Fall 2016

ASSESSING AND IMPROVING THE RELIABILITY AND SECURITY OF CIRCUITS AFFECTED BY NATURAL AND INTENTIONAL FAULTS

Hoda Pahlevanzadeh
University of New Hampshire, Durham

ASSESSING AND IMPROVING THE RELIABILITY AND SECURITY OF CIRCUITS AFFECTED BY NATURAL AND INTENTIONAL FAULTS

BY

HODA PAHLEVANZADEH

Bachelor of Science, Tehran Azad University, 2008

Master of Science, University of Colorado, Boulder, 2012

DISSERTATION

Submitted to the University of New Hampshire

in Partial Fulfillment of

the Requirements for the Degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

September, 2016

This thesis/dissertation has been examined and approved in partial fulfillment of the requirements for the degree of PhD in Electrical Engineering by:

Dr. Thesis/Dissertation Director, Qiaoyan Yu, Assistant Professor (Electrical and Computer Engineering)

Dr. W. Thomas Miller III, Professor (Electrical and Computer Engineering)

Dr. Nicholas J. Kirsch, Associate Professor (Electrical and Computer Engineering)

Dr. Edward Song, Assistant Professor (Electrical and Computer Engineering)

Dr. May-Win Thein, Associate Professor (Mechanical Engineering)

On June 13 2016

Original approval signatures are on file with the University of New Hampshire Graduate School.

Dedication

I would like to dedicate this dissertation to:

my parents Hassan and Nasrin,
my sister Azadeh,
my husband Majid,
and my daughter Yasmin.

Acknowledgment

I would like to thank my parents Dr. Hassan Pahlevanzadeh and Nasrin Attaran-Rezai for believing in, supporting, and providing me the best educational opportunities over the years.

I thank my beloved husband Dr. Majid Ghayoomi for all his emotional support that helped me to conquer my fears, all his encouragement that kept me moving, and all his great advices that made this journey smooth.

I thank my advisor Prof. Qiaoyan Yu for the research opportunity and advice, and express my sincere gratitude to the rest of my thesis committee: Prof. W. Thomas Miller III, Prof. Nicholas J. Kirsch, Prof. Edward Song, and Prof. May-Win Thein for their insightful comments and advices.

A special thanks goes to Prof. May-Win Thein for being such a supportive and caring person that helped me through the most difficult times. I am very grateful that her spiritual support motivated me to continue and finish my studies.

I also thank the Department of Electrical and Computer Engineering for the teaching assistantship opportunity throughout my studies at the University of New Hampshire that helped me improve my teaching skills.

Last but not least, I thank my fellow groupmates Jaya Dofe and Mohammd Raashid Ansari for the academic and emotional support, and of course for all the fun we have had.

Table of Contents

List of Tables

List of Figures

ABSTRACT

ASSESSING AND IMPROVING THE RELIABILITY AND SECURITY OF CIRCUITS AFFECTED BY NATURAL AND INTENTIONAL FAULTS

by

Hoda Pahlevanzadeh

University of New Hampshire, September, 2016

The reliability and security vulnerability of modern electronic systems have emerged as concerns due to the increasing natural and intentional interferences. Radiation of high-energy charged particles generated from space environment or packaging materials on the substrate of integrated circuits results in natural faults. As the technology scales down, factors such as critical charge, voltage supply, and frequency change tremendously that increase the sensitivity of integrated circuits to natural faults even for systems operating at sea level. An attacker is able to simulate the impact of natural faults and compromise the circuit or cause denial of service. Therefore, instead of utilizing different approaches to counteract the effect of natural and intentional faults, a unified countermeasure is introduced. The unified countermeasure thwarts the impact of both reliability and security threats without paying the price of more area overhead, power consumption, and required time.

This thesis first proposes a systematic analysis method to assess the probability of natural faults propagating the circuit and eventually being latched. The second part of this work focuses on the methods to thwart the impact of intentional faults in cryptosystems. We exploit a power-based side-channel analysis method to analyze the effect of the existing fault detection methods for

natural faults on fault attack. Countermeasures for different security threats on cryptosystems are investigated separately. Furthermore, a new micro-architecture is proposed to thwart the combination of fault attacks and side-channel attacks, reducing the fault bypass rate and slowing down the key retrieval speed. The third contribution of this thesis is a unified countermeasure to thwart the impact of both natural faults and attacks. The unified countermeasure utilizes dynamically alternated multiple generator polynomials for the cyclic redundancy check (CRC) codec to resist the reverse engineering attack.

Chapter 1. Introduction

1.1.Challenges of Natural Faults' Impact

As the technology is emerging and integrated circuits are getting more complicated the circuit reliability is affected tremendously. The importance of the reliability factor of integrated circuits is increasing as our daily lives are more dependent on electronic devices. Circuit reliability is a statistical concept that determines the period of time that the circuit can perform successfully under a certain condition. Therefore, the reliability factor needs to be determined during the IC design before the circuit is finalized and ready for manufacturing. After evaluating the circuit reliability hardening techniques will be applied to protect the circuit against the addressed failures. The circuit reliability can be affected by a combination of various and simultaneous failure mechanisms like electromigration failure, Oxide failure, radiation effects, and etc. Evaluating the sensitivity of an IC to failures in an early phase of the design cycle is mandatory to determine if the circuit must be protected. Various protection techniques also need to be assessed and the most suitable and efficient one will be chosen. To avoid extra design and fabrication cycles in the protection circuit the reliability evaluation needs to be as accurate as possible. Over or under estimation of circuit reliability will result in extra or lack of sufficient protection circuitry. Therefore, addressing the challenges of natural faults is inevitable to avoid unnecessary tradeoffs that affect the efficiency of ICs.

As it was mentioned above, various types of natural fault can cause an error that potentially result in a system failure. In this work we are going to consider radiation as the fault. The radiation effect is due to high-energy particles hitting on the substrate of integrated circuits and can cause transient error(s) that is random and unpredictable. The assessment of circuit reliability gets complicated due to the transient error characteristics; randomness and unpredictability. Therefore,

in order to have an accurate estimation of the circuit reliability random simulation is required. However, considering all the possible cases of random simulation is time consuming and will result in loss of market opportunities. Thus, in this proposal we are providing a time efficient systematic approach to evaluate soft error rate (SER) which is one of the significant factors in circuit reliability and needs to be assessed accurately and fast.

1.1.1. Soft Errors from Space to Ground

High energy particles striking on the substrate of integrated circuits may cause unexpected and random faults known as soft errors. In 1975, Binder et al. [1] published the first report of soft errors in the space applications. In this paper the failure was related to the extra charges that were generated by Neutrons generated by cosmic radiation interacting with the earth's atmosphere. In addition the generated Neutrons ionized the IC substrate and created electron-hole pairs. The extra charges resulted in charging the base-emitter capacitance of critical transistors to the threshold voltage and caused the flip flops to store incorrect values. Since the number of soft errors were very small the failure mechanism was not considered a significant problem in the space level applications. Soft errors were not considered in the ground level applications at all since heavy ion rays, such as the 100-MeV iron particles, are not capable of crossing the earth's atmosphere [2].

On the ground level especially in the lower altitudes the number of high-energy particles is low but there is still a chance of random errors. 1978, May and Woods of Intel [3] discussed about soft errors at the ground level applications. They determined that soft errors were caused by Alpha particles emitted in the radioactive decay of Uranium and Thorium impurities just in few parts per million levels in packaging materials. Therefore, assessing the reliability of circuits at both ground and space applications is required to protect and increase the efficiency of integrated circuits.

1.1.2. Impact of Altitude on SER

O'Gorman [4] provided SER as a function of altitude and cosmic ray components in the energy range from 10 to 170 MeV at sea level (Durham, NH), Boulder CO, and Leadville CO and is plotted in Fig. 1.1. The data show that soft error rate has a linear relation with Neutron flux. At higher altitudes the amount of Neutron flux increases due to the fact that the flux of energetic neutrons generated in the atmosphere by cosmic rays increases with altitude. Therefore, in higher altitudes the chances of electronic devices being struck by high energy particles are higher. In 2008, Quantas Airbus A330-303 dropped over 400 feet twice and seriously injured a flight attendant and 11 passengers due to soft errors in an onboard computer. Although at sea level the flux of Neutrons is not significant but still cosmic rays have an impact on the soft error rate and can affect more sensitive circuits. In addition, as the existence of soft errors especially at space

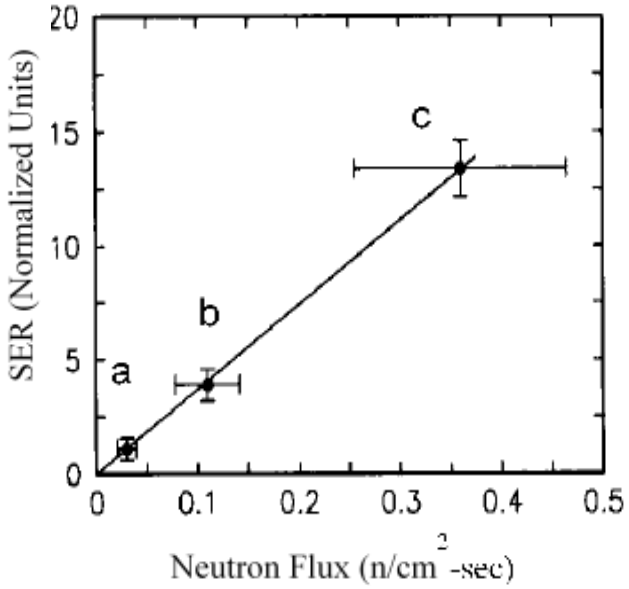


Fig. 1. 1 The cosmic ray component of the SER as a function of altitude (a) Durham NH (Sea level), (b) Boulder CO (1.6 km), (c) Leadville CO (3.1 km).

level applications is more due to the higher energy level of the charged particles, circuit protections are required that needs certain knowledge such as the amount of reliability estimation and circuit susceptibility.

1.1.3. Impact of Technology Scaling on SER

Sensitivity to soft errors caused by particle strikes are expected to increase as technology scales down [5]. Integrated circuits in smaller technology size require lower operating voltage for functionality. Thus, the critical charge shrinks and the noise margin gets narrower. As the critical charge reduces the circuit is more vulnerable to the impact of particle strike because the parasitic capacitances require less charge to be fully charged and change the status of the circuit. Narrower noise margin also means higher sensitivity to any extra charges in the circuit and results in false outcome. Another factor that is affected by technology size is the functionality speed. The speed of the circuit increases as the critical charge reduces so that charging/discharging happens significantly faster. The higher the functionality speed the more chances for faults being captured by the registers. On the other hand a soft error that has occurred in the combinational logic can be easily captured by the memory elements as the frequency of the circuit increases. Cohen et al. published a prediction of the SER trend in 1999 [7] and is illustrated in Fig. 1.2. This trend was found experimentally based on inducing alpha particles in SRAM and dynamic logic arrays. The SER trend shows as the technology size changes over time the SER increases significantly

As the technology size shrinks down the ICs get denser and more complex that increases the possibility of a particle striking the sensitive areas of the circuit and results in more soft errors.

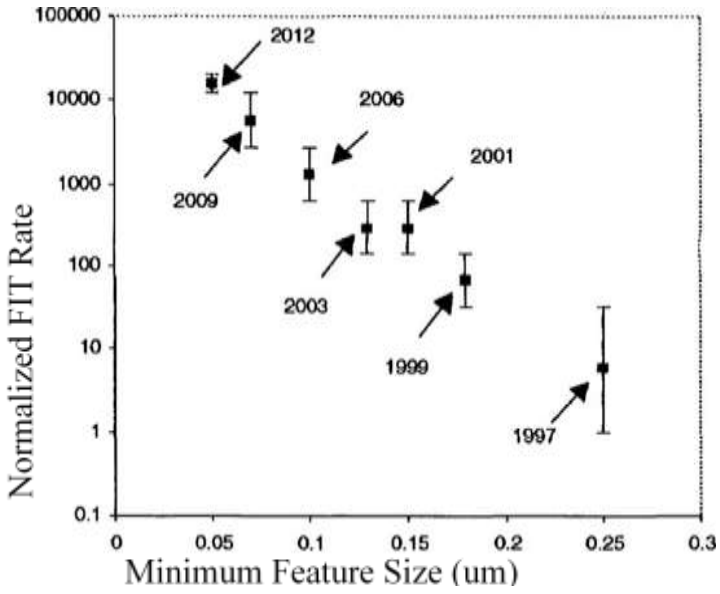


Fig. 1. 2 Scaling trend for the FIT/Mbit of SRAM and dynamic logic arrays, predicted in 1999.

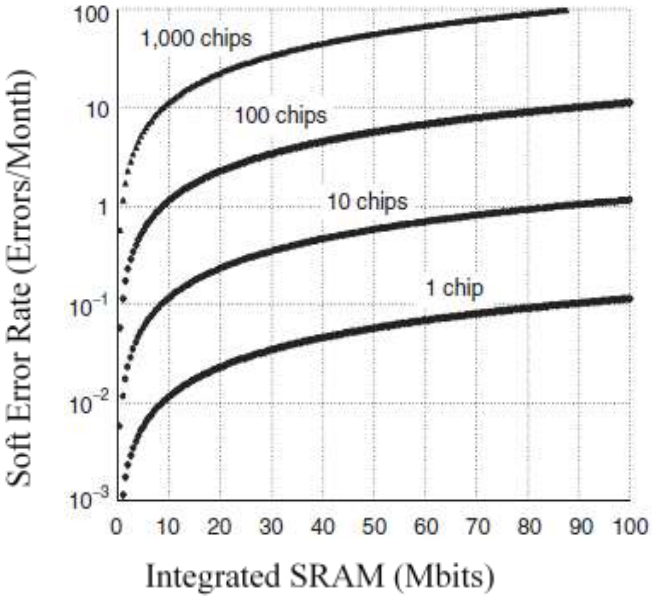


Fig. 1. 3 Monthly system soft-error rate as a function of the number of chips in the system and the amount of embedded SRAM per chip.

This fact can be observed from Fig. 1.3 [6] and indicates the importance of assessing the circuit reliability at the design stage in order to improve the functionality.

1.1.4. Increasing the Importance of Soft Errors in Combinational Logic

According to Shivakumar et al. [8] the sensitivity of combinational circuits to SER is significantly increasing as the technology size shrinks down comparing to sequential circuits. This paper compared the SER of multiple combinational and sequential circuits and plots the results in Fig. 1.4. The most significant reason that can explain this fact is related to the masking effect that can happen in combinational circuits. Soft errors may or may not cause denial of service due to the impact of error masking.

There are three types of masking effect; logical masking, electrical masking, and latch window

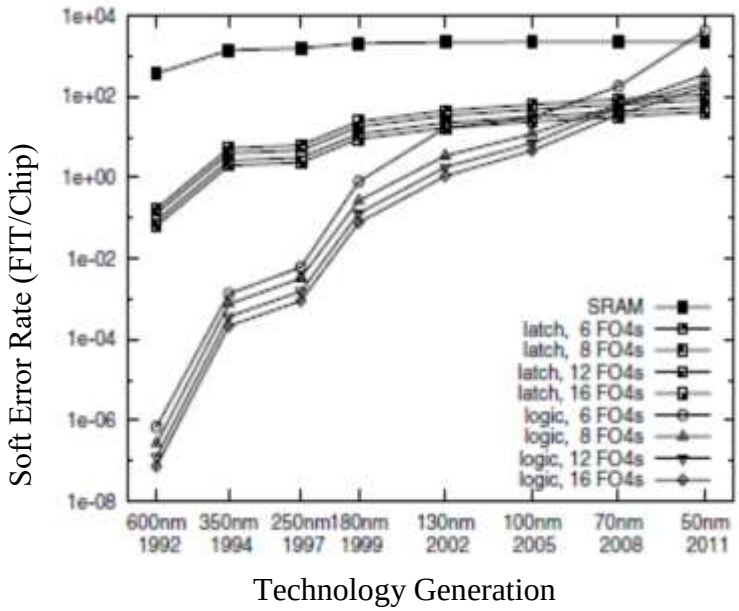


Fig. 1. 4 SER of individual circuits

masking. Due to the existence of various types of gates in the combinational logic an error can be masked when reaches one of the inputs of a gate with a dominant value on the other input. Let's consider an AND gate as an example, suppose one of the inputs of the AND gate is 0 so no matter what is on the other input the outcome is always 0. In addition, when an error propagates through the combinational logic and reaches the mentioned AND gate it will be masked due to the logical masking effect. XOR does not result in logical masking due to its characteristic. Therefore, circuits

that contain only XOR gates will be more vulnerable to errors since there is no logical masking effect to reduce the number of faults reaching the output. Another masking effect that increases the sensitivity of combinational logic to SER is electrical masking. Electrical masking has a very close relation with noise margin. As the technology scales down the noise margin becomes narrower and results in more sensitivity to extra charges and increases the SER. Latch window masking effect decreases as the frequency of the circuit gets higher. This is because higher frequency results in faster sampling by the memory elements so there is a higher chance that the propagated error gets caught by the registers. In addition, the decrease in the electrical and latch window masking effect, will let the error propagate easier through the circuit. As a result, more propagated errors will be latched in the memory elements and cause an increase in SER. As the SER value is getting more sensitive to the number of soft errors occurring in the combinational logic known as single event transient (SET) in this proposal we have estimated the SER affected only by SETs.

1.2. Challenges of Intentional Faults' Impact

1.2.1. Cryptosystems Security Threat

The widespread use of cryptographic algorithms in critical applications such as military, government, and banking systems have made the cryptosystems more significant these days. Recovering the secret key through brute-force guessing is time consuming due to the high computation complexity of ciphers. Advanced security attacks such as side-channel analysis attack (SCA) and fault attack (FA) are able to make the cryptosystem fail and retrieve the secret information.

1.2.1.1. Fault Analysis Attack

Fault attack either analyzes the security level of the cryptographic algorithm or reveal the secret information by injecting a fault and analyzing the system response. In a fault attack procedure the fault is usually injected to produce a faulty intermediate state and reveal the secret information. For instance, Fig. 1.5 illustrates a portion of an encryption algorithm that has been attacked by FA. Let's assume the injected fault changes one of the intermediate ciphertext bits to be stuck at logical 1. The intermediate ciphertext and the key are unknown to the attacker but the output is available. Therefore, by simply using the XOR operation the key will be revealed. $Y' = C'(\text{stuck at } 1) \oplus K$, the affected ciphertext and output are shown as $Y'$and C', respectively.

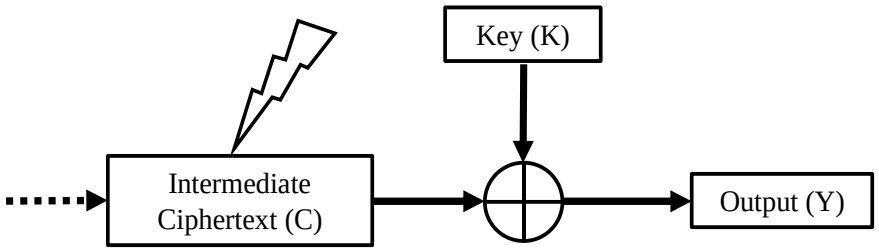


Fig. 1. 5 Fault attack on the intermediate ciphertext of a cryptographic algorithm.

