Smith and J. Di, "Designing asynchronous circuits using null convention logic (NCL)," *Synthesis Lectures on Digital Circuits and Systems*, July 2009.

[14] K. Fant and S. Brandt, "Null convention logicTM: a complete and consistent logic for asynchronous digital circuit synthesis," *Proceedings of International Conference on Application Specific Systems, Architectures and Processors*, pp. 261–273, Aug 1996.

[15] S. Hauck, "Asynchronous design methodologies: an overview," *Proceedings of the IEEE*, vol. 83, pp. 69 –93, jan 1995.

[16] V. B. A. Joshi, M.V. Jegadeesan, "NCL implementation of dual-rail 2 s complement 88 booth2 multiplier using static and semi-static primitives," *IEEE Region 5 Technical Conference*, pp. 59–64, April 2007.

[17] A. DeHon, "Deterministic addressing of nanoscale devices assembled at sublithographic pitches," *IEEE Transactions on Nanotechnology*, vol. 4, pp. 681 – 687, Nov. 2005.

[18] J. Mustafa and R. Waser, "A novel reference scheme for reading passive resistive crossbar memories," *IEEE Transactions on Nanotechnology*, vol. 5, pp. 687 – 691, Nov. 2006.

[19] M. Ziegler and M. Stan, "Design and analysis of crossbar circuits for molecular nanoelectronics," *Proceedings of the IEEE Conference on Nanotechnology*, pp. 323–327, 2002.

[20] N. A. Melosh, A. Boukai, F. Diana, B. Gerardot, A. Badolato, P. M. Petroff, and J. R. Heath, "Ultrahigh-Density Nanowire Lattices and Circuits," *Science*, vol. 300, no. 5616, pp. 112–115, 2003.

[21] M. Tehranipoor, "Defect tolerance for molecular electronics-based nanofabrics using built-in self-test procedure," in *Proceedings of the 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 305–313, 2005.

[22] J. R. Heath, P. J. Kuekes, G. S. Snider, and R. S. Williams, "A defect-tolerant computer architecture: Opportunities for nanotechnology," *Science*, vol. 280, pp. 1716–1721, 1998.

[23] J. Hu, T. W. Odom, and C. M. Lieber, "Chemistry and physics in one dimension: Synthesis and properties of nanowires and nanotubes," *Accounts of Chemical Research*, vol. 32, no. 5, pp. 435–445, 1999.

[24] Y. Chen and D. Ohlberg, "Nanoscale molecular-switch devices fabricated by imprint lithography," vol. 82, pp. 1610–1612, Mar 2003.

[25] L. Paulson, "Researchers work on transistor successor," *Computer*, vol. 38, pp. 17–23, May 2005.

[26] A. Dehon, "Nanowire-based programmable architectures," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 1, no. 2, pp. 109–162, 2005.

[27] K. P. H. T. Snider, G. and R. Stanley Willams, "Nanoelectronic architectures," *Applied Physics A 80*, pp. 1183–1195, March 2005.

[28] G. Snider and P. Kuekes, "Nano state machines using hysteretic resistors and diode crossbars," *IEEE Transactions on Nanotechnology*, vol. 5, pp. 129 – 137, March 2006.

[29] D. B. Strukov, G. S. Snider, D. R. Stewart, and S. R. Williams, "The missing memristor found," *Nature*, vol. 453, pp. 80–83, May 2008.

[30] R. S. Chakraborty, S. Paul, and S. Bhunia, "Analysis and robust design of diode-resistor based nanoscale crossbar pla circuits," in *Proceedings of the 21st International Conference on VLSI Design*, pp. 441–446, 2008.

[31] S. C. Goldstein and M. Budiu, "Nanofabrics: Spatial computing using molecular electronics," in *Proceedings of the 28th annual international symposium on Computer architecture*, pp. 178–191, 2001.

[32] S. C. Goldstein and Rosewater, "Digital logic using molecular electronics," in *International Solid-State Circuits Conference*, pp. 125–128, 2002.

[33] M. Mishra and S. C. Goldstein, "Scalable defect tolerance for molecular electronics," in *International Symposium on High-Performance Computer Architecture (HPCA)*, p. 78, 2002.