Since the ciphertext is stuck at logical 1, the output will reflect the value of the key.

1.2.1.2. Side-Channel Analysis Attack

In a classic cryptosystem using a block cipher, the side-channel analysis attack is typically performed on the primary inputs and outputs as potential sources of secret information. Therefore, the plaintext, the key, and the ciphertext have to be protected against side channel attacks. Figure 1.6 shows the general structure of a cryptosystem and the side channels.

The secondary outputs mentioned in Fig.1.6 such as power consumption, time variation, electromagnetic radiation, and thermal radiation may leak information and result in a successful

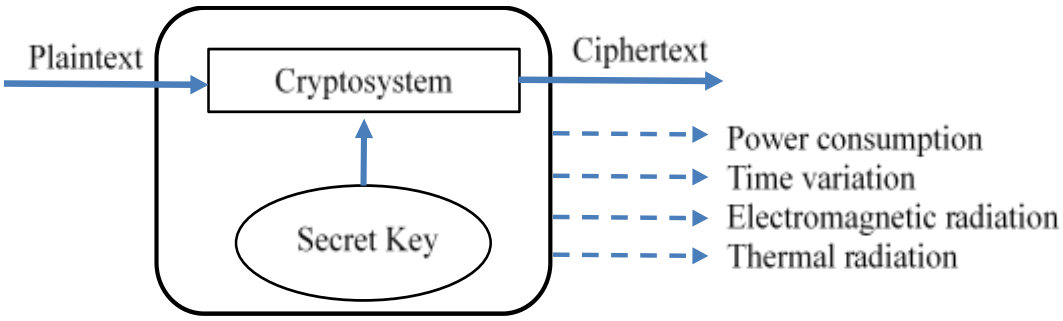


Fig. 1. 6 Block cipher implementation with secondary outputs.

attack [15]. The attacker can steal the secret key by side channel attack when the side channel information is correlated to the secret key. Side channel attacks use physical implementations to leak information from the cryptosystem. For instance, the state of a bit in the key might be dependent on the execution time of the cryptographic algorithm. Thus, the difference in time can be measured and a timing attack is performed.

1.2.1.2.1. Power Analysis Attacks

Power analysis attack is one of the most powerful and common types of side-channel attack. This type of attack uses the instantaneous power consumption by a component in the target cryptosystem and correlates it with one of the intermediate states of the cryptographic implementation.

As the different modules in the cryptosystem consume different amount of power, it is possible to learn what operation is occurring if one study the power traces carefully. For example, Fig. 1.7 shows the instantaneous power consumption of a cryptographic device as it performs the Advanced Encryption Standard (AES) algorithm [23]. The AES algorithm contains 10 rounds that is clearly noticeable from the repeating patterns in Fig. 1.7. Although the details of individual data bits being manipulated in the cipher cannot be visually determined, the power traces can be potentially used to arrange a more powerful attack.

The differences in the instantaneous power consumption can also be related to the bit values that are being manipulated. Standaert [24] showed that the hardware consumes power when a

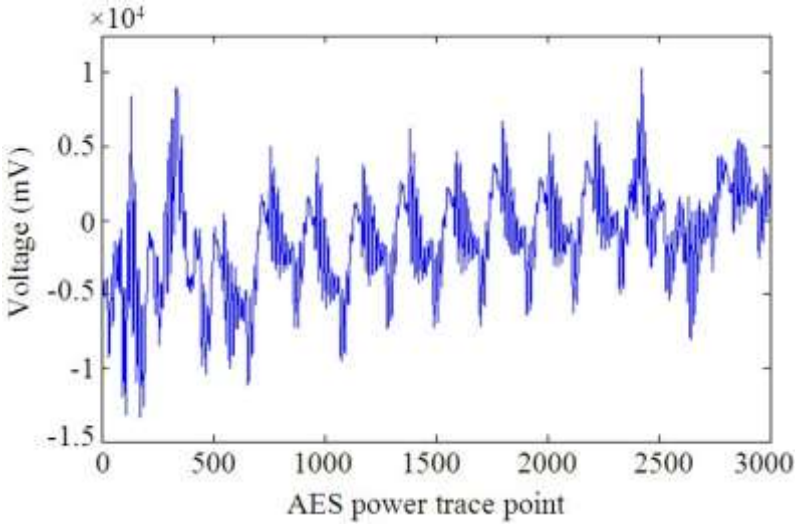


Fig. 1. 7 Power trace related to the 10 rounds of AES-128 encryption operation [26].

transition takes place in any bit, but on a much lower scale than the algorithm level that makes the detection more difficult. In order to retrieve the secret information by analyzing the power traces, statistical techniques are required to correlate the power consumption to the target key. Fig. 1.8 illustrates the voltage variations due to bit transitions. Since the current is considered to be fixed the power consumption is directly related to the voltage. As more data bit values are transitioning from logic 0 to logic 1 or vice versa, more power is consumed [25]. As it was mentioned side-channel analysis attack studies the side channel signals of the cryptographic hardware, such as power, delay, and temperature to guess the secret key. Simple power analysis (SPA) [15], differential power analysis (DPA) [15], and correlation power analysis (CPA) [14] are types of SCA. CPA requires less number of traces for recovering the key than SPA and DPA, therefore it is the most commonly used power analysis method lately. Power analysis attacks such as DPA and CPA are non-invasive and can be mounted without knowing the design of the target device [24].

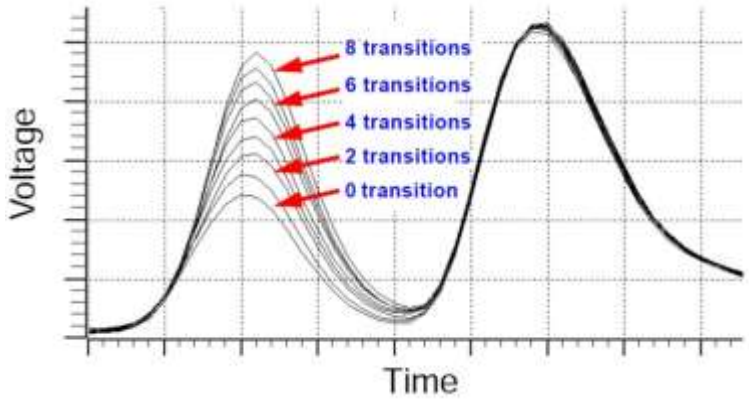


Fig. 1. 8 Variations in voltage due to bit transitions [25].

In addition, these attacks cannot generally be detected by a device, since the adversary's monitoring is normally passive.

1.2.1.3. Combined Attack

Another type of security attack is the combined attack where the adversary takes advantage of both SCA and FA to create a more powerful attack in cryptosystems [10]. After injecting a fault in the cryptographic algorithm by means of white light, laser beam, voltage/clock glitch, and temperature control [13], statistical analysis can be performed on the correct and faulty outputs to retrieve the key. This statistical analysis is known as differential fault analysis (DFA) and represents a type of combined attack. Figure 1.9 illustrates a fault injection equipment that uses laser to generate fault in the circuit. Another purpose of fault analysis is to assess the vulnerability of the circuit to attacks and leads the designer to come up with the most optimized protection method for improving the circuit reliability and security.

The main principle of side channel analysis attack is to exploit a physical leakage from the target cryptosystem in order to recover the secret information. Fault attack is another kind of

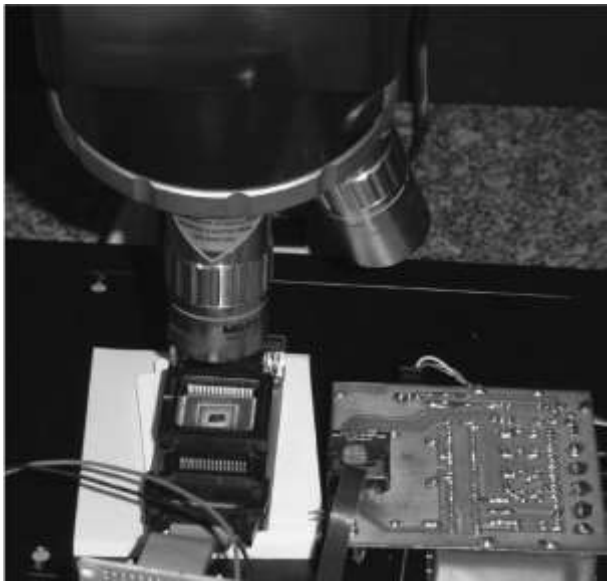


Fig. 1. 9 Laser fault injection equipment [13].

physical attack that first injects a fault then evaluates the faulty behavior of a cryptosystem in order to extract secret information. Many countermeasures have been used in different cryptosystems to thwart physical attacks. The principle of combined attacks is to attack cryptosystems which are protected against both side-channel attacks and fault attacks. The most common countermeasure for protecting block cipher implementations (e.g. AES) against side channel analysis attack is the masking technique. Random values are used in the masking approach to hide intermediate ciphertexts. One of the countermeasures that is used for protecting block ciphers against FA is DMR that utilizes the duplication of the cipher for determining any possible fault attack. Another countermeasure against FA is using the inverse operation of the cryptographic algorithm. One of the contributions of this work is to investigate the impact of FA countermeasures on the success rate of one of the side-channel attack types.

1.2.2. Similarity of Intentional Faults to Natural Faults

As it was mentioned in section 1.1, transient errors caused by high energy particle striking the substrate of ICs are a threat to circuit reliability. Soft errors are categorized as natural faults that

are injected in the circuit unintentionally. The energy of charged particles generated by cosmic rays at the ground level is very low due to the earth's atmosphere. However, soft errors exist in lower altitudes mainly due to radioactive particles produced by impurities in packaging materials (e.g. alpha particles) that cause faults in chips and result in corruption in the functionality of the circuit.

Intentional radiation of white light, laser, X-rays, and ion beams [13] on specifically cryptosystems can be used as an attack by adversary in order to break the cryptographic algorithm and reveal the secret key. Intentional radiation can lead into denial of service but in most of the cases it follows a more significant outcome that is beyond corrupting the system. Intentional faults are inspired by natural faults but with a slightly different purpose. These types of fault are controlled by the attacker respect to time and location to execute secret key and compromise the cryptosystem. Thus, crypto circuits need to be protected against both natural and intentional faults.

1.3. Unified Reliability and Security countermeasure

Providing data reliability and security in on-chip communication systems is essential especially in protecting IP cores from unlicensed usage. Therefore, security and reliability are accomplished through cryptographic algorithms and error control coding (ECC), respectively [102]. According to Fig.1.10 although protection techniques have been added to the transmission channel but the system still suffers from reverse engineering in the ECC.

Reverse engineering attack targets the intellectual property or illegal access with the purpose of duplication or extracting the beneficial information and makes it a serious threat [103]. Therefore, it is required to extend the protection for the security of systems against reverse engineering. This can be done by increasing the cost of reverse engineering in terms of time [104]. Obfuscation [105] is a common protection approach that is used to counteract reverse engineering.

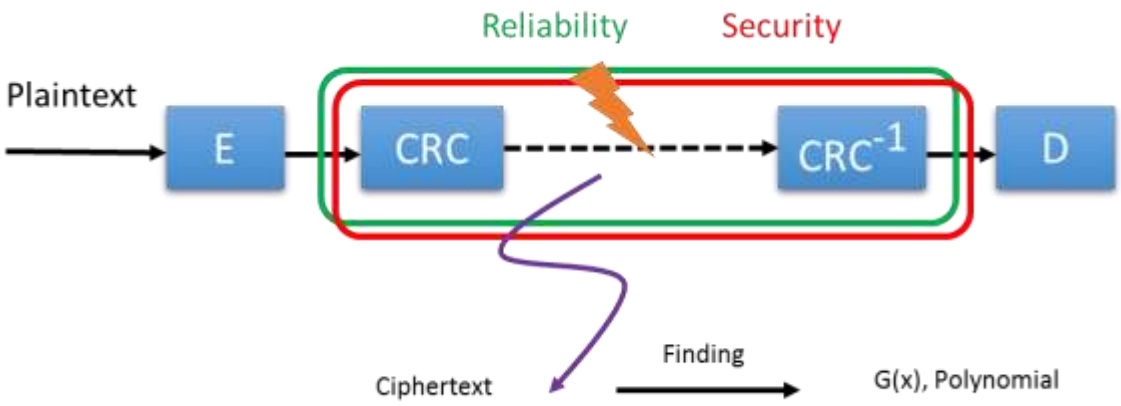


Fig. 1. 10 Security threat in a unified countermeasure.

1.4.Organization of the Thesis

There are so many works that have been done in this area but there are still gaps that need to be addressed. This thesis fills some of the gaps in space and ground level applications facing reliability and security issues. The preliminary concept and basic knowledge of this thesis will be discussed in Chapter 2.

Then, the reliability of circuits affected by soft errors is investigated in Chapter 3. This can be partly achieved by assessing the error latching probability which is one of the crucial factors in evaluating the circuit reliability of the circuit under test. Chapter 3 proposes a systematic analysis for latching probability of single-event transients with pulse widths equal or less than one period of the clock.

As single-event transients are expected to occur more often and remain longer in the smaller technology nodes, it is imperative to precisely estimate soft error rate in ICs. This means SETs can be longer and remain up to multiple clock cycles. The latching probability of SETs with multiple clock cycle pulse widths is proposed in Chapter 4.

Chapter 5 will look into the success rate of correlation power analysis attack on Advanced Encryption Standard that is protected by fault detection mechanisms. This chapter will talk about

the necessity of analyzing the impact of adding more FT methods on different modules on the AES implementation.

In order to overcome the negative impact of FD mechanism on the key retrieval speed in AES algorithm we will propose an optimal design in Chapter 6 to thwart the combined attack, reduce the fault bypass rate and decrease the key retrieval speed. The proposed countermeasure is based on dynamic masking and error deflection to thwart the combined CPA and FA attacks.

A unified countermeasure is introduced in Chapter 7 that utilizes a dynamic generator polynomial that each polynomial is selected through a selection function. The input of the selection function is the message bits that makes the selection unpredictable and increase the complexity of reverse engineering for finding the correct generator polynomial.

Chapter 2. Background

2.1. Natural Faults

2.1.1. Reliability

Before sending out the final design for fabrication, designers run a large number of simulations to predict the circuit's performance accurately. It is inconvenient and unreasonable to trust entirely on the finished IC test results and discover the errors of the design. One of the common units that is used for measuring reliability is failure in time (FIT). FIT is the number of failure per billion device hours. For instance, let's consider a system with 100,000 devices that has one failure per month. The failure rate for such system is:

$(1\ failure)/(100000 \times 30 \times 24hrs) = 14 \times 10^{-9} = 14$ FIT.

Hu [33] showed that in order to prove a 10 FIT failure rate of 50% confidence level, 2×10^7 device-hours of testing is required.

2.1.2. Soft Error Mechanism

When a high energy charged particle strikes the substrate of a semiconductor device that is revered-biased, then three mechanisms take place in the device. First, the charged particle ionizes the substrate of the device and creates pairs of electron-holes. Figure 2.1 shows a high energy particle striking the substrate of an NMOS. The electron-hole pair will separate due to the external electric field that pulls up the electrons and pushes down the holes. Then, the created charge is collected by the depletion region. This stage is known as funneling. The funnel size depends on the substrate doping. As the substrate doping decreases, the funnel distortion increases. The last stage is diffusion. In this stage the remaining charge which was generated in the device will be recombined, or diffused away from the junction for a longer time scale than the funneling stage (tens of picoseconds) until all excess carriers have been collected.

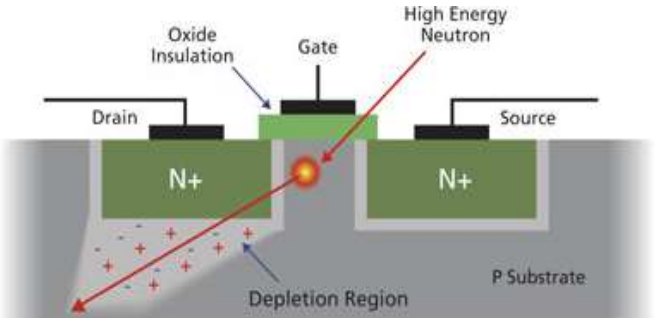


Fig. 2. 1 High energy charged particle hitting the substrate of an NMOS.

The three stages of the soft error mechanism is illustrated in Fig. 2.2 [6]. Equal number of electrons and holes in the depletion region results in a balanced net charge that means the electrons and holes have compensated each other. The shifting of the charge carriers in the funneling stage causes the net charge shifts in the depletion region that results in a change in the potential difference. The amount of potential difference that is created depends on the number of electron-hole pairs generated by the high energy particle. Therefore, the extended electric field helps in collecting the extra charge. Figure 2.2 d shows the corresponding current pulse that is resulted from the three mentioned phases. The chances of occurring a soft error increases as a particle with higher energy strikes the substrate closer to the p-n junction [34].

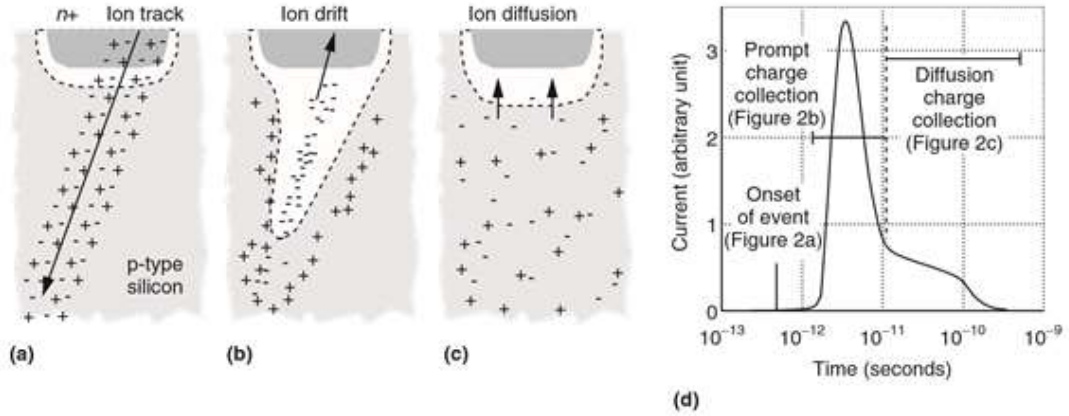


Fig. 2. 2 Three stages of the soft error mechanism: (a) ionization, (b) funneling, (c) diffusion, and (d) the corresponding current pulse [6].

2.1.3. Different Types of Soft Errors

There are different types of soft errors that are categorized based on the location of the strike; single event transient (SET) and single event upset (SEU); and the number of events that occur by one strike; multiple bit upset (MBU).

When a high energy particle strike the combinational logic and cause an upset in the primary output, if the false output propagates the combinational logic and is stored in the memory element then a SET has occurred. SEU occurs when the high energy particle strike the memory element directly and change the output to the opposite value. Figure 2.3 shows an example of a combinational logic on the left and a memory element on the right that are susceptible to a particle strike and results in SET/SEU.

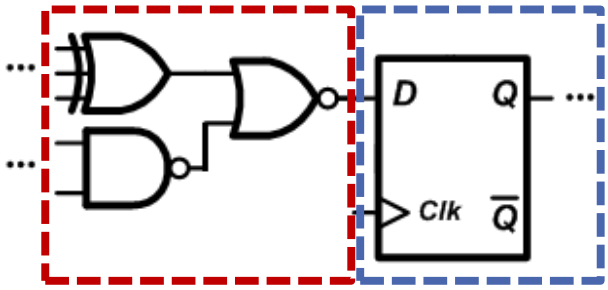


Fig. 2. 3 Soft errors striking the combinational logic (on the left) or the memory element (on the right).

2.1.4. Estimation of SER_{SET}

Evaluating SER during the design phase is the only way to avoid bad surprises when the final product is tested. The SER assessment is required to be as accurate as possible to prevent extra design and fabrication cycles for protection purposes. Protection circuits are at the expense of extra cost, area, power consumption, and time delay so in order to avoid unnecessary fault detection and fault tolerance circuits the reliability factor needs to be estimated accurately. Since soft errors are random and unpredictable, a significant large set of test vectors is required to estimate the SER. Unfortunately, a large set of test vector needs longer simulation time that may result in loss of

market opportunities. Previous work [9] tried to reduce the simulation time by identifying the critical SET pulse width in benchmark circuits and computation units.

Wang and Agrawal [9] showed that in deep circuits like inverter chains the soft error with a determined critical width can be masked by the electrical masking effect if the error occurs at the earlier gates. Therefore, SER only depends on a few gates near the primary output. In addition only the gates near the output need to be protected against faults. The weakness of this approach is it cannot be applied on all types of circuits. Unfortunately, some logic topologies, such as shallow and wide circuits like a ripple-carry adder, do not have critical SET pulse width [9].

Alternatively, FPGA-based emulation was utilized to improve the SER evaluation speed [10]. Although hardware emulation could be faster than HSPICE simulations, the selection of SET pulse width and SET injection timing is still random in typical evaluation processes. The conclusions drew from previous simulation/emulation-based work may not be applied to other new circuit topologies. Consequently, it is urgent to produce a systematic analysis method to predict the impact of latching widow masking on circuit soft error rate (SER).

To speed up the assessment of SET-induced soft errors, we propose a systematic analysis method to examine the probability of a SET eventually being latched. In previous works, the latching probability of SET is only modeled as a function of SET pulse width and clock period. As soft error rate also strongly depends on other timing parameters, our novel analytical model additionally includes logic gate delays and setup/hold time of memory elements. In this proposal, we are going to generate a set of closed-form expressions for the latching probability for various logic gate delays and SET pulse widths. The preliminary simulation results show that the soft error rate predicted by our model matches to that obtained from random simulations.

2.1.5. Estimating SER for Single and Multiple SET pulse widths

Circuits operating in Gigahertz can possibly have soft errors with durations more than a clock cycle. Due to the higher frequencies and lower voltage supplies an SET pulse width can vary from a portion of one clock cycle to multiple clock cycles. The increase in the ratio of SET pulse duration over clock period challenges the analysis of SET-induced soft errors. Therefore, we studied a systematic analysis to assess the probability of SER_{SET} for SETs with single or multiple durations with a significant faster speed.

2.2.Intentional Faults

2.2.1. Cryptosystems Security Protection

The security of cryptographic algorithms has become increasingly important in recent years due to the widespread usage of cryptosystems in critical and minor applications. For instance, Tunstall et al. [11] were able to retrieve the secret key by DFA when a single random byte fault was induced at the input of the eighth round of AES.

The impact of both natural and intentional faults on integrated circuits are relatively same therefore the traditional fault detection and fault tolerant mechanisms have been widely investigated and applied on cryptosystems to address both types of fault. Error detecting codes such as basic parity based schemes is one of the countermeasures [12] that uses a parity bit for each byte in the AES encryption conversion. In each transformation, the output parity bit is predicted from the inputs and is compared with the actual parities.

The fault detection structure introduced in [16] and [17] is applicable to algorithm-level, round-level, or operation-level. Operation-level fault detection scheme is only applied to a specific module while algorithm-level fault detection mechanism protects the whole algorithm. Karri et al. [16] introduced a fault detection scheme that is based on the original transformation or round or

whole algorithm flowed by its inverse. The output of the inversegiga function is compared with the original input. If there is any mismatch at the comparison block an alert signal will be generated. The major downside of this detection scheme is the area and time delay overhead that is due to the extra hardware used in the inverse block.

Another fault detection mechanism is time redundancy. In time redundancy the computation is repeated and the result is compared to a stored copy of the previous result. Malkin et al. [18] proposed running the transformations in an AES round twice for the same data to detect any potential transient error. The weakness of this scheme is, it is not able to detect permanent faults or even fault attacks that last for a long period.

Hardware redundancy is referred to duplication or generating more copies of a specific module, round, or the entire encryption/decryption algorithm. The comparison of the outputs from each copy indicates the conflict due to the existence of faults in the system. The classic fault detection methods provided in this section are highly effective against natural faults that follow a uniform fault model [10]. Since intentional faults are injected by an attacker and may not follow a constant fault model, the classic fault detection models may not be practical and need to be modified to thwart intentional faults.

2.2.2. Impact of Existing Countermeasures for Fault Attack on Cryptosystem Security

Countermeasures for AES to thwart side-channel attack and fault attack are typically investigated in a separate fashion. There is lack of thorough investigation of how one countermeasure specifically for one attack affects the efficiency of another attack. The additional hardware cost and power consumption induced by adding traditional fault detection circuitry to thwart fault attacks in cryptographic systems, simultaneously affects the efficiency of CPA. Maingot and Leveugle [21] were the first to show that the CPA efficiency also depends on the

existence of fault detection circuits in cryptographic systems. According to [21] the robustness of fault detection schemes to CPA are correlated with the dynamic power consumption of the chip. In addition, they indicated that in order to decrease the vulnerability of fault detection schemes to side channel attacks, the dynamic power consumption should be as little correlated as possible with the manipulated data.

Regazzoni et al. [20] show that the use of parity codes in the S-Box of AES helps the attacker to retrieve the key through CPA with less power traces comparing with the case when there is no fault detection mechanism. Unfortunately, that observation is based on the gate-level simulation on the S-Box only, rather than a real hardware emulation of the complete AES implementation. Moreover, the power model used in the existing work is Hamming weight rather than the powerful Hamming distance one.

In this thesis we are going to investigate the factors that have an influence on the key retrieval speed. We will analyze three module-level fault detection schemes, including double modular redundancy (DMR), inverse, parity check on advanced encryption standard (AES). Our preliminary experimental results show that, in some scenarios, the use of fault tolerance mechanisms in AES improves the resistance of AES against side-channel attack. In addition, it is imperative to look into all the effective factors on the key retrieval success rate therefore we will investigate the impact of elements such as type of redundancy, module under protection, and CPA attack power model. The assessment of the vulnerability of AES protected by different types of fault detection mechanisms to CPA will be examined on a FPGA platform.

In addition, we will investigate an optimum solution to overcome the negative impact of fault detection circuitries on the AES resistance to side channel based attacks.

2.3. Advanced Encryption Standard

The Advanced Encryption Standard (AES) encryption algorithm is a symmetric key block cipher that the input and output known as plaintext and ciphertext, respectively. The plaintext and ciphertext are each 128 bits. Figure 2.4 shows the 128 bit plaintext that is arranged into a 4x4 matrix called the State. The strength of the encryption algorithm increases with the length of the cipher key. The length of the cipher key can be 128, 192, or 256 bits. In this work the 128 bit cipher key is used.

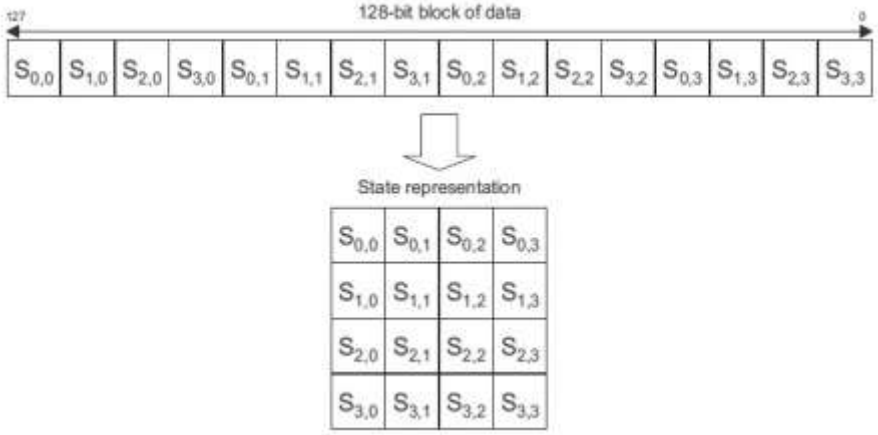


Fig. 2. 4 State representation of 128-bit data blocks [85].

The AES encryption algorithm is composed of four major transformations known as SubBytes, ShiftRows, MixColumns and AddRoundKey. AES encryption algorithm with different key sizes requires different number of rounds. For an AES with 128, 192, or 256 bits of cipher key there are a total of 10, 12, or 14 rounds respectively. Figure 2.5 illustrates the AES encryption algorithm.

The SubBytes transformation is a nonlinear transform that replaces each byte in the state with a byte from a substitution table (S-box).

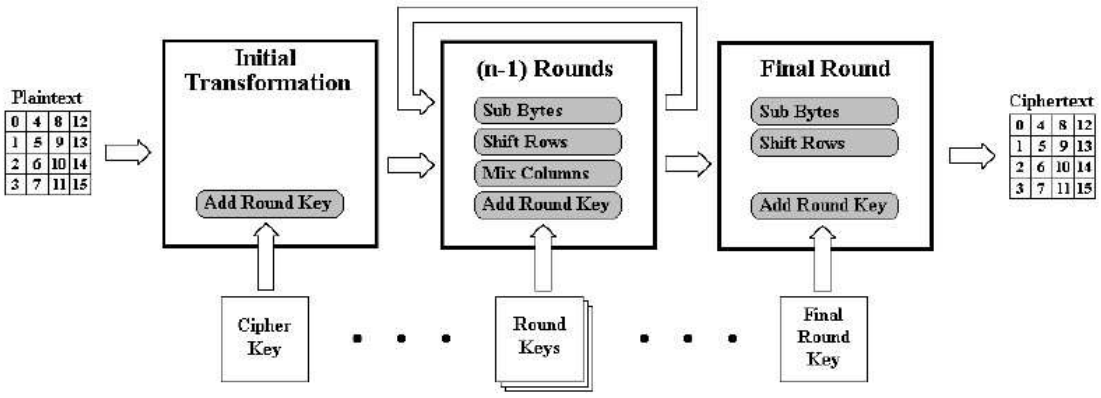


Fig. 2. 5 AES Algorithm [86].

The ShiftRows transformation cyclically shifts the rows of the state to provide horizontal diffusion. The first row is not shifted, the second row is left shifted by one byte, the third row is left shifted by two bytes and the fourth row is left shifted by three bytes.

The MixColumns transformation provides a vertical diffusion in the matrix and composes new columns with the current ones.

The AddRoundKey transform is a bitwise addition with a key generated from the Key Expansion algorithm.

2.4. Correlation Power Analysis

Correlation Power Analysis (CPA) is the advanced form of power analysis attack that utilize a power consumption model to perform an attack on a cryptographic algorithm. The power model approximates the power consumption of the target cryptographic device during an encryption operation. The predicted power consumption will then be correlated to the actual measured power consumption using a key hypothesis. The correlation plots will be generated for all the possible guesses for each subkey. The highest peak of the correlation plot gives the correct subkey hypothesis.

There are multiple methods for constructing the power model, such as simulating the cryptographic structure in a designed environment to find the power consumption [30]. The predicted power consumption will be accurate if the architecture of the target device is known. If the architecture is not known or it is not possible to simulate, a more general power model can be used for predicting the power consumption. The two common power models that are being used are the Hamming weight and Hamming distance models [31].

2.4.1. Hamming Weight and Hamming Distance Power Models

The Hamming weight model is a basic power consumption model. This power model is relied on the basic idea that the power consumption of a bus is proportional to the number of bits that are switched on. Therefore, Hamming weight is simply the number of bits set to 1, $H(D) = \sum_{j=0}^{m-1} d_j$. A bus that none of the bits are switched on consumes very little power compared to a data bus with all bits switched on [29]. Thus, the Hamming weight is proportional to the power consumption of the bus when bits switch from 0 to 1. The Hamming distance model is an extension of the Hamming weight model which uses bit transitions to determine power consumption. The change in the bit value can occur in many different circuit components such as a data or address bus, register, memory, or some other components. The power consumption in Hamming distance power model is proportion to the number of 0→1 and 1→0 transitions made within the circuit under attack. The number of bit transitions is simply the Hamming weight of the exclusive OR of the two values, HD(R0, R1) = HW(R0 $\oplus$ R1). Hamming distance encloses the Hamming weight model that assumes that R0 which is the reference is 0. It is assumed that bits which do not change (0→0, 1→1) do not contribute to the power consumption of a circuit. It is also assumed that a 0→1 and 1→0 transition consume an equal amount of power. The Hamming distance power model is generally used for Correlation Power Analysis.

2.4.2. Pearson's correlation coefficient

Pearson's correlation coefficient is a commonly used measure of correlation [32]. Equation 2.1 shows how the Pearson's correlation coefficient value (ρ) is found. The value of ρ tells us how much the linear relationship between the two variables X and Y is.

$$\rho_{X,Y} = corr(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \tag{2.1}$$

The covariance (cov) indicates how much two random variables X and Y change together. σ_X and σ_Y are the standard deviations of X and Y. The correlation coefficient value varies from +1 to -1. If the correlation coefficient is 0 it means that there is no linear relationship between the two variables. In order to improve the correlation, we need to increase the number of power traces. Due to the high number of measured traces the Pearson correlation can be estimated by the sample correlation coefficient (r_{xy}). Equation 2.2 shows how the sample correlation coefficient is found: n is the number of power traces, $\bar{x}$ and $\bar{y}$ are the sample means of X and Y, and s_x and s_y are the sample standard deviations of X and Y [29].

$$r_{x,y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y} \tag{2.2}$$

The variables X and Y in CPA will be the power consumption measurement samples and power consumption hypothesis samples. The power consumption hypothesis samples are generated from a power model that is usually the Hamming distance model.

2.4.3. Correlation Power Analysis on AES

In order to perform a CPA attack on a cryptographic algorithm, the adversary needs to analyze the design under attack and choose a data register for the attack. The chosen register has to contain sensitive data that holds a relationship with the power consumption. The more

correlation between the data and power consumption, the stronger the power model is that results in a faster key retrieval.

In the AES algorithm, there exists a relationship between the intermediate ciphertext and the power consumed in the final round of encryption. The hardware implementation of AES is shown in Fig. 2.6. The target register for the CPA attack is labeled as RB located in the data path prior to the SubBytes transformation and contains 8 bits.

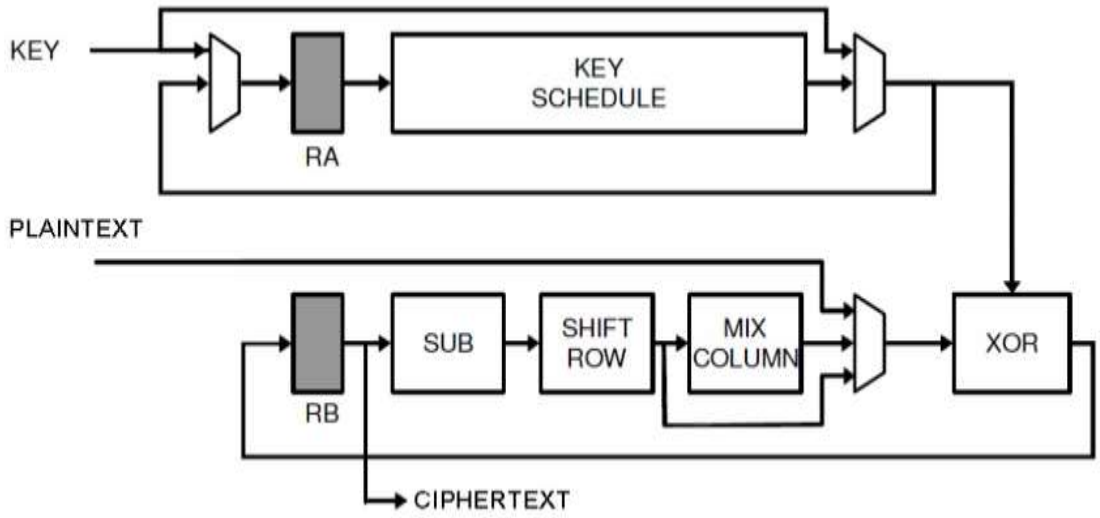


Fig. 2. 6 AES implementation [32]

After selecting the sensitive register location, a power model has to be determined to perform the CPA. In this work, the Hamming distance is chosen as the power model. Before the CPA attack is performed, the encryption operation is executed multiple times by using random plaintexts and an unknown cipher key. Then the power traces and the related intermediate ciphertexts are captured and stored for the CPA. After capturing the power traces and the intermediate ciphertexts the analytical phase of the CPA begins: the encryption process is finished by calculating the final ciphertext using the first stored intermediate ciphertext and the first 8-bit partial key (subkey) guess. Next, the resulted ciphertext in the analytical step is compared with

the stored ciphertext from the capturing step and the Hamming distance is found. In order to find the hypothetical power consumption a linear relationship between the power consumption and the Hamming distance is assumed: h = aH(D) + b, which h is the hypothetical power consumption, H(D) is the hamming distance between the mentioned ciphertexts, a is a scalar gain, and b is an offset and noise. The assumed linear relationship could be considered as a limitation, but since the bus lines that are the data dependent parts of the circuit are the most power consuming elements we can assume the power dissipation of an operation at a specific time is proportional to the hamming distance of the processing data to facilitate the analytical process [14].

After finding the hypothetical power consumption we are ready to find the sample correlation coefficient by using equation 2.3 or 2.4 The Pearson's sample correlation coefficient equation can be utilized to determine the correlation between the power and the sensitive data.

$$r_{i,j} = \frac{\sum_{d=1}^{D}\left[(h_{d,i} - \overline{h_i})(t_{d,j} - \overline{t_j})\right]}{\sqrt{\sum_{d=1}^{D}(h_{d,i} - \overline{h_i})^2 \sum_{d=1}^{D}(t_{d,j} - \overline{t_j})^2}} \tag{2.3}$$

$$r_{i,j} = \frac{D\sum_{d=1}^{D} h_{d,i}t_{d,j} - \sum_{d=1}^{D} h_{d,i}\sum_{d=1}^{D} t_{d,j}}{\sqrt{\left(\left(\sum_{d=1}^{D} h_{d,i}\right)^2 - D\sum_{d=1}^{D} {h_{d,i}}^2\right)\left(\left(\sum_{d=1}^{D} t_{d,j}\right)^2 - D\sum_{d=1}^{D} {t_{d,j}}^2\right)}} \tag{2.4}$$

The resulted $r_{i,j}$ is related to the first subkey guess and can be plotted against the total number of traces (D). Figure 2.7 shows an example of the sample correlation coefficient versus the number of traces. All the above steps have to be repeated for all the possible subkey guesses to find out the highest correlation that results in the correct subkey. The highest correlation will occure at the correct subkey. After finding all the 256 correlation coefficients for each subkey we can plot them versus the number of possible subkey guesses and there will be a spike in the graph for the correct guess of subkey byte as shown in Fig. 2.8.