[34] D. Strukov and K. Likharev, "Reconfigurable hybrid cmos/nanodevice circuits for image processing," *IEEE Transactions on Nanotechnology*, vol. 6, pp. 696 –710, Nov. 2007.

[35] K. Likharev, "Defect-tolerant hybrid CMOS/Nanoelectronic Circuits," in *IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems*, p. 504, Oct. 2008.

[36] S. Mitra, L. Avya, and E. McCluskey, "Efficient multiplexer synthesis techniques," *IEEE Design and Test of Computers*, vol. 17, pp. 90–97, Oct-Dec 2000.

[37] S. C. Smith, R. F. Demara, J. S. Yuan, M. Hagedorn, and D. Ferguson, "Delay-insensitive gate-level pipelining," *Integration, the VLSI journal*, vol. 30, pp. 103–131, 2001.

[38] J. Huang, M. Tahoori, and F. Lombardi, "On the defect tolerance of nano-scale two-dimensional crossbars," *IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 96–104, Oct. 2004.

[39] S. Venkateswaran, J.-S. Lee, and M. Choi, "Novel functional testing technique for asynchronous nanowire crossbar system," in *IEEE Instrumentation and Measurement Technology Conference*, pp. 1121 –1125, May 2009.

[40] S. Venkateswaran and M. Choi, "Post-configuration testing of asynchronous nanowire crossbar architecture," in *IEEE Conference on Nanotechnology*, pp. 899 –902, Aug. 2008.

[41] L. Chua, "Memristor-the missing circuit element," *IEEE Transactions on Circuit Theory*, vol. 18, pp. 507–519, January 2003.

[42] J. R.Colin, "Memristor created: Rewrite the textbook," *EETIMES*, 2008.

[43] P. O. Vontobel, W. Robinett, P. J. Kuekes, D. R. Stewart, J. Straznicky, and R. S. Williams, "Writing to and reading from a nano-scale crossbar memory based on memristors," *Nanotechnology*, vol. 20, no. 42, pp. 425–504, 2009.

[44] Y. Ho, G. Huang, and P. Li, "Nonvolatile memristor memory: Device characteristics and design implications," in *IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers*, pp. 485 –490, 2009.

[45] K.-H. Jo, C.-M. Jung, K.-S. Min, and S.-M. Kang, "Self-adaptive write circuit for low-power and variation-tolerant memristors," *IEEE Transactions on Nanotechnology*, vol. 9, pp. 675 –678, Nov. 2010.

[46] S. Shin, K. Kim, and S.-M. Kang, "Memristor applications for programmable analog ICs," *IEEE Transactions on Nanotechnology*, vol. 10, pp. 266 –274, March 2011.

[47] T. Raja and S. Mourad, "Digital logic implementation in memristor-based crossbars - a tutorial," in *IEEE International Symposium on Electronic Design, Test & Applications*, pp. 303–309, 2010.

[48] J. Wu and M. Choi, "Memristor lookup table (mlut)-based asynchronous nanowire crossbar architecture," in *IEEE Conference on Nanotechnology*, pp. 1100 –1103, Aug. 2010.

[49] M. Lazzaroni, V. Piuri, and C. Maziero, "Computer security aspects in industrial instrumentation and measurements," in *IEEE Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 1216 –1221, May 2010.

[50] J. Kocher, P. Jaffe and B. Jun, "Introduction to differential power analysis and related attacks," in *Technical Report, Cryptography Research Inc., San Francisco, California*, 1998.

[51] D. Macii and D. Petri, "Accurate software-related average current drain measurements in embedded systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 56, pp. 723 –730, June 2007.

[52] V. Konstantakos, K. Kosmatopoulos, S. Nikolaidis, and T. Laopoulos, "Measurement of power consumption in digital systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 55, pp. 1662 –1670, Oct. 2006.

[53] D. Macii and D. Petri, "An effective power consumption measurement procedure for bluetooth wireless modules," *IEEE Transactions on Instrumentation and Measurement*, vol. 56, pp. 1355 –1364, aug. 2007.

[54] L. Angrisani, M. D'Apuzzo, and M. Vadursi, "Power measurement in digital wireless communication systems through parametric spectral estimation," *IEEE Transactions on Instrumentation and Measurement*, vol. 55, pp. 1051 – 1058, Aug. 2006.