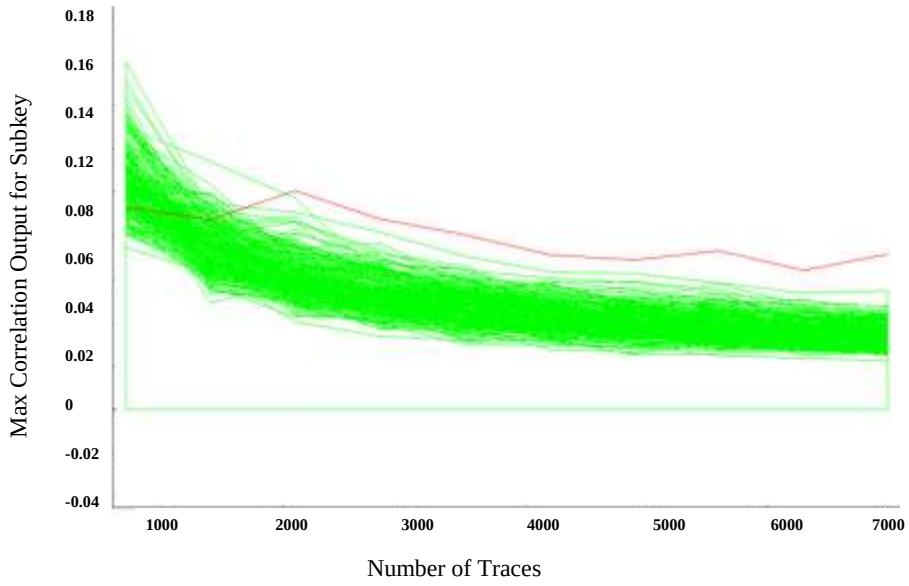


Fig. 2. 7 The correlation coefficient versus the number of traces for one retrieved subkey.

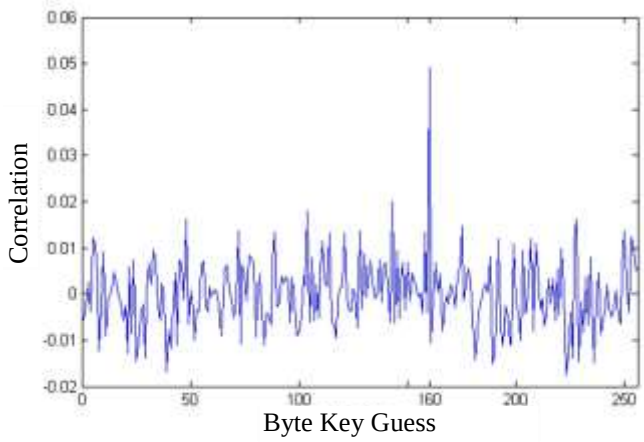


Fig. 2. 8 The correct subkey guess is found at the 160th guess in AES algorithm [29].

2.5.Partial Guessing Entropy

Massey [27] defined the guessing entropy as the average number of successive guesses required with an optimum strategy to determine the value of a random variable X. In this proposal the optimum strategy is ranking all the possible values of the subkey from most to least likely base on the correlation attack (higher correlation means the hypothetical key is more likely to be the correct one) [28]. In addition, there is an individual guessing entropy for each subkey in this work,

therefore the term “partial” has been added. At the end of the analysis each subkey will have a partial guessing entropy (PGE) that indicates how many guesses is required to achieve the correct value of the subkey. A PGE of 0 specifies the subkey is perfectly known. In order to increase the reliability of the power analysis attack, it is encouraged to run the encryption algorithm for multiple times. Therefore the PGE for each subkey is averaged over several attacks to generate a single PGE.

2.6. CRC Codec

Cyclic Redundancy Check (CRC) is a powerful error detection scheme that is based on Finite Algebra Theory [100]. Memories are prone to natural and intermittent faults that can result in the whole memory chip to fail [101]. Therefore, it is imperative to prevent the propagation of error in digital memories by detecting the faults in advance and possibly discard the faulty message. CRCs are generally used for detecting random errors in storage devices and memories. The hardware structure of CRC is composed of shift registers and XOR logic gates.

2.6.1. CRC Mechanism

Depending on the configuration of the message and the check bits there are two types of CRC; systematic and non-systematic. In the systematic CRC the check bits are located in front of the message and can be extracted by knowing the degree of the generator polynomial. In the non-systematic CRC the message bits are mixed with the generator polynomial that makes it more complicated in terms of reverse engineering.

In this section the encoding and decoding operations are represented in the form of polynomials as follows:

$$C(x) = C_n x^n + C_{n-1} x^{n-1} + \cdots + C_1 x^1 + C_0 x^0 \quad (2.1)$$

$$M(x) = m_{n-k} x^{n-k} + m_{n-k-1} x^{n-k-1} + \cdots + m_1 x^1 + m_0 x^0 \quad (2.2)$$

$$G(x) = g_k x^k + g_{k-1} x^{k-1} + \cdots + g_1 x^1 + g_0 x^0 \quad (2.2)$$

C(x), M(x), and G(x) are the polynomial representations for the codeword, message, and the generator polynomial.

2.6.1.1. Systematic CRC

The check bits or known as remainder bits R(x) are found through dividing $x^k M(x)$ by the generator polynomial G(x). By adding the remainder to $x^k M(x)$ the codeword is created. Equation 2.3 indicates the adding process of remainder bits to the message bits that forms the codeword.

$$C(x) = x^k M(x) + R(x) \quad (2.3)$$

In order to extract the message, first the syndrome has to be checked to confirm if the process is error free. The syndrome is simply the remainder of the division of C(x) by G(x). If the syndrome is zero it means there is no error and the message can be extracted securely. Extracting the message is simply done by removing the k-bits from the codeword.

2.6.1.2. Non-systematic CRC

The non-systematic CRC follows a different approach for generating the codeword and is presented in equation 2.4.

$$C(x) = \sum_{j=0}^{N-k} \left(\left(m_j x^j \right) G(x) \right) = \text{M(x).G(x)} \quad (2.4)$$

As shown in equation 2.4 the codeword is created by a simple multiplication of the generator polynomial and the message polynomial. Similar to the systematic CRC the remainder of the division of C(x) by G(x) determines whether there is any error in the system or not. In case of no error the message is the quotient of the long division of C(x) and G(x) polynomial.

2.6.2. Error detection in CRC

The strength of any detection scheme is evaluated by its ability of detecting a range of errors. The detection of error by CRC depends on the chosen generator polynomial. Equation (2.5) shows a condition were error has been introduced to the CRC. In Equation (2.6) the remainder of $C(x)/G(x)$ is always zero, if the remainder of $E(x)/G(x)$ is also zero then the CRC is not able to detect E(x) because it means the remainder of $C'(x)/G(x)$ is zero. Therefore, in order to have error detection for each certain error pattern the remainder of $E(x)/G(x)$ should not be zero. The detection of each error pattern depends on the type of the generator polynomial. CRC has the potential to detect a single bit error, two bit errors, odd number of errors, and burst errors as wide as the check bits.

$$C'(x) = C(x) + E(x) \tag{2.5}$$

$$\frac{C'(x)}{G(x)} = \frac{C(x)}{G(x)} + \frac{E(x)}{G(x)} \tag{2.6}$$

- Single Bit Error Detection

The single bit error is in the form of x^i for any i (including 0). If G(x) is chosen such that E(x) is not dividable by G(x) then that generator polynomial is able to detect any 1-bit error.

- Two Bit Error Detection

If G(x) does not divide $x^i(x^{j-i} + 1)\ for\ j > i$ then it is able to detect all 2 bit errors.

- Odd Number of Error Detection

If g(x) has an even number of terms then it is able to detect any odd number of errors. Any G(x) that has a factor of (x+1) will be 0 for x=1 that indicates even number of terms in the generator polynomial.

- Burst Error Detection

In order to detect burst errors with such pattern $E(x) = x^i (x^{k-1} + ... + 1)$, any $G(x)$ with no x^i that contains a factor with a degree greater than k is able to detect burst errors.

Chapter 3. Systematic Analyses for Latching Probability of Single-Event Transients

3.1. Introduction

High-energy particles hitting on integrated circuits (ICs) result in single-event upset (SEU) and single-event transient (SET) [35, 36]. Because of dominating in submicron IC, SEUs have been extensively studied for robust memory elements [37, 38]. As technology feature size shrinks, increased clock frequency and reduced supply voltage result in more SETs being latched and creates more soft errors than before [5]. It has been predicted that, for 45 nm and blow technology nodes, SET in combination logics will be the dominant reason for soft errors in ICs [40]. Consequently, it is imperative to thoroughly study the impact of SET pulse width and injection timing on different logic functions, in order to ensure the error resilience of ICs in nanoscale technologies

Measurement results of test chips exposed to heavy ions or alpha particles indicate that the SET pulse varies from 25 ps (65nm) [41] to 700 ps (130nm) [5]. If a system operates in GHz regime, the SET pulse may be less than one clock cycle or cover multiple clock cycles. As electrical masking, logical masking, or latching window masking could filter out SETs, assessment of SET effects is more difficult than SEU. As presented in [52], the probability of SET causing a soft error is a product of (1) the probability of the particle strike generating a strong pulse that is beyond noise margin, (2) the probability of SET not being masked by logic gates, and (3) the probability of SET propagated to the setup and hold time window of the memory element connected to the combinational circuit output.

Electrical masking occurs when the injected SET pulse is attenuated and eventually filtered out by logic gates that the SET travels through. In charge deposition mechanism, the effect of SET was modeled as two exponential current pulses to predict the SET propagation [42]. FPGA emulation exploits delay quantization [43] and voltage quantization [44] to examine SET attenuations. Logical

masking means that SET-induced logic error does not affect the output of next logic gates. Take NAND2 gate as an example. If one of the inputs is '0', no matter what the other input is, the NAND2 always has output '1'. As a result, NAND2 can tolerate the SET-induced logic error on the second input. Probability-based approaches [45, 46, 48] have been widely used to estimate the soft error reduction originated from logical masking.

Latching window masking is the phenomena that, after propagation, the injected SET does not fall in the setup and hold time window of the followed memory element. SPICE/HSPICE simulations [49, 50] have indicated that the probability of latching window masking tightly depends on the SET pulse width and SET injection timing with respect to the flip-flop sampling moment. To reduce simulation time, previous work [47] tried to identify the critical SET pulse width in benchmark circuits and computation units. Unfortunately, some logic topologies, such as shallow and wide circuit like a ripple-carry adder, do not have critical SET pulse width [47]. Alternatively, FPGA-based emulation was utilized to improve the evaluation speed [9]. Although hardware emulation could be faster than HSPICE simulations, the selection of SET pulse width and SET injection timing is still random in typical evaluation processes. The conclusions drew from previous simulation/emulation-based work may not apply to other new circuit topologies. Consequently, it is urgent to produce a systematic analysis method to predict the impact of latching widow masking on circuit soft error rate (SER).

Early work [52] presented a probability model of latching window masking. If the propagated SET pulse covers the entire setup and hold timing window, that SET would result in a soft error for sure. If only a fraction of SET pulse is in the latching window, the SET latching probability is 50%. If the SET pulse is not in the latching window, that SET pulse will be filtered out. Analysis method in [53] is effective to study the situation that a single SET pulse is latched by multiple flip-flops

after being broadcasted through a high fan-out logic node. Existing works have provided a good estimation on latching window masking effects, considering SET pulse width. As indicated in [35], soft error rate has a strong dependence on timing parameters, it is important to create a model that comprehensively considers the effect of SET pulse width, clock period, setup/hold time, and logic gate delay from the faulty gates to the memory element.

In this work, we assume that electrical masking can be estimated by the existing approaches in [42–44], and effective probabilistic methods [45–48] can calculate the logical masking probability. Our focus is on improving previous analysis of latching window masking in [52, 53] by including logic gate delays and setup/hold time of the memory elements. Our key contributions are as follows:

- Propose an analytical model to predict the combined impact of SET pulse width, logic gate delay, clock period and setup/hold time on the probability of latching SET. We observe that, the SET latching probability is proportional to SET pulse width; the corresponding coefficient and constant offset depend on clock period, logic gate delay and setup/hold time.
- Provide explicit boundaries of SET pulse width, which will result in silent errors, uncertain errors and sure errors. Those boundaries are expressed as a function of logic delay, and a flip-flop's setup/hold time. Rather than randomly selecting a SET pulse width for simulation, a user can choose a few representable pulse widths indicated in our model to speed up SER assessment.
- Suggest appropriate SET injection locations for different combinational circuits. If the combinational circuit is composed of logic gates without logical masking capability, varying logic gate selection does not lead to the estimated SER. However, if the circuit contains

logic gates with logical masking capability, a logic gate nearer to the flip-flop typically results in a higher SER than farther gates.

Section 2.2 describes the proposed analytical model for the probability of latching SET pulse. In Section 2.3, the accuracy of our model is evaluated on representable combinational circuits. In Section 2.4, experimental results show that the impact of logic gate delay, SET pulse width and injection timing on SER. Conclusions are provided in Section 2.5.

3.2. Proposed Model for the Probability of Being in Latching Window

3.2.1. Definitions Used in Proposed Analytical Model

The total tested cases are categorized into three conditions: uncertain error, sure error and silent error. The probability of SET being latched is defined in (3.1).

$$P_{in\_latching\_window} = \sum_{i=1}^{3} \left(\frac{Case\ \#\ in\ different\ categories}{T_{CLK} / \Delta t_0} \right) \times weight_i \tag{3.1}$$

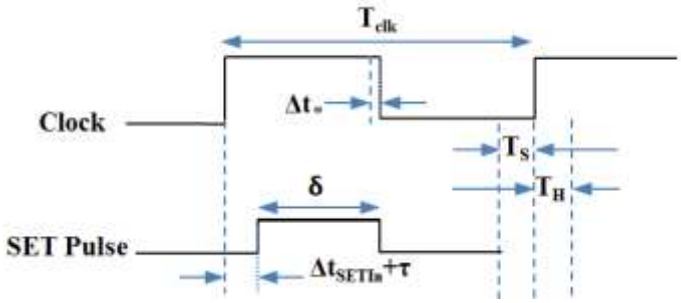


Fig. 3. 1 General definitions for SET injection in this work.

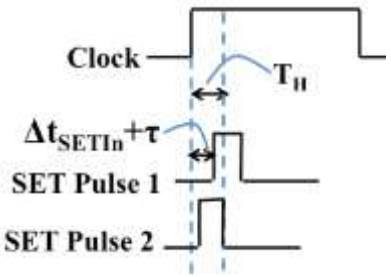


Fig. 3. 2 SET boundaries of the pulse latched by current cycle.

where $P_{in\_latching\_window}$ is the probability of a SET pulse being latched. T_{CLK} is system clock period. Δt_0 is the basic time unit of SET pulse insertion. The *weight* is equal to 0, 0.5 and 1 for silent error, uncertain error and sure error, respectively. The "*case # of different categories"* in equation (3.1) is the total number of cases that generates each category. The number of cases are found based on the location of the SET pulse respect to the clock. For instance, the number of cases to have a 50% chance of latching by the leading edge of the SET pulse is equal to $\frac{(T_S+T_H)}{\Delta t_0}$ with the assumption that the SET pulse is larger than the latching window. The overall parameters used in this work are shown in Fig. 3.1. T_S and T_H are a flip-flop's setup time and hold time, respectively. Δt_{SETIn} is the starting point of the injected SET pulse with respect to the closest clock rising edge. Δt_{SETIn} is equal to $m\Delta t_0$, where $m=[0,1,2,\ldots\lceil T_{CLK} - \Delta t_0 \rceil/\Delta t_0]$. δ is SET pulse width. τ is logic gate delay. The SET pulse shown in Fig. 3.1 represents a SET location after that SET propagates through a combinational logic and before reaches the flip-flop's input.

3.2.2. Error Categories

3.2.2.1. Uncertain Errors

When the SET pulse is propagated through the combinational logic, it will reach the latching window of a flip-flop at the primary output of the combinational circuit. According to the observation in [52], if the SET pulse does not cover the entire setup and hold time period, the probability of latching error is 50%. This is the uncertain error condition. Figs. 3.2 and 3.3 show the uncertain error cases. For simplicity, we assume hereafter T_{CLK}-T_S-T_H is larger than T_S+T_H, which is reasonable for most of sequential circuits.

As shown in Fig. 3.2, if the SET pulse injection time (Δt_{SETIn}) plus the logic delay (τ) is within T_H and the SET falling edge ends before the next latching window, that SET could be latched as a soft error. Equations (3.2) and (3.3) define the SET timing boundaries for this condition.

$$0 < \Delta t_{SETIn} + t < T_H \tag{3.2}$$

$$t + d < \Delta t_{SETIn} + t + d < T_{CLK} - T_S \tag{3.3}$$

After rearranging (2.3) and (3.3), we have a common range for Δt_{SETIn}. Since the injection step is Δt_0, the total number of cases satisfying this condition $N_{uncertain,1}$ is expressed in (3.4).

$$N_{uncertain,1} = \frac{T_H - \tau}{\Delta t_0}, \left(0 < \tau < T_H, 0 < \delta < T_{CLK} - T_S - T_H\right) \tag{3.4}$$

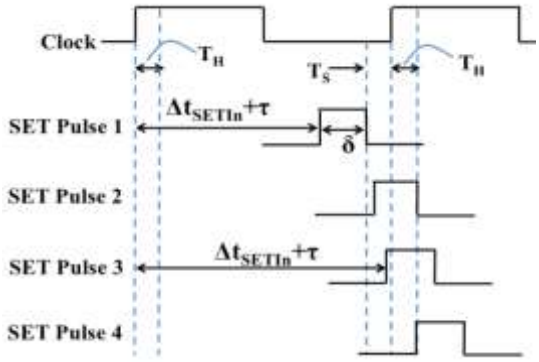


Fig. 3. 3 Boundaries of SET pulse partially latched by next clock cycle.