[55] T. Lopez and R. Elferich, "Measurement technique for the static output characterization of high-current power MOSFETs," *IEEE Transactions on Instrumentation and Measurement*, vol. 56, pp. 1347 –1354, Aug. 2007.

[56] S. Ors, F. Gurkaynak, E. Oswald, and B. Preneel, "Power-analysis attack on an ASIC AES implementation," in *International Conference on Information Technology: Coding and Computing*, vol. 2, pp. 546 – 552, Apr. 2004.

[57] Y. Han, X. Zou, Z. Liu, and Y. Chen, "Improved differential power analysis attacks on AES hardware implementations," in *International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 2230 –2233, Sep. 2007.

[58] P. Kocher, "Design and validation strategies for obtaining assurance in countermeasures to power analysis and related attacks," in *Proceedings of the NIST Physical Security Workshop*, 2005.

[59] A. Bogdanov, "Multiple-differential side-channel collision attacks on AES," in *Cryptographic Hardware and Embedded Systems (CHES), Washington, D.C., USA, August*, pp. 30–44, 2008.

[60] M. Alioto, L. Giancane, G. Scotti, and A. Trifiletti, "Leakage power analysis attacks: A novel class of attacks to nanometer cryptographic circuits," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 57, pp. 355 –367, feb. 2010.

[61] J. Wu, Y.-B. Kim, and M. Choi, "Low-power side-channel attack-resistant asynchronous S-box design for AES cryptosystems," in *Proceedings of the 20th Symposium on Great Lakes Symposium on VLSI*, pp. 459–464, 2010.

[62] K. Tiri and I. Verbauwhede, "Securing encryption algorithms against DPA at the logic level: Next generation smart card technology," in *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, pp. 125–136, 2003.

[63] D. Sokolov, J. P. Murphy, A. Bystrov, and A. Yakovlev, “Improving the security of dual-rail circuits,” in *Workshop on Cryptographic Hardware and Embedded Systems(CHES)*, pp. 282–297, 2004.

[64] K. Tiri and I. Verbauwhede, “A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation,” in *Design, Automation and Test in Europe Conference and Exhibition*, vol. 1, pp. 246–251, Feb. 2004.

[65] S. Yang, W. Wolf, N. Vijaykrishnan, D. N. Serpanos, and Y. Xie, “Power attack resistant cryptosystem design: A dynamic voltage and frequency switching approach,” in *Proceedings of the conference on Design, Automation and Test in Europe*, pp. 64–69, 2005.

[66] J. Golic and R. Menicocci, “Universal masking on logic gate level,” *Electronics Letters*, vol. 40, pp. 526 – 528, Apr. 2004.

[67] C. Gebotys, “A table masking countermeasure for low-energy secure embedded systems,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, pp. 740 –753, july 2006.

[68] J.-S. Coron, “Resistance against differential power analysis for elliptic curve cryptosystems,” *Cryptographic Hardware and Embedded Systems*, vol. 1717, pp. 725–726, 1999.

[69] D. Suzuki, M. Saeki, and T. Ichikawa, “Random switching logic: A countermeasure against DPA based on transition probability,” tech. rep., International Association for Crpyotologic Research (IACR) EPrint Archive, 2004.

[70] A. Razafindraibe, M. Robert, and P. Maurine, “Analysis and improvement of dual rail logic as a countermeasure against DPA,” in *Integrated Circuit and System Design. Power and Timing Modeling, Optimization and Simulation*, vol. 4644, pp. 340–351, 2007.

[71] S. Mangard, “Masked dual-rail pre-charge logic: DPA-resistance without routing constraints,” in *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, pp. 172–186, 2005.

[72] T. Popp, M. Kirschbaum, T. Zefferer, and S. Mangard, “Evaluation of the masked logic style MDPL on a prototype chip,” in *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, pp. 81–94, 2007.

[73] T. Popp, M. Kirschbaum, and S. Mangard, “Practical attacks on masked hardware,” in *The Cryptographers’ Track at the RSA Conference on Topics in Cryptology*, pp. 211–225, 2009.

[74] M. Ahn and H. Lee, “Experiments and hardware countermeasures on power analysis attacks,” in *Computational Science and Its Applications (ICCSA)*, vol. 3982, pp. 48–53, 2006.