The SET pulses 1 and 2 in Fig. 3.3 depict the situation that the SET falling edge is in the latching window. The boundaries for SET pulses are in (3.5) and (3.6).

$$T_H < \Delta t_{SETIn} + t < T_{CLK} - T_S \tag{3.5}$$

$$T_{CLK} - T_S < \Delta t_{SETIn} + t + d < T_{CLK} + T_H \tag{3.6}$$

We rearrange (3.5) and (3.6) and obtain the total number of Δt_{SETIn} that satisfies those conditions $N_{uncertain,2}$ in (3.7).

$$N_{uncertain,2}=\begin{cases}\dfrac{T_{CLK}-\delta}{\Delta t_0} & (3.7a)\\ (0<\tau<T_H, T_{CLK}-T_S-T_H<\delta<T_{CLK}) & \\ \dfrac{T_{CLK}+T_H-\delta-\tau}{\Delta t_0} & (3.7b)\\ (T_H<\tau<T_{CLK}-T_S, T_{CLK}-T_S-T_H<\delta<T_{CLK}) & \\ \dfrac{\delta}{\Delta t_0} & (3.7c)\\ (T_H<\tau<T_{CLK}-T_S-\delta, 0<\delta<T_S+T_H) & \\ \dfrac{T_{CLK}-T_S-\tau}{\Delta t_0} & (3.7d)\\ (T_{CLK}-T_S-\delta<\tau<T_{CLK}-T_S, 0<\delta<T_S+T_H) & \\ \dfrac{T_S+T_H}{\Delta t_0} & (3.7e)\\ (0<\tau<T_{CLK}-T_S-\delta, T_S+T_H<\delta<T_{CLK}-T_S-T_H) & \\ \dfrac{T_{CLK}+T_H-\tau-\delta}{\Delta t_0} & (3.7f)\\ (T_{CLK}-T_S-\delta<\tau<T_{CLK}-T_S, T_S+T_H<\delta<T_{CLK}-T_S-T_H) & \end{cases}$$

We use the similar approach to analyze the SET pulses 3 and 4 in Fig. 3.3, and obtain the new boundaries in (3.8) and (3.9).

$$T_{CLK}-T_S<\Delta t_{SETIn}+\tau<T_{CLK}+T_H \tag{3.8}$$

$$T_{CLK}+T_H<\Delta t_{SETIn}+t+d<2T_{CLK}-T_S \tag{3.9}$$

Similarly, we can have the total case number for this situation $N_{uncertain,3}$ in (10.3).

$$N_{uncertain.3}=\begin{cases}\dfrac{\delta}{\Delta t_0} & (3.10a)\\ (T_H<\tau<T_{CLK}+T_H-\delta, 0<\delta<T_S+T_H) & \\ \dfrac{T_S+T_H}{\Delta t_0} & (3.10b)\\ (0<\tau<T_{CLK}-T_S, T_S+T_H<\delta<T_{CLK}-T_S-T_H) & \end{cases}$$

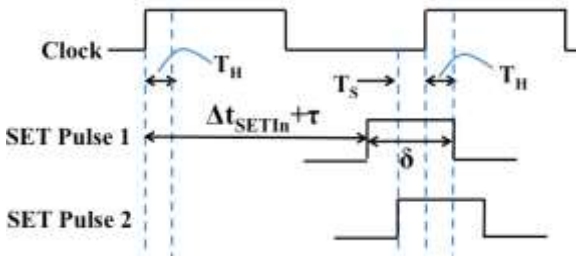


Fig. 3. 4 Boundaries of SET pulse fully latched by next clock cycle.

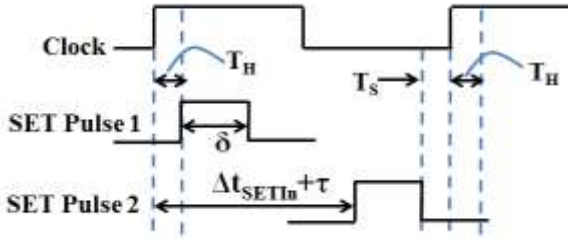


Fig. 3. 5 SET pulse boundaries for silent error in the first clock cycle.

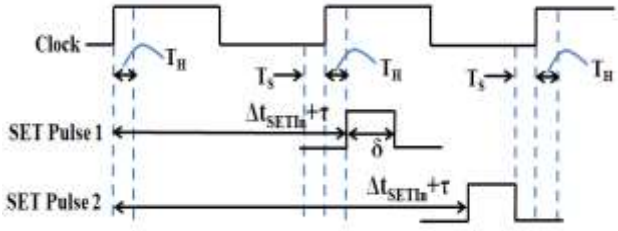


Fig. 3. 6 SET pulse boundaries for silent error in the second clock cycle.

3.2.2.2. Sure Errors

If the SET pulse covers the entire setup and hold time periods, shown in Fig. 3.4, that SET pulse will be latched for sure. The boundary conditions for pulses 1 and 2 in Fig. 3.4 are expressed in (3.11) and (3.12). The number of total cases N_{sure} that meets those boundaries is in (3.13).

$$T_H \le \Delta t_{SETIn} + \tau < T_{CLK} - T_S \tag{3.11}$$

$$T_{CLK} + T_H \le \Delta t_{SETIn} + \tau + \delta < 2T_{CLK} - T_S \tag{3.12}$$

$$N_{sure} = \frac{d - T_S - T_H}{Dt_0},\ (T_S + T_H < d < T_{CLK}) \tag{3.13}$$

3.2.2.3. Silent Errors

Silent error means that the injected SET pulse is not latched by the flip-flop following the combinational logic, which is experiencing the SET injection. We discovered two cases for the silent error. The conditions are as shown in Fig. 3.5 and Fig. 3.6. As the number of cases experiencing silent error is equal to the total test cases minus the sum of (3.4), (3.7), (3.10) and (3.13), we do not provide the derivation details here.

3.3. Overall Probability of Latching SET Pulse

We substitute (3.4), (3.7), (3.10) and (3.13) in (3.1) and obtain the probability of latching the SET pulse in (3.14). We precisely look into the boundary for each parameter and provide the close-form expression in (3.15).

$$P_{in\_latching\_window} = \frac{\left(N_{sure} + 0.5 * \sum_{i=1}^{3} N_{uncertain,i} \right)}{\frac{T_{CLK}}{\Delta t_0}} \tag{3.14}$$

For $0 < \delta < T_S + T_H$:

$$P_{in\_latching\_window} = \begin{cases} \frac{\frac{1}{2}(\delta + T_H + \tau)}{T_{CLK}}, (0 < \tau < T_H) & (3.15a) \\ \frac{\delta}{T_{CLK}}, \ (T_H < \tau < T_{CLK} - T_S - \delta) & (3.15b) \\ \frac{T_{CLK} - T_S - \tau}{T_{CLK}}, (T_{CLK} - T_S - \delta < \tau < T_{CLK} - T_S) & (3.15c) \end{cases}$$

For $T_S+T_H < \delta<T_{CLK}-T_S-T_H$:

$$P_{in\_latching\_window} = \begin{cases} \dfrac{\delta + \frac{1}{2}(T_H - \tau)}{T_{CLK}}, (0 < \tau < T_H) & (3.15d) \\ \dfrac{\delta}{T_{CLK}}, \quad (T_H < \tau < T_{CLK} - T_S - \delta) & (3.15e) \\ \dfrac{\frac{1}{2}(\delta + T_{CLK} - T_S - \tau)}{T_{CLK}}, (T_{CLK} - T_S - \delta < \tau < T_{CLK} - T_S) & (3.15f) \end{cases}$$

For $T_{CLK}-T_S-T_H <\delta< T_{CLK}$:

$$P_{in\_latching\_window} = \begin{cases} \dfrac{\frac{1}{2}(\delta + T_{CLK} - T_S - T_H)}{T_{CLK}}, \quad (0 < t < T_H) & (3.15g) \\ \dfrac{\frac{1}{2}(\delta + T_{CLK} - T_H - 2T_S - t)}{T_{CLK}}, \ (T_H < t < T_{CLK} - T_S) & (3.15h) \end{cases}$$

As predicted, the SET latching probability is a function of setup time, hold time, clock period, SET pulse width and logic gate delay. More interestingly, when $T_H< \tau < T_{CLK}–T_S-\delta$ and $0< \delta<T_S+T_H$ or $T_S+T_H< \delta <T_{CLK}-T_S-T_H$, the SET latching probability only depends on SET pulse width.

3.4. Accuracy of Proposed Model

Post-synthesis simulation in Cadence NCVerilog was used to evaluate the accuracy of the proposed model in Section II. The ultimate output of each logic network under test, without/with logical masking capability, was connected to a D flip-flop (DFF). The constructed circuits were described in HDLVerilog and synthesized in Synopsys Design Compiler with a TSMC 65nm library. The clock period was set to 160ps (less than the worst-case delay). The setup time T_S and hold time T_H are all equal to 19ps. Each data point was repeated thousand times that is sufficient to cover all cases. The SET pulse was XORed with original outputs to mimic logic gate error. Soft

error rate hereafter is defined as a ratio of the number of total error cases over the number of total test cases.

3.4.1. Verification on Logic Network without Logical Masking

We first verify the proposed model on an inverter chain, as shown in Fig. 3.7. Because inverter does not have logical masking, SER of this circuit is equal to the probability of the

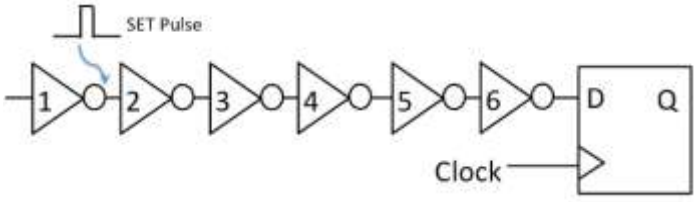


Fig. 3. 7 Inverter chain followed with a D flip-flop.

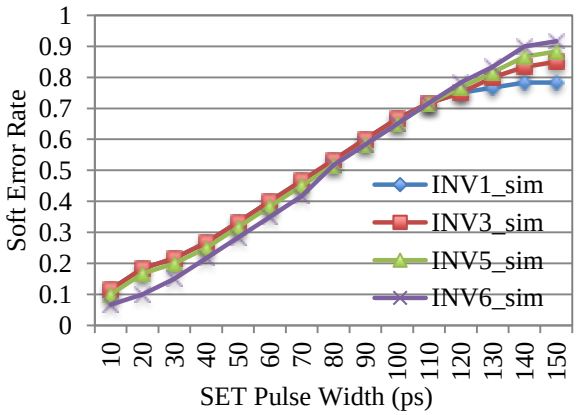


Fig. 3. 8 Simulated soft error rate for the inverter chain.

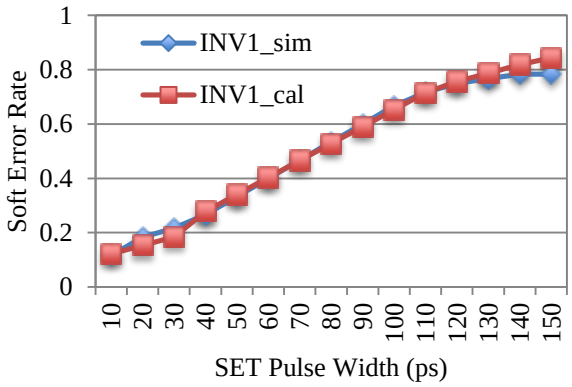


Fig. 3. 9 Comparison of simulated and derived soft error rates for the inverter 1 in Fig. 3.7.

propagated SET falling in the DFF latch window. We compared the outputs of the circuit experiencing SETs with that of the golden circuit after every SET injection.

Equation (3.15) indicates that the probability that SET enters the flip-flop latch window is typically proportional to the SET pulse width δ, despite of different coefficients and constant offsets. Equation (3.15) also shows that the logic gate delay τ affects the latch probability. Simulation results shown in Fig. 3.8 confirm our predictions: (i) *SER increases with SET pulse width*; (ii) *the SER slope increasing varies with different logic gate delay and SET pulse width*.

As the Inverters 1 and 6 represent different τ regions, we compared our derivation with simulation results. Figs. 3.9 and 3.10 show that the proposed model perfectly matches to the simulation results. We averaged the soft error rate over a wide range of SET pulse width in Table 3.1. As shown, the average accuracy of our proposed model is over 92% and up to an average accuracy of 95.7%. The high accuracy achieved here is mainly contributed by the accurate probability related to the latching window masking effect. The proposed analytical

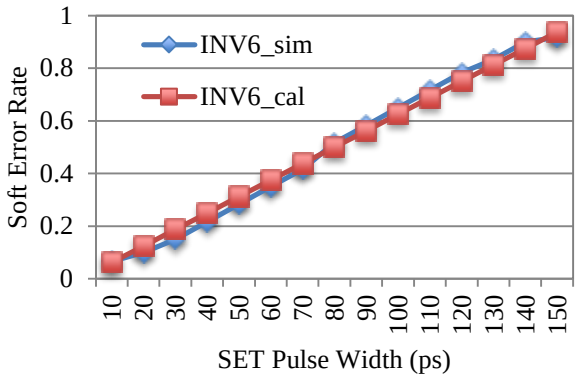


Fig. 3. 10 Comparison of simulated and derived soft error rates for the inverter 6 in Fig. 3.7.

Table 3. 1 Accuracy of proposed model

Gates in Fig. 7.3 having SET	INV1	INV3	INV5	INV6
Average Accuracy	95.7%	95.6%	93.4%	92.0%

model precisely considers various timing-dependent situations of when SETs are latched by memory elements, rather than a single SET latching moment [53]..Although they noticed the impact of latching window effects on SER estimation, Miskov-Zivanov's work [53] only considers the situation of $T_S+T_H < \delta$. Since our approach further zooms in different SET-latching timing conditions, our model achieves a better estimation accuracy for the SET latching probabilities and reliable SER prediction.

3.4.2. Verification on Logic Network with Logical Masking

Inverter chain is a special combinational logic that does not have logical masking. Some logic gates, such as NAND and NOR gates, have a capability to tolerate errors from previous gates, which is explained in Section 3.1. For logic network with logical masking, soft error rate is the product of the probability of no logical masking and the probability of SET being in latch window, as expressed in (3.16).

$$SER = P_{no\_logical\_masking} * P_{in\_latching\_window} \tag{3.16}$$

Every logic gate has its special $P_{no\_logical\_masking}$. To clearly demonstrate the impact of logical and latching window masking effects, we take a NAND network, shown in Fig.3.11, as an example. If the input port B of gates after gate *j* is '1', the SET-induced logic error on gate *j* will be propagated through the gate *j* and latched by the followed DFF. The probability of the SET pulse being propagated through gate *j* is expressed in (3.17).

$$P_{no\_logical\_masking_j} = P\{B_{j+1}=1; B_{j+2}=1; \ldots B_N=1\} = \begin{cases} \prod_{i=0}^{N-j-1} P(B_{N-i}=1), & j \in [1, N-1] \\ 0, & j = N \end{cases} \tag{3.17}$$

in which, *N* is the logic depth and it is four in Fig. 3.11. The probability of the input port B is recursively defined in (3.18).

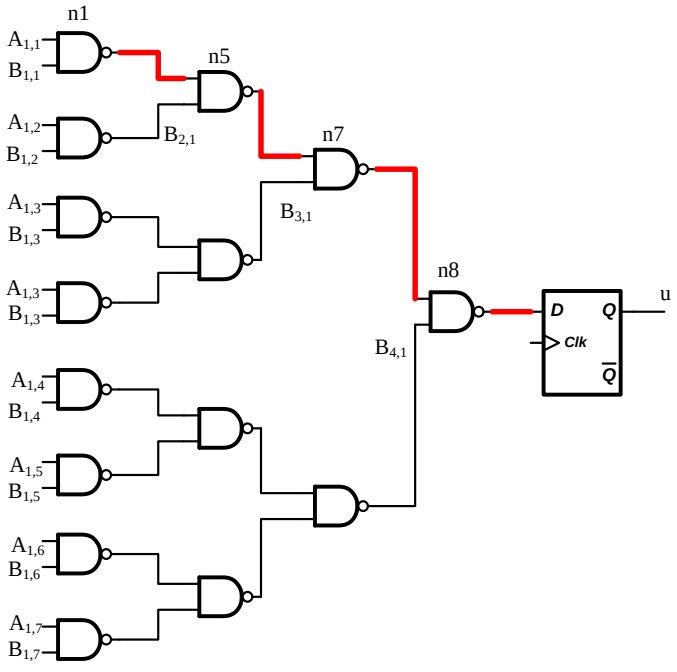


Fig. 3. 11 A NAND gate network.

Table 3. 2 Probability of no logical masking

SET injected node	$P_{no\_logic\_masking}$
n1	(3/4)*(7/16) * (207/256)
n5	(7/16)* (207/256)
n7	207/256
n8	1

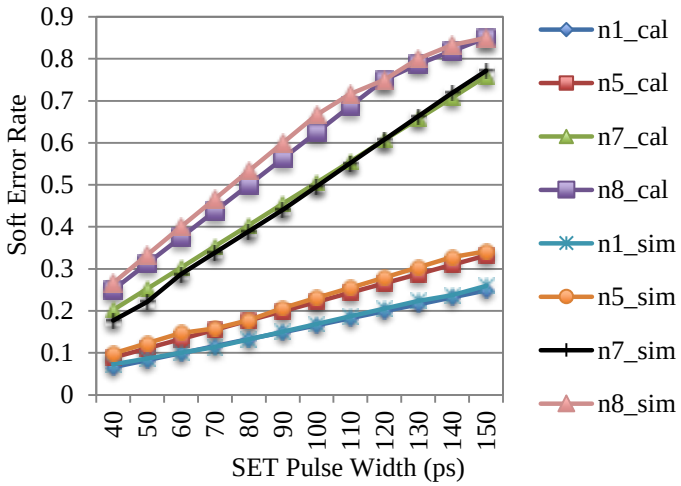


Fig. 3. 12 Soft error rate comparison between the proposed soft error model and simulation results for the NAND network.

$$P(B_m=1)=\begin{cases}1-\left(P(B_{m-1}=1)\right)^2, & m\in[2,N];\\ 0.5, & m=1.\end{cases} \tag{3.18}$$

For the very beginning input, B_1, we assume that $P(B_1=1)$ is 0.5, which is reasonable for a large amount of input patterns. According to the NAND logic function and Bayes probability theorems, we obtain the probability for a logic gate without logical masking, as shown in Table 3.2. Solve (3.17) by substituting $P_{no\_logical\_masking}$ in Table 3.2 and (3.18), we obtain the theoretical SER originated from logic gates n1, n5, n7 and n8.

In Fig. 3.12, SER calculated by the proposed model was compared to that obtained from simulations. As shown, the proposed model perfectly matches to the simulation results.

Table 3. 3 Accuracy of proposed model

Gate in Fig. 11.3 having SET	n1	n5	n7	n8
Average Accuracy	97.1%	94.2%	96.0%	95.5%

SER for each gate increases with the increasing SET pulse width. Because of the logical masking effect, difference on SER for different gates in the NAND network is more significant than that for inverter chain. As shown in Fig. 3.12, *SETs that are injected on the gates nearer to the flip-flop are more likely to result in a soft error than that on the farther locations*. Table 3.3 shows that the average accuracy of our model is over 94.2% and up to 97.1% in the NAND network.

3.5. Experimental Results

The proposed model provides a set of closed-expressions for the probability of propagated SETs falling in the register's latch window. As indicated in our model, the soft error rate has a linear relationship with SET pulse width, logic delay, and registers' setup and hold time. Our model proves that examining SER at a few boundary conditions can reflect the entire soft error dependence picture. As a result, a small number of pseudo-random simulations are sufficient to characterize SER for a specific circuit.