[75] K. J. Kulikowski, M. Su, A. Smirnov, A. Taubin, M. G. Karpovsky, and D. MacDonald, “Delay insensitive encoding and power analysis: A balancing act,” in *Proceedings of the 11th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, pp. 116–125, 2005.

[76] F. Gurkaynak, S. Oetiker, H. Kaeslin, N. Felber, and W. Fichtner, “Improving DPA security by using globally-asynchronous locally-synchronous systems,” in *Proceedings of the 31st European Solid-State Circuits Conference*, pp. 407 – 410, Sept. 2005.

[77] S. Moore, R. Anderson, P. Cunningham, R. Mullins, and G. Taylor, “Improving smart card security using self-timed circuits,” in *Proceedings of the 8th International Symposium on Asynchronous Circuits and Systems*, pp. 211–218, 2002.

[78] S. Moore, R. Anderson, R. Mullins, G. Taylor, and J. J. A. Fournier, “Balanced self-checking asynchronous logic for smart card applications,” *Journal of Microprocessors and Microsystems*, vol. 27, pp. 421–430, 2003.

[79] D. Sokolov, J. Murphy, A. Bystrov, and A. Yakovlev, “Design and analysis of dual-rail circuits for security applications,” *IEEE Transactions on Computers*, vol. 54, pp. 449–460, April 2005.

[80] V. Satagopan, B. Bhaskaran, A. Singh, and S. C. Smith, “Automated energy calculation and estimation for delay-insensitive digital circuits,” *Microelectron Journal*, vol. 38, pp. 1095–1107, October 2007.

[81] A. Bailey, A. A. Zahrani, G. Fu, J. Di, and S. C. Smith, “Multi-threshold asynchronous circuit design for ultra-low power,” *Journal of Low Power Electronics*, vol. 4, Dec.

[82] J. Wu, Y. Shi, and M. Choi, “FPGA-based measurement and evalution of power analysis attack resistant asynchronous S-Box,” in *IEEE Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 1–6, May. 2011.

[83] J. Wolkerstorfer, E. Oswald, and M. Lamberger, “An ASIC implementation of the AES S-Boxes,” in *Proceedings of the The Cryptographer’s Track at the RSA Conference on Topics in Cryptology*, pp. 67–78, 2002.

[84] *NIST*, “Advanced encryption standard (AES),FIPS PUBS 197, national institute of standards and technology,” in *NIST*, 2001.

[85] L. Medina, R. de Jesus Romero-Troncoso, E. Cabal-Yepez, J. de Jesus Rangel-Magdaleno, and J. Millan-Almaraz, “FPGA-based multiple-channel vibration analyzer for industrial applications in induction motor failure detection,” *IEEE Transactions on Instrumentation and Measurement*, vol. 59, pp. 63 –72, Jan. 2010.

[86] J. Hunsinger and B. Serio, "FPGA implementation of a digital sequential phase-shift stroboscope for in-plane vibration measurements with subpixel accuracy," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, pp. 2005 –2011, Sept. 2008.

[87] R. Jevtic and C. Carreras, "Power measurement methodology for FPGA devices," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, pp. 237 –247, Jan. 2011.

[88] R. C. for Information Security, "Side-channel attack standard evaluation board SASEBO-GII specification," September 2009.

[89] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," pp. 388–397, Springer-Verlag, 1999.

[90] T. Katashita, A. Satoh, T. Sugawara, N. Homma, and T. Aoki, "Enhanced correlation power analysis using key screening technique," in *International Conference on Reconfigurable Computing and FPGAs*, pp. 403 –408, Dec. 2008.

[91] S. Guilley, L. Sauvage, F. Flament, V.-N. Vong, P. Hoogvorst, and R. Pacalet, "Evaluation of power constant dual-rail logics countermeasures against DPA with design time security metrics," *IEEE Transactions on Computers*, vol. 59, pp. 1250 –1263, Sept 2010.

[92] H. Li, K. Wu, B. Peng, Y. Zhang, X. Zheng, and F. Yu, "Enhanced correlation power analysis attack on smart card," in *International Conference for Young Computer Scientists*, pp. 2143 –2148, Nov. 2008.