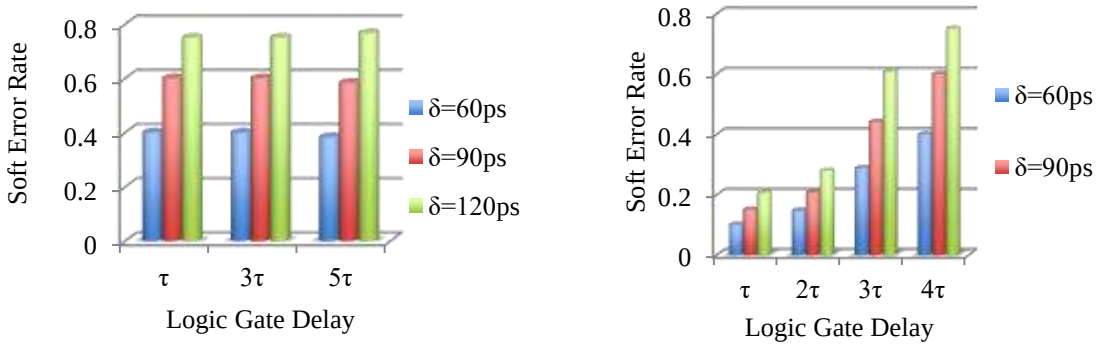
3.5.1. Impact of Logic Delay on Soft Error Rate

For a given SET injection timing with respect to the clock edge, we can use the logic delay to determine the minimum

SET pulse width that results in uncertain and sure errors. This has been explained in Section II. In addition, the logic type also affects SER, as logical masking may filter out some SET pulses. We examined the impact of logic delay on soft error rate with inverter chain (without logical masking)

and NAND gate network (with logical masking). We specifically select these circuits to demonstrate the accuracy of the proposed method, as the probabilities of logical masking and latching window masking can be independently collected both in calculation and simulations.

As shown in Fig. 3.13(a), for the circuit without logical masking, soft error rate for varying logic gate delays remains almost flat when we average all SET injection cases. In this case, soft error rate increases with the increase of SET width. As shown in Fig. 3.13(b), for the circuit with



(a) (b)

Fig. 3. 13 Impact of logic gate delay and pulse width on soft error of the circuits. (a) Inverter chain (without logical masking), (b) NAND network (with logical masking).

logical masking, soft error rate increases with logic delay and SET pulse width. Again, Fig. 3.13(b) is obtained by averaging all SET injection moments. Results shown in Fig. 3.13 confirm the correctness of the proposed model expressed in equations (3.15) - (3.18).

If the circuit under test is a hybrid of with/without logical masking gates, the dependence of soft error rate on logic gate delay is not straightforward. We performed simulations on an ITC'99 benchmark circuit, b02, and observed that the relationship of logic delay and soft error rate varies with SET pulse width. As shown in Fig. 3.14, for small SET pulse width (e.g. <230ps), the gate with smaller delay (L1) could have a higher soft error rate than the gates with larger delay (L3 and L4). For large SET pulse width (i.e., >230ps), we observe that larger gate delay leads to higher soft error rate. Wang and Agrawal [47] concluded that some circuits do not have a critical SET

pulse width. Based on our experiments and findings, it is possible that a circuit has multiple critical SET pulses, rather than a single one.

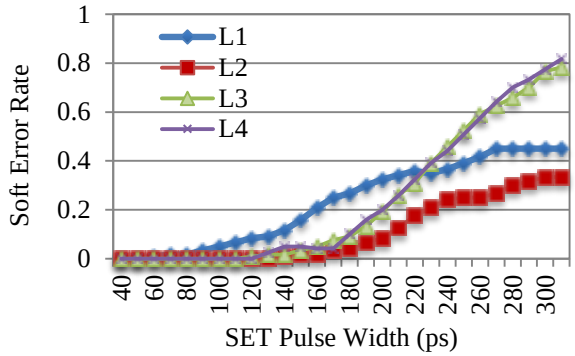


Fig. 3. 14 Impact of SET pulse width and SET injection location on soft error rate for an ITC'99 benchmark circuit, b02.

3.5.2. Impact of SET Injection Timing on Soft Error Rate

The SET injection timing (i.e., Δt_{SETIn} in Fig. 3.1) affects the probability of SET pulse entering the latch window. We examine this dependence on the NAND gate network shown in Fig. 3.11. The SET pulse width for this set experiment is 120ps. We vary Δt_{SETIn} from 20ps to 150ps, each step 10ps. As shown in Fig. 3.15, the starting point of SET pulse on n1 (the farther gate to DFF) is earlier than that of SET pulse on n8 (the nearer gate to DFF) to reach a saturated error rate. This means, a later starting point of SET pulse on the farther gate may lead to an underestimated soft error rate.

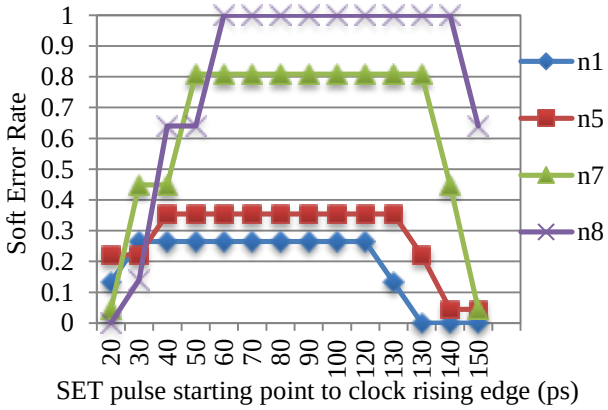


Fig. 3. 15 Impact of SET pulse injection timing on soft error rate of NAND network.

Figure 3.15 also shows that the saturated error rate obtain at SET injection on farther gates does not reflect the maximum error rate. This is because logical masking has reduced the soft error rate. To find the maximum error rate, one needs to inject SET pulses to the gate closer to registers. When the SET pulse starting point is greater then 100ps, soft error rate begins to drop because the propagated SET pulse starts to leave the latch window of the followed DFF.

Experimental results on XOR gate network (i.e., replacing NAND2 with XOR2 in Fig. 3.11) are more interesting than those for NAND gate network. As shown in Fig. 3.16, the SET pulse injected on any gate in the XOR network reaches the same saturated error rate, although the starting points for SET pulse are different. In Fig. 3.16, the soft error rate demonstrates a periodic feature. Take n5 as an example, the soft error rate first increases with the increase of Δt_{SETIn}; after saturation, the soft error rate of n5 grows again at 140ps (in Fig. 3.16(a)). The first peak of error rate is caused by the factor that SET is latched in the next cycle. The error rate growth at 140ps is because the SET pulse is latched in the cycle after next clock (e.g., pulse 2 in Fig. 3.6). As the SET pulse width for Fig. 3.16(a) is 60ps, the saturation width of soft error rate is short. When the SET pulse is bigger, the saturated error rate region is bigger, as shown in Fig. 3.16(b).

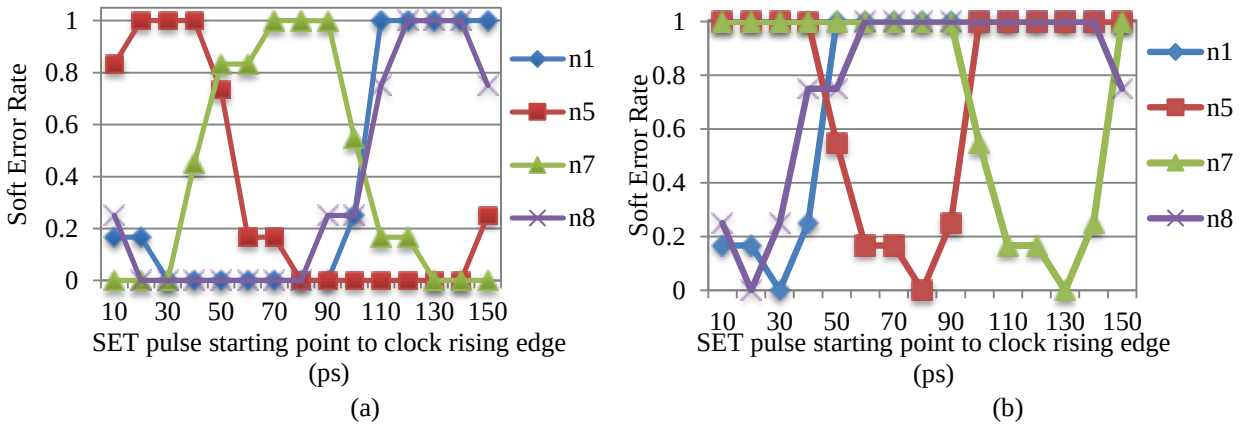


Fig. 3. 16 Impact of SET injection timing on soft error rate of XOR network. (a) SET pulse width=60ps (b) SET pulse width=120ps.

3.6. Conclusion

Investigation on the impact of SETs on soft error rate is more complicated than that on SEUs, because of the randomness of SET pulse width, SET injection timing and the affected logic type. The impacts of SET pulse width on soft error rate were typically studied via measurements on test chips. Although accurate, physical measurements are expensive and time consuming. To save cost, Monte-Carlo simulations on switch-level, circuit-level and system-level have been extensively used at the cost of long simulation time. To reduce cost and simulation time, analytical analyses are favorable.

In this work, we propose a systematic analysis method to examine the probability of SETs eventually being latched by storage elements following logic network. We prove that the probability of SET entering in latching window is one of the dominant factors for soft error rate estimation. In previous work, the probability of SET being latched is only modeled as a function of SET pulse width and clock period. As soft error rate also strongly depends on other timing parameters, our novel analytical model additionally considers logic gate delay and setup/hold time of memory elements. We determine the boundaries of SET pulse width and logic gate delay with respect to clock period and setup/hold time, and provide a set of closed-form expressions for the latching probability under different SET pulse width and logic delay conditions. Our simulation results confirm that the proposed model matches to the soft error rate obtained from random simulations. On average, our analytical model achieves an accuracy of up to 97.1%. The maximum estimation error is less than 8%.

In addition to high accuracy, the proposed analysis is also useful to speed up the evaluation process of SET effects. In our model, we identify a few representable SET pulse widths and logic delays with respect to clock period and setup/hold time for memory elements. By examining the

circuit with those representable SET pulse widths and logic delays, we can obtain the complete picture of soft error rate over a wide range of random SET injection cases. Our simulations show that, for a combinational circuit without logical masking capability, the average soft error rate does not vary with the SET injection location. For a combinational circuit with logical masking capability, the average soft error rate increases with the increasing of logic gate delay and SET pulse width. The timing distance between the SET starting point and clock rising/falling edge affects the soft error rate, as well. For a circuit without logical masking, SET injection on every gate will lead to the same saturated soft error rate; in other words, the logic gate selection does not affect the error rate estimation.

The limitation of this work is that the proposed SET latching probability is for SET pulse width less than one clock cycle. In the next chapter, we will extend our model to consider the SET lasting multiple clock cycles. More benchmark circuits will be used to validate the proposed analytical model.

Chapter 4. A New Analytical Model of SET Latching Probability for Circuits Experiencing Single- or Multiple-Cycle Single-Event Transients

4.1. Introduction

High-energy particles strikes on integrated circuits (ICs) result in soft errors [54]. While single-event upsets (SEUs) have been extensively studied primarily for memory elements, single-event transients (SETs) gain more attention than before because the increasing number of SETs is latched to cause more system failures [37, 38, 40]. Measurement results of test chips exposed to heavy ions or alpha particles indicate that the SET pulse varies from 25 ps [41] to 700 ps [5]. If a system operates in Gigahertz regime, the duration of a SET pulse varies from a portion of one clock cycle to multiple clock cycles. The increase in the ratio of SET pulse duration over clock period challenges the analysis of SET-induced soft errors, as multiple soft errors may be introduced by a single SET and thus the superimposition of multiple SETs may cause a complicate error case.

The probability of SET-induced soft error is the combination of (1) the probability that a particle strikes on an IC substrate generates a SET pulse that is strong enough to exceed the noise margin, (2) the probability of a SET is propagated through the logic network and not eliminated by logical masking effects, and (3) the probability that a SET reaches the setup and hold time window of a storage element at the end of the combinational logic circuit [59]. The three probabilities above are often referred as the probabilities of electrical masking, logical masking and latch window masking, respectively. Electrical masking effects have been investigated in [44]. Two exponential functions are used to model a SET current pulse [61]. Recently, the impact of a SET injection is modelled as a voltage source [62]. Logical masking effects have been widely studied in [48]. Electrical, logical and latch window masking are considered simultaneously in [53, 71].

As clock period and critical charge decrease, a SET pulse may cover multiple clock cycles. Analysis methods for electrical and logical masking can be reused, as these two masking effects are independent with the duration of a SET pulse. Latch window masking, unfortunately, is sensitive to the SET pulse width. As a result, the models of SET latching probability obtained from previous single-cycle analysis are not suitable now. Moreover, deep pipelining design shrinks the critical path, resulting in the setup and hold time for each pipeline stage being comparable to clock period. Consequently, the probability of a SET being latched is expected to increase. The simplified models in previous work [48, 53, 69, 65] for the latch window masking probability need to be revised to improve the SER estimation accuracy. In this work, we re-study the SET latching probability to address the emerging challenges. In addition, we also propose a fast simulation method for SET assessment that includes new SET latching scenarios induced by the increased clock frequency and shortened pipeline stage.

The remainder of this chapter is organized as follows. We summarize the related work and highlight our main contributions in Section 4.2. Section 4.3 describes the proposed analytical model for SET latch masking probability for the single-cycle and multiple-cycle SET injection scenarios. We exploit our analytical model for SET latching probability and propose a fast assessment method for the impact of SET injection in Section 4.4. In Section 4.5, the impact of SET injection location and timing, clock period and SET pulse width on the SET latching probability are evaluated. Conclusions and future work are provided in Section 4.6.

4.2. Related Work

4.2.1. Previous Work

The SET latching probability is recognized as a function of SET pulse width (δ), clock period (T_{CLK}), setup time (T_S) and hold time (T_H). The method in [65] estimates the probability of SET being latched in the storage cell, P_{ILW}, expressed in equation (4.1).

$$P_{ILW} = \begin{cases} 0, & \delta < (T_S + T_H) \\ \dfrac{\delta - (T_S + T_H)}{T_{CLK}}, & \delta \geq (T_S + T_H) \end{cases} \quad (4.1)$$

Although the P_{ILW} is categorized for two ranges of SET pulse widths, the model in (4.1) are loosely constrained as only no SET latched and full-SET latched cases are considered. The case that a SET partially covers the latch window is ignored in this work. In reality, the SET latching probability contributed by partially latched SET is not trivial. Moreover, the impact of SET injection location and SET starting timing are not reflected in (4.1). As expressed in (4.1), the P_{ILW} has a linear relation with the SET pulse width and there is not upper bound for δ. If δ is larger than the clock period, the P_{ILW} in (4.1) exceeds 1, which is not realistic for a definition of probability. An upper bound for P_{ILW} is necessary for multiple-cycle SETs.

The model in [52] improves the probability of latch window masking by assuming that the partially latched SET pulse yields a soft error with the probability of 0.5. The SET latching probability is modeled in equation (4.2), in which the first non-zero probability is the consequence of partially latched SETs.

$$P_{ILW} = \begin{cases} \dfrac{1}{2} \cdot \dfrac{T_H + T_S + \delta}{T_{CLK}}, & (\delta < T_S + T_H) \\ \dfrac{\delta}{T_{CLK}}, & (\delta \geq T_S + T_H) \end{cases} \quad (4.2)$$

In this model, three latching cases are considered: (i) if a SET pulse covers the entire latch window, that SET will be latched for sure (referred to *sure error*), (ii) if a SET pulse partially covers the latch window, that SET has a 50% chance to be latched (referred to *uncertain error*), (iii) if a SET pulse does not enter the latch window at all, that SET does not create a soft error (referred to *silent error*). As this model zooms in different situations happened in the latching window, the accuracy of the SET latching probability is improved. However, similar to (4.1), the effects of SET injection timing and multiple-cycle SETs are not considered in (4.2), either.

The recent probabilistic symbolic model [53] suggests to considering the impact of SET re-convergence on the SET latching probability for accurate SER estimation. In that model, the situations that the duration of SET pulse is less than the width of latching window or larger than the clock period are not considered.

In our previous work [67], we additionally consider the starting moment of the SET pulse with respect to the clock edge and the logic delay of the cell contaminated by a SET injection. A set of closed-form expressions for the latching probability are provided in [67] for a wide range of SET pulse widths, different logic gate delays, clock period, setup and hold time. The preliminary results in [67] indicate that the parameter boundaries in our closed-form expression have a potential to be used in fast and efficient SER evaluation. The limitations of our previous work are: the SET pulse width in our analysis is no more than one clock cycle. In this work, we will address our limitations in the previous work and demonstrate the importance of the new dependent parameters for the single- and multiple-cycle SET latching probability.

4.2.2. Our Main Contributions

The main contributions of this work are summarized as follows:

- The proposed analysis method studies the new dependency factors for SET latching probability, such as the moment when SET is injected and the logic delay of the gate received a SET, in addition to clock period, setup and hold time, and SET pulse width. We derived explicit boundaries of the SET pulse width and the gate delay that are used in the situation of silent errors, uncertain errors and sure errors.
- The proposed analytical model for the SET latching probability differentiates the condition of single-cycle SET injection from that of multiple-cycle SET injection. Previous models cannot be used for the scenarios of multiple-cycle SET injection. We zoomed in the conditions that cause uncertain errors and studied the combination of different error types created by one SET injection. Our analysis and simulation results confirm that the SET latching probability reaches a saturation point after a threshold. Our derivation explicitly indicates the dependency factor for the threshold point.
- We propose a SET injection method that exploits the boundaries of dependent parameters for SET latching probability to estimate the soft error rate. As our method is capable of estimating soft error rate through limited SET injection locations and SET pulse width, our entire SER evaluation process is time-efficient. This method facilitates the soft error rate evaluation on the gate level with affordable simulation time. Compared with existing approaches, our method does not require additional logic gate format change for symbolic analysis and only choose very limited SET injection locations.

4.3. Proposed Latching Window Masking Probability

The terminologies and symbols used in the following analysis are depicted in Fig. 4.1. τ_0 is the starting point of the injected SET pulse with respect to the closest clock rising edge. τ_{gate} is the logic gate delay from the beginning of a critical path to the gate receiving a SET pulse. The SET pulse

shown in Fig. 4.1 represents a SET location after that SET propagates through a combinational logic and before reaches the storage element's input.

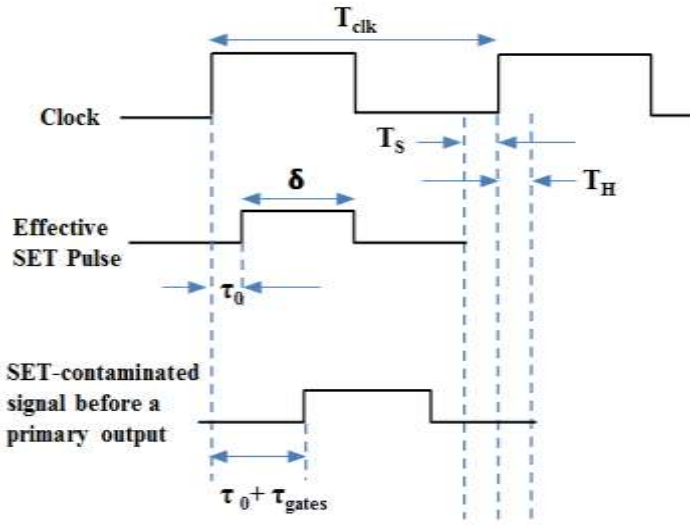


Fig. 4. 1 Parameters used in the analysis of SET injection in this work.

Assume a SET pulse is injected into a logic chain followed with a D-flip-flop (D-FF), as shown in Fig. 4.2. An input signal for the circuit in Fig. 4.2 takes τ_{gate1}, τ_{gate2}, τ_{gate3} and τ_{gate4} to reach the

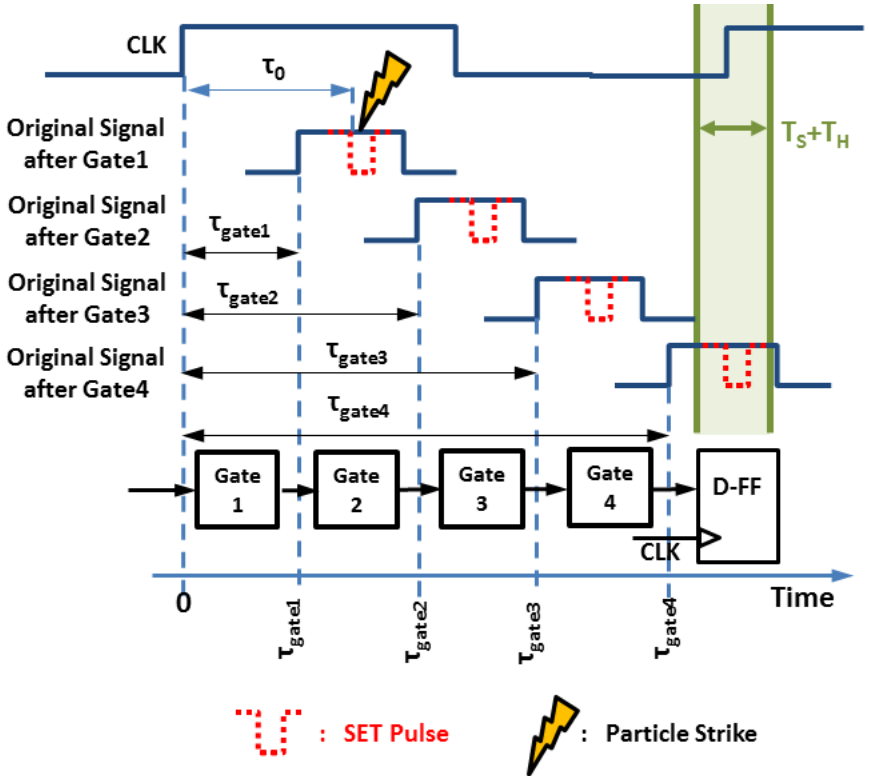


Fig. 4. 2 SET injection on a combinational logic followed with a D flip-flop.

output of Gate 1, Gate2 and Gate3 and Gate4, respectively. If a SET pulse is injected in one of the gates, the logic of the original signal is pulled down, as shown in the dotted-line in Fig. 4.2. When no logical masking happens on the affected signal, the logic dent may enter the D-FF latch window and cause a soft error. As indicated in Fig. 4.2, the starting point of the SET pulse and the pulse width are critical in determining whether the SET pulse will be latched or not. SETs injected on gates farther from the D-FF are more sensitive to smaller τ_0, as the small τ_0 plus the rest of propagation delay is close to the path length to reach the latch window. For the gates near the memory cell, a SET injection with a larger τ_0 is desired in order to ensure that SET to be latched.

4.3.1. Latching Single-Cycle SETs

In our previous work [67], we provided the close-form expressions for the SET latching probability, as well as the explicit boundaries for each dependent parameter. For readers' convenience, we copied them in equation (4.3). As indicated in equation (4.3), the SET latching probability is a function of the SET pulse width, setup and hold time, clock period, and the gate delay. When $T_H < \tau_{gate} < T_{CLK} - T_S - \delta$ and $0 < \delta < T_S + T_H$ or $T_S + T_H < \delta < T_{CLK} - T_S - T_H$, the SET latching probability only depends on SET pulse width and clock period. This sub-equation is consistent with previous models expressed in equations (4.1) and (2.4).

For $0 < \delta < T_S + T_H$:

$$P_{ILW} = \begin{cases} \dfrac{\frac{1}{2}\left(\delta + T_H + \tau_{gate}\right)}{T_{CLK}}, \ (0 < \tau_{gate} < T_H) & (4.3a) \\ \dfrac{\delta}{T_{CLK}}, \quad (T_H < \tau_{gate} < T_{CLK} - T_S - \delta) & (4.3b) \\ \dfrac{T_{CLK} - T_S - \tau_{gate}}{T_{CLK}}, (T_{CLK} - T_S - \delta < \tau_{gate} < T_{CLK} - T_S) & (4.3c) \end{cases}$$

For $T_S+T_H < \delta<T_{CLK}-T_S-T_H$:

$$P_{ILW} = \begin{cases} \dfrac{\delta + \frac{1}{2}(T_H - \tau)}{T_{CLK}}, (0 < \tau_{gate} < T_H) & (4.3d) \\ \dfrac{\delta}{T_{CLK}}, \ (T_H < \tau_{gate} < T_{CLK} - T_S - \delta) & (4.3e) \\ \dfrac{\frac{1}{2}(\delta + T_{CLK} - T_S - \tau_{gate})}{T_{CLK}}, (T_{CLK} - T_S - \delta < \tau_{gate} < T_{CLK} - T_S) & (4.3f) \end{cases}$$

For $T_{CLK}-T_S-T_H <\delta< T_{CLK}$:

$$P_{ILW} = \begin{cases} \dfrac{\frac{1}{2}(\delta + T_{CLK} - T_S - T_H)}{T_{CLK}}, \ (0 < \tau_{gate} < T_H) & (4.3g) \\ \dfrac{\frac{1}{2}(\delta + T_{CLK} - T_H - 2T_S - \tau_{gate})}{T_{CLK}}, (T_H < \tau_{gate} < T_{CLK} - T_S) & (4.3h) \end{cases}$$

4.3.2. Latching Multiple-Cycle SETs

In this work, we extend our analysis to the latch probability for multiple-cycle SETs. In the case of single-cycle SETs, we categorize the soft errors induced by SETs into three types: uncertain errors, sure errors, and silent errors [52]. If the SET pulse does not cover the entire setup and hold time period, the probability of latching SET is 50%. We refer it as the *uncertain error* condition (i.e. 50% error). If the SET pulse covers the entire setup and hold time periods, that SET pulse will be latched for sure. We name the 100% SET latching as *sure error* (i.e. 100% error). Silent error means that the injected SET pulse is not latched by the D-FF (i.e. 0% error). We name the silent error as 0% error. For a multiple-cycle SET, we extend the error type to six categories: 50%-50%, 50%-100%, 100%-50%, 100%-100%, 0%-100%, and 100%-0%, in which the first and second numbers represent the probabilities of latching the SET in the following two cycles.

We use a 50%-100% SET latching as an example to introduce our analysis method for the estimation of SET latching probability. According to the definition of a 50%-100% SET error, the beginnig of the SET pulse is partially in the first latch window and the rest of SET pulse remains through the next clock cycle (i.e. covering the entire second latching window). Figure 4.3 shows the 50%-100% error caused by a multiple-cycle SET. The boundaries for each parameter are expressed in equations (4.4) and (4.5), which describe the boundaries for the SET starting and ending edges shown in Fig. 4.3.

$$\mathrm{T_{CLK}} - \mathrm{T_S} < \tau_0 + \mathrm{\tau_{gate}} < \mathrm{T_{CLK}} + \mathrm{T_H} \tag{4.4}$$

$$2T_{CLK} + T_H < \tau_0 + \tau_{gate} + \delta < 3T_{CLK} - T_S \tag{4.5}$$

We arrange equations (4.4) and (4.5) through moving τ_{gate} to the two sides of the inequality and obtain the boundary of SET injection time τ_0, as expressed in equations (4.6) and (4.7), respectively. We label the four boundary conditions, (a), (b), (c) and (d). By comparing (a)-(d), we conclude the boundary for $\mathrm{\tau_{gate}}$ and δ to ensure the SET pulse resulting in the 50%-100% SET latching condition.

$$(a)\colon \mathrm{T_{CLK}} - \mathrm{T_S} - \mathrm{\tau_{gate}} < \tau_0 < (\mathrm{b})\colon \mathrm{T_{CLK}} + \mathrm{T_H} - \mathrm{\tau_{gate}} \tag{4.6}$$

$$(c)\colon 2T_{CLK} + T_H - \tau_{gate} - \delta < \tau_0 < (d)\colon 3T_{CLK} - T_S - \tau_{gate} - \delta \tag{4.7}$$

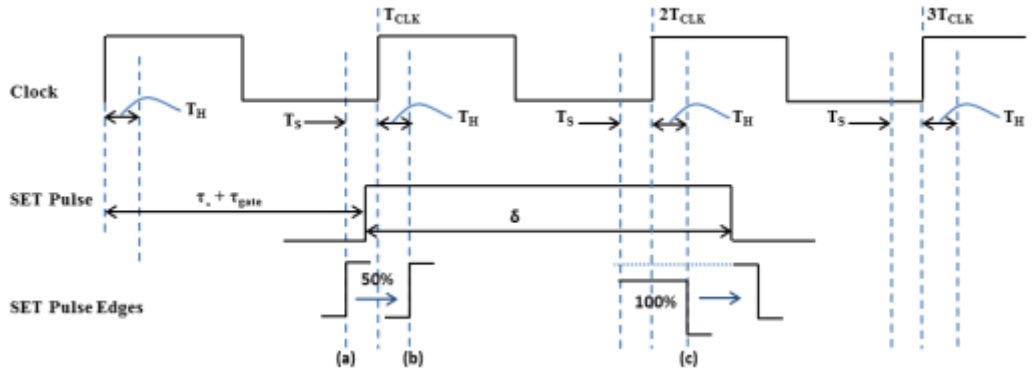


Fig. 4. 3 The multi-cycle SET has entered in to two latch windows, partially covering the first latch.

window (50% chances of error latching) and completely covering the second latch window (100% error latching).

All possible overlap τ_0 ranges in equations (4.6) and (4.7) need to be examined. For instance, to satisfy Case 1 in Fig. 4.4, we perform the comparison expressed in equation (4.8). To compute the number of cases shown in the shadow area, we need further ensure the conditions of (b) and (c) all greater than zero. Consequently, we obtain the τ_{gate}boundary, as expressed in equation (4.9).

$$\left.\begin{array}{l}(c) > (a):\ T_{CLK} + T_H + T_S > \delta \\ (b) > (c):\ \delta > T_{CLK} \\ (d) > (b):\ 2T_{CLK} - T_H - T_S > \delta \\ (d) > (a):\ 2T_{CLK} > \delta\end{array}\right\} \rightarrow T_{CLK} < \delta < T_{CLK} + T_H + T_S \tag{4.8}$$

$$\left.\begin{array}{l}(b) > 0:\ T_{CLK} + T_H > \tau_{gate} \\ (c) > 0:\ 2T_{CLK} + T_H - \delta > \tau_{gate}\end{array}\right\} \rightarrow\ \tau_{gate} < T_{CLK} - T_S \tag{4.9}$$

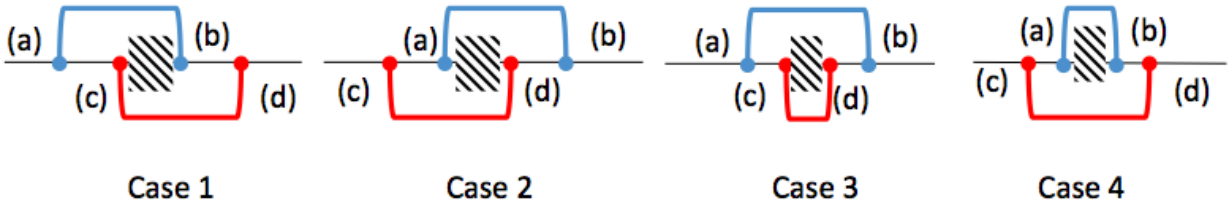


Fig. 4. 4 Four boundary comparison scenarios that determine the number of lathed SET cases. (a)-(d) are four boundary conditions in equations (4.6) and (4.7). The shadow area represents the overlapped range defined by (a)-(d).

Based on the defined conditions for δ and τ_0 in equations (4.8) and (4.9), we can obtain the number of cases that result in such latching condition in equation (4.10), in which $\Delta t_\circ$ is the step size between two consecutive τ_0s in simulations. Finally, we obtain the probability of 50%-100% SET latching, as expressed in equation (4.11).

$$N_{50\%-100\%} = \frac{(b) - (c)}{\Delta t_\circ} = \frac{\delta - T_{CLK}}{\Delta t_\circ} \tag{4.10}$$

$$P_{50\%-100\%} = \frac{\frac{\delta - T_{CLK}}{\Delta t_{\circ}}}{\frac{T_{CLK}}{\Delta t_{\circ}}} = \frac{\delta - T_{CLK}}{T_{CLK}} \tag{4.11}$$

Table 4. 1 Latching probabilities for the multiple-cycle SETs leading to different soft error categories.

Latching condition	Corresponding δ and τ boundaries for each latching case	Error latching probability
50%-50%	$T_{CLK} < \delta < T_{CLK} + T_H + T_S$ $\tau_{gate} < T_{CLK} - T_S$	$\frac{T_{CLK} + T_H + T_S - \delta}{T_{CLK}}$ (4.12)
50%-100%	$\mathrm{T_{CLK}} < \delta < \mathrm{T_{CLK}} + \mathrm{T_H} + \mathrm{T_S}$ $\tau_{gate} < \mathrm{T_{CLK}} - \mathrm{T_S}$	$\frac{\delta - \mathrm{T_{CLK}}}{\mathrm{T_{CLK}}}$ (4.13a)
	$\mathrm{T_{CLK}} + \mathrm{T_H} + \mathrm{T_S} < \delta < 2\mathrm{T_{CLK}} - \mathrm{T_H} - \mathrm{T_S}$ $\tau_{gate} < \mathrm{T_{CLK}} - \mathrm{T_S}$	$\frac{\mathrm{T_H} + \mathrm{T_S}}{\mathrm{T_{CLK}}}$ (4.13b)
	$2\mathrm{T_{CLK}} - \mathrm{T_H} - \mathrm{T_S} < \delta < 2\mathrm{T_{CLK}}$ $\tau_{gate} < \mathrm{T_{CLK}} - \mathrm{T_S}$	$\frac{2\mathrm{T_{CLK}} - \delta}{\mathrm{T_{CLK}}}$ (4.13c)
100%-50%	$T_{CLK} < \delta < T_{CLK} + T_H + T_S$ $\tau_{gate} < 2T_{CLK} - T_S - \delta$	$\frac{\delta - T_{CLK}}{T_{CLK}}$ (4.14a)
	$T_{CLK} + T_H + T_S < \delta < 2T_{CLK} - T_H - T_S$ $\tau_{gate} < 2T_{CLK} - T_S - \delta$	$\frac{T_H + T_S}{T_{CLK}}$ (4.14b)
	$2T_{CLK} - T_H - T_S < \delta < 2T_{CLK}$ $\tau_{gate} < T_H$	$\frac{2T_{CLK} - \delta}{T_{CLK}}$ (4.14c)
100%-100%	$T_{CLK} + T_H + T_S < \delta < 2T_{CLK}$ $\tau_{gate} < 2T_{CLK} + T_H - \delta$	$\frac{\delta - T_{CLK} - T_H - T_S}{T_{CLK}}$ (4.15)
0%-100%	$T_{CLK} < \delta < 2T_{CLK} - T_H - T_S$ $\tau_{gate} < T_{CLK} - T_S$	$\frac{2T_{CLK} - T_H - T_S - \delta}{T_{CLK}}$ (4.16)
100%-0%	$T_{CLK} < \delta < 2T_{CLK} - T_H - T_S$ $\tau_{gate} < T_H$	$\frac{2T_{CLK} - T_H - T_S - \delta}{T_{CLK}}$ (4.17)

The rest of SET latching window masking probabilities (4.12)-(4.17) for other error scenarios are summarized in Table 4.1. SET latching probabilities provided in Table 4.1 are differentiated from a specific range for δ and τ_{gate}. The main difference between our analytical model and the existing model is that we have considered factors such as τ_{gate} and τ_0. By including new dependent factors in the model, we obtain more precise SET latching probabilities. Another key difference is, the parameter boundaries in Table 4.1 are more refined than other models. Our new model facilitates fast SET effect assessment as we can exploit the parameter boundaries to avoid a large amount of random simulation that yields the same SET latching probability or soft error rate.

4.4.Fast SET Injection Approach for SET Assessment

In Monte-Carlo random simulation approaches, all possible situations that result in an error need to be considered. These cases depend on the input pattern, SET pulse width, logical gates used in the circuit, SET injection time and location. However, considering all the possible cases of random simulation is time-consuming. In order to speed up the procedure by decreasing the number of test cases, we propose to use the determined specific boundaries for δ and τ_{gate}. These boundaries define a set of situations that reflect the same error latching probability. Therefore, the number of test cases depends on the number of boundaries defined for each error latching probability derivation (certainly, one can also slightly increase the test cases). For each set of logic gate delay boundary we have kept the SET pulse width fixed. Since the SET latching probability depends on the moment when SET reaches the latch window, logic gate delay and the injection time play a significant role in SET latching. Each circuit can be divided into a few number of blocks containing gates with the same logic delay boundary. In this case we can group the gates based on their distance to the memory cell. Categorizing the logic gates gives us the ability to choose limited test cases that result in a faster soft error rate (SER) estimation.

The proposed SER estimation model is presented in equation (4.18).

$$SER = \frac{\sum_{i=1}^{n}\left[(P_{LG} * P_{ILW})_i * \left(\frac{N_{gi}}{\sum_{k=1}^{n} N_{gk}}\right)\right]}{n} \quad (4.18)$$

where P_{LG} is the probability of no logical masking, P_{ILW} is the probability of no latch window masking, N_{gi} is the number of logic gates in the i^{th} logic category, and $\sum_{k=1}^{n} N_{gk}$ represents the total number of logic gates. P_{LG} is achieved from random simulations by keeping the SET pulse width constant. P_{ILW} is obtained from equations (4.3) and (4.12)-(4.17). In this work, the electrical masking has not been considered due to the gate level analysis. According to our model we have separated the entire circuit in to different blocks and each block contains gates that are in a specific range of logic delays. We also need to choose limited gates from each selection to be able to decrease the simulation time significantly. Thus, the number of selected gates in each block should be divided over the number of total logic categories to represent an average value.

4.5.Experimental Results

4.5.1. Experiment Setup

The SET injection is typically modeled with a current source expressed in equation (4.19) [15]

$$I(t) = \frac{Q_{coll}}{t_f - t_r}\left(\exp(-\frac{t}{t_f}) - \exp(-\frac{t}{t_r})\right) \quad (4.19)$$

in which, t_f is the falling time, t_r is the rising time, and Q_{coll} is the total collection charges that are deposited by a particle-strike-induced current pulse. For instance, a SET pulse is injected to the drain terminal of Gate1 in Fig. 4.2. Because of charging the output capacitor, the SET current pulse is converted to a non-ideal square voltage pulse, as shown in Fig. 4.5. As the pulse is propagated through the logic network, the SET-induced voltage pulse gradually becomes a square pulse.