[93] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks-Revealing the Secrets of Smart Cards.* Springer, March, 2007.

[94] N. Homma, S. Nagashima, T. Sugawara, T. Aoki, and A. Satoh, "A high-resolution phase-based waveform matching and its application to side-channel attacks," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E91-A, pp. 193–202, January 2008.

[95] H. Alstad and S. Aunet, "Improving circuit security against power analysis attacks with subthreshold operation," in *IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems (DDECS)*, pp. 1–2, April 2008.

[96] W. Qian and M. Riedel, "The synthesis of robust polynomial arithmetic with stochastic logic," in *45th ACM/IEEE Design Automation Conference*, pp. 648 –653, June 2008.

[97] W. Qian, X. Li, M. Riedel, K. Bazargan, and D. Lilja, "An architecture for fault-tolerant computation with stochastic logic," *IEEE Transactions on Computers*, vol. 60, pp. 93 –105, Jan. 2011.

[98] S. Toral, J. Quero, and L. Franquelo, "Stochastic pulse coded arithmetic," in *The 2000 IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 599 –602, 2000.

[99] B. Brown and H. Card, "Stochastic neural computation. i. computational elements," *IEEE Transactions on Computers*, vol. 50, pp. 891 –905, Sep 2001.

[100] W. Qian, M. Riedel, H. Zhou, and J. Bruck, "Transforming probabilities with combinational logic," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, pp. 1279 –1292, Sep. 2011.

[101] B. J. Weikang Qian and M. Riedel, "International journal of nanotechnology and molecular computation (ijnmc)," *The Synthesis of Stochastic Circuits for Nanoscale Computation*, pp. 39 –57, April. 2009.

[102] W. Wijesinghe, M. Jayananda, and D. Sonnadara, "Hardware implementation of random number generators," in *Proceedings of the Technical Session of Institute of Physics*, vol. 1, pp. 25–36, 2006.

[103] J. Wu, Y. Shi, and M. Choi, "Measurement and evaluation of power analysis attacks on asynchronous S-Box," in *IEEE Transactions on Instrumentation and Measurement*, Dec. 2012.

[104] M. Dong and L. Zhong, "Logic synthesis with nanowire crossbar: reality check and standard cell-based integration," in *Proceedings of the conference on Design Automation and Test*, DATE '08, pp. 268–271, ACM, 2008.

[105] W. Lu and C. M. Lieber, "Nanoelectronics from the bottom up," *Nature Materials*, vol. 6, pp. 841–850, Nov. 2007.

[106] P. Lu, Wei. Xie and C. M. Lieber, "Nanowire transistor performance limits and applications," *IEEE transactions on electron devices*, vol. 55, pp. 2859–2876, Nov. 2008.

[107] L. Feng, Wayne. Cheng and W. Lu, "Esaki tunnel diodes based on vertical si-ge nanowire heterojunctions," *Applied Physics Letters*, vol. 99, April 2012.

[108] A. Mehonic, S. Cueff, M. Wojdak, S. Hudziak, O. Jambois, C. Labbe, B. Garrido, R. Rizk, and A. Kenyon, "Resistive switching in silicon suboxide films," *Journal of Applied Physics*, vol. 111, April 2012.

[109] C. Sui, J. Wu, Y. Shi, Y.-B. Kim, and M. Choi, "Random dynamic voltage scaling design to enhance security of NCL S-Box," in *Internatinal Midwest Symposium on Circuits and Systems*, 2011.

VITA

Jun Wu was born in Lushan, Jiangxi, China. She received her B.S. degrees in Electrical Engineering from Beijing University of Technology, Beijing, China in 2006. She joined Coolsand Technologies in May 2006 as an ASIC Verification Engineer responsible for verifying multimedia chips for mobile phones. After two years working in Coolsand Technologies, she started her Ph.D. program in Computer Engineering at Missouri University of Science and Technology in August 2008. She worked as a research assistant and a teaching assistant. In December 2012, she received her Ph.D. degree in Computer Engineering from the Department of Electrical and Computer Engineering, Missouri University of Science and Technology (Missouri S&T), Rolla, MO, USA. Upon graduation, she will be a hardware engineer at Algotochip Corporation, Sunnyvale, CA.