Considering this characteristic, we assume the SET pulse injected to the circuit under test is a square pulse in this work.

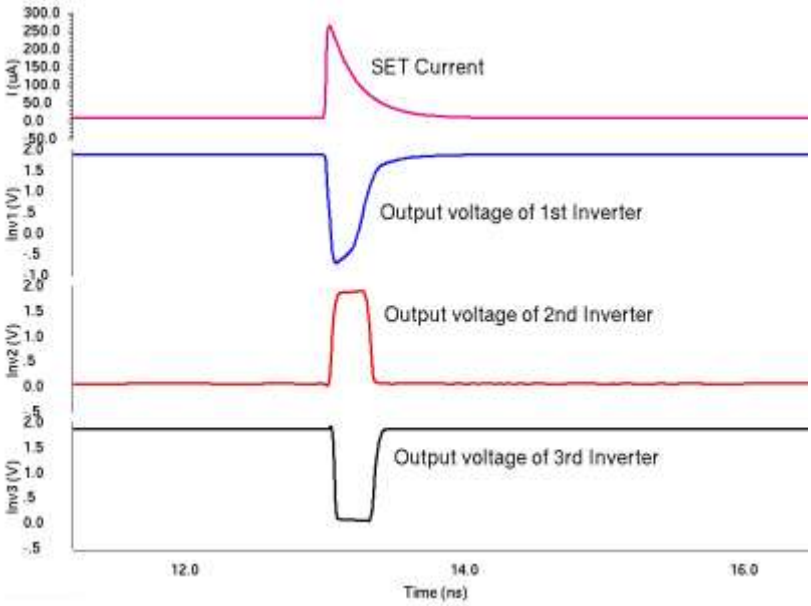


Fig. 4. 5 SET voltage pulse modeling in this work.

In the following experiment, a chain of exclusive-OR (XOR) gates and three modified ISCAS'85 benchmark circuits were synthesized in Synopsys Design Compiler. Three ISCAS'85 benchmark circuits, c432, c1355, and c6288, are modified by adding D-FFs for each primary output, in order to examine the latching probability of SETs injected in the middle of circuits. Post-synthesize simulation were performed in Cadence Verilog-XL tools to collect the D-FF output errors, circuit output errors and masked errors. A TSMC 65nm CMOS technology was used in all of the following simulations.

4.5.2. Accuracy Evaluation of Proposed Analytical Model

We first verify the proposed model on an inverter chain. Because inverter does not have logical masking capability, SER of this circuit is equal to the probability of the propagated SET falling in the DFF latch window. We compared the outputs of the circuit experiencing SETs with that of the golden circuit after every SET injection. Equation (3.4) indicates that the probability that SET enters

the flip-flop latch window is typically proportional to the SET pulse width δ, despite of different coefficients and constant offsets. Simulation results shown in Fig. 4.6 confirm our predictions: (i) SER increases with SET pulse width; (ii) the SER slope increase varies with different logic gate delay and SET pulse width. The accuracy of our proposed model is over 92% and the average accuracy is 95.7%. The high accuracy achieved here is mainly contributed by the accurate probability related to the latching window masking effect. The proposed analytical model precisely considers various timing-dependent situations of when SETs are latched by memory elements, rather than a single SET latching moment [53]. Although Miskov-Zivanov et. al. [53] noticed the impact of latching window effects on SER estimation, their work only considers the situation of $T_S+T_H < \delta$. Since our approach further zooms in different SET-latching timing conditions, our model achieves a better estimation accuracy for the SET latching probabilities and reliable SER prediction.

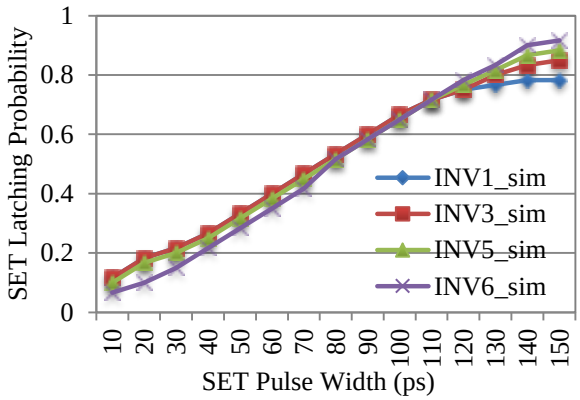


Fig. 4. 6 Simulated soft error rate for the inverter chain.

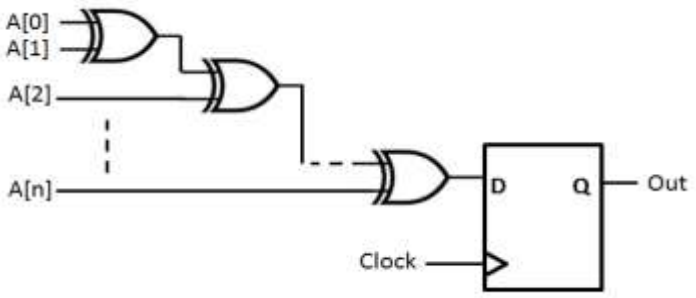
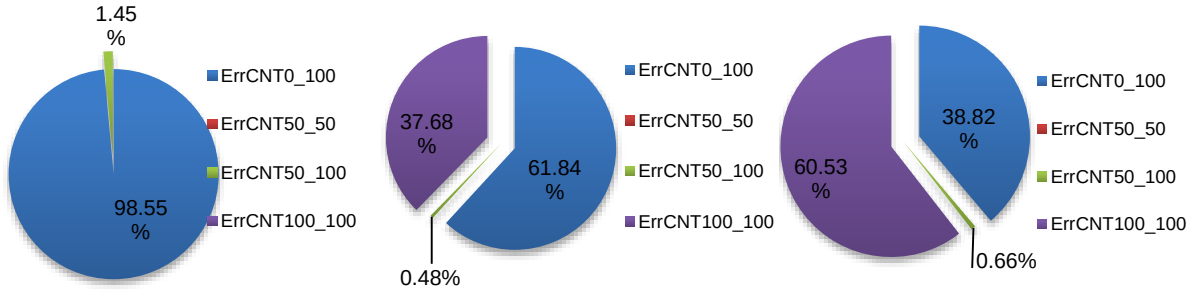


Fig. 4. 7 XOR chain.

In this work, we also verify the latching probability model for multiple-cycle SETs. As an XOR gate does not have logical masking effect, we first use XOR chain (shown in Fig. 4.7) to assess the latching window masking effect. SET pulses were injected to one XOR gate in Fig. 4.7 to flip the gate output logic. Because XOR lacks logical masking capability, the consequence of SET injection is propagated to the entry of D-FF. If the propagated SET pulse is in the range of DFF latching window, that SET is latched and causes an error.

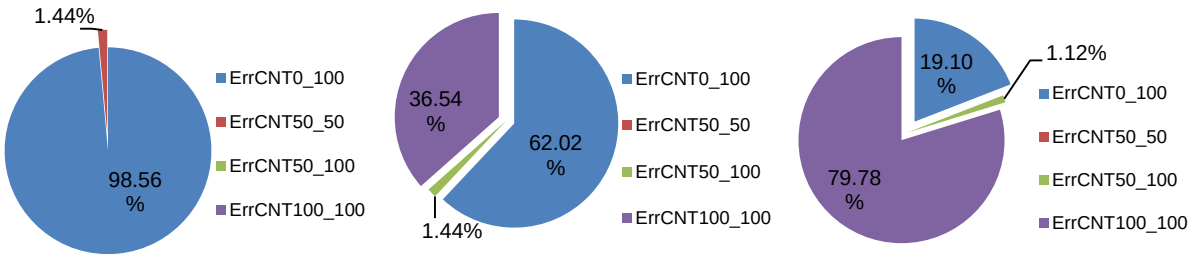
We verified the SET latching probabilities in Table 4.1 by varying SET pulse width and SET injection location. As equations (12)-(17) are the SET latching probability after averaging τ_0, we randomized the SET starting time within a clock cycle in this set of experiments. The number of latched SETs were recorded and sorted for different SET latching cases as categorized in Table 4.1. We use nine pie charts shown in Figs. 4.8-4.10 to conclude the trend of SET latching probabilities and identify the dominant SET latching cases. The SET latching probability for 100%-100% cases increases with the increase of δ. For instance, as shown in Fig.4.8 (a)-(c), the SET latching probability increases from zero to 60.53% as the SET pulse width δ increases from one clock-cycle to 1.975 clock-cycles. Similar trends are observed in Figs 4.9 and 4.10. This trend matches to the linear proportional relationship between δ and the latching probability as indicated in equation (4.15). Those results also match to our physical understanding that a SET pulse with a higher δ can inherently cover more latch windows. As a result, the probability for 100%-0% cases decreases due

to the increase of δ. This negative trend matches to the negative proportional relationship between and latching probabilities indicated in our equations (4.16) and (4.17). Comparing three figures in Figs. 4.8-4.10, we observed that the 0-100% and 100%-100% SET latching cases are alternatively dominant in the overall latching probability. We placed the 50%-100% and 100%-50% categories together in our simulation results as the SET latching probabilities for these two categories are similar. In the 65nm TSMC library used in our experiments, T_S and T_H are all equal to19ps. As the clock period T_{CLK} = 1.6ns, the probability indicated in equations (4.13) and (4.14) is relative smaller than other SET latching probabilities. This is consistent with our simulation results shown in Fig. 4.8(a), Fig. 4.9(a) and Fig. 4.10(a). The 50%-50% latching case can only be found when δ is in the smallest values. Both equation (4.12) and Figs.4.8-4.10 confirm that 50%-50% SET latching probability are the smallest portion for the overall SET latching probability.



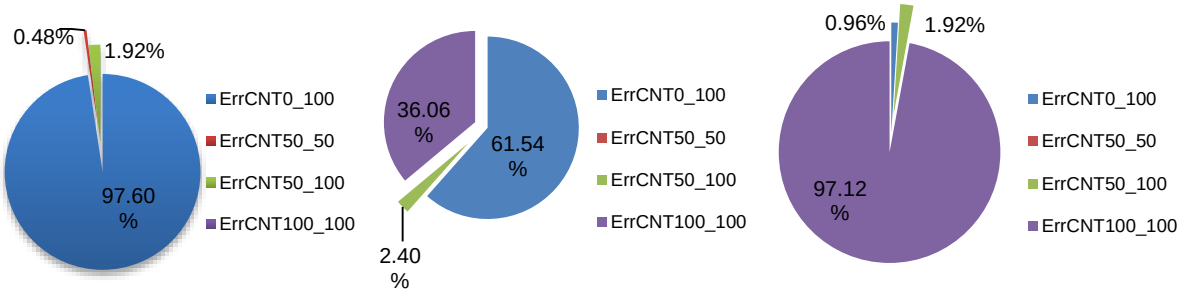
(a) (b) (c)

Fig. 4. 8 The first XOR gate in the XOR chain receiving SET pulse (a) 1 cycle, (b) 1.375 cycles, and (c) 1.975 cycles.



(a) (b) (c)

Fig. 4. 9 The fourteenth XOR gate in the XOR chain receiving SET pulse (a) 1 cycle, (b) 1.375 cycles, and (c) 1.975 cycles.



(a) (b) (c)

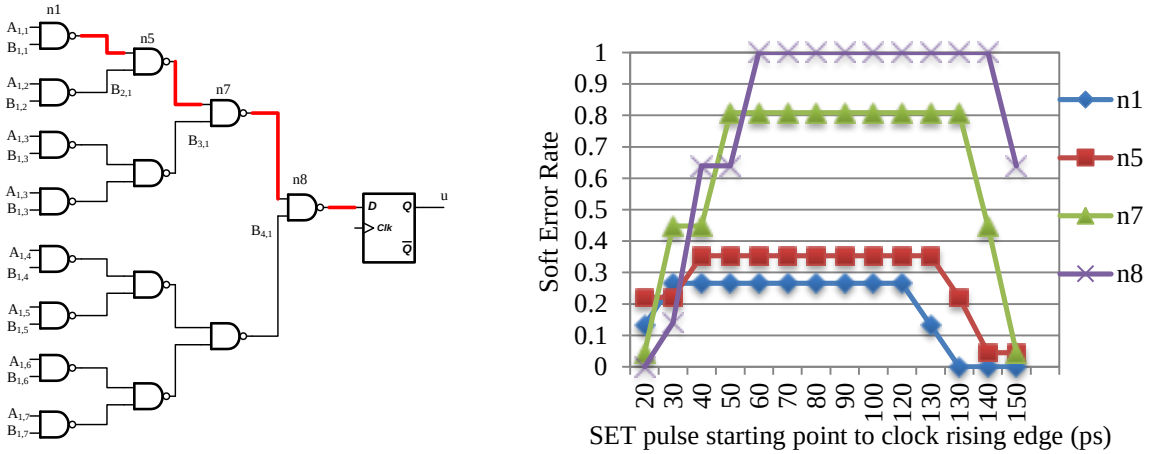
Fig. 4. 10 The 23rd XOR gate in the XOR chain receiving one SET pulse with the length of (a) 1 cycle, (b) 1.375 cycles, and (c) 1.975 cycles.

4.5.3. Impact of Dependency Factors on Latching Window Masking

4.5.3.1. Impact of SET Injection Timing

The SET injection timing affects the probability of SET pulse entering the latch window. We examine this dependence on the NAND gate network shown in Fig. 4.11(a). The SET pulse width for this set experiment is 120ps. We vary τ_0 from 20ps to 150ps, each step 10ps. As shown in Fig. 4.11(b), the starting point of SET pulse on n1 (the gate farther to D-FF) is earlier than that of SET pulse on n8 (the gate nearer to D-FF) to reach a saturated error rate. This means, a later starting point of SET pulse on the farther gate may lead to an underestimated soft error rate. Figure 4.11(b) also shows that the saturated error rate obtain at SET injection on farther gates does not reflect the maximum error rate. This is because logical masking has reduced the soft error rate. To find the

maximum error rate, one needs to inject SET pulses to the gate closer to registers. When the SET pulse starting point is greater than 100ps, soft error rate begins to drop because the propagated SET pulse starts to leave the latch window of the followed D-FF.



(a) (b)

Fig. 4. 11 Impact of SET starting time on soft error rate. (a) A NAND gate network. (b) Soft error rate for NAND network.

Experimental results on XOR gate network (i.e., replacing NAND2 with XOR2 in Fig. 4.11(a)) are more interesting than those for NAND gate network. As shown in Fig. 4.12, the SET pulse injected on any gate in the XOR network reaches the same saturated error rate, although the starting points for SET pulse are different. In Fig. 4.12, the soft error rate demonstrates a periodic feature. Take n5 as an example, the soft error rate first increases with the increase of τ_0; after saturation, the soft error rate of n5 grows again at 140ps. The first peak of error rate is caused by the factor that SET is latched in the next cycle. The error rate growth at 140ps is because the SET pulse is latched in the cycle after next clock. As the SET pulse width for Fig. 4.12 is 60ps, the saturation width of soft error rate is short.

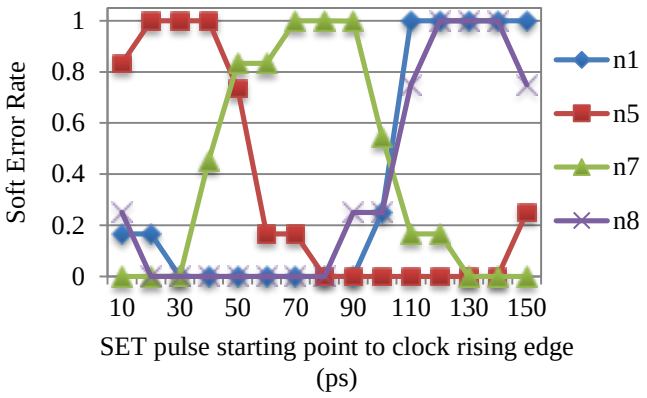
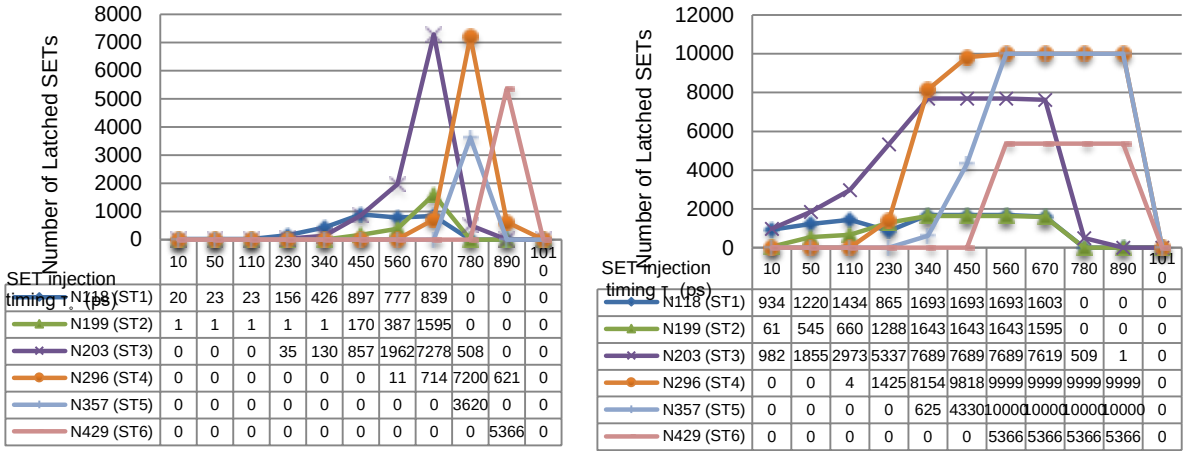


Fig. 4. 12 Impact of SET starting time on soft error rate for a XOR network.

We use one benchmark circuit, c432, to demonstrate the impact of SET injection timing on the SET latching probability. More precisely, the SET injection timing τ_0 means the starting point of SET pulse with respect to the clock edge that the DFF starts to sample new inputs. A larger τ_0 means the starting time of the injected SET pulse is later in the clock cycle. Figure 4.13 shows the number of latched SETs for eight million test cases while SET injection timings were varied. In this experiment, the clock period is 1600ps (slightly larger than the critical path delay of c432 in 65nm). Eleven data points for different τ_0s are plotted in Fig. 4.13. Six gates on one of the critical paths of c432 are selected to inject SET pulses. The subscript of ST_j (j=1…6) on the right side of the gate instance name (e.g. N118) indicates the occurrence order in time. As shown in Fig. 4.13(a), as the gate gets closer to the end of the critical path, the peak value of the latched SETs occurs at a larger τ_0. For example, N429 (ST6) reaches the peak value at 890ps; in contrast, N203 (ST3) reaches the peak value at 670ps. Since δ of 100ps in Fig. 4.13(a) is close to the step size of τ_0, only one peak point is observed. When δ increases to 500ps, the peak value of latched SETs remains same for four data points, as shown in Fig. 4.13(b). Moreover, the peak value in Fig. 4.13(b) occurs at a smaller τ_o than that in Fig. 4.13(a). The difference on the number of latched SETs for each gate is originated from logical masking effects. It is clearly shown in Fig. 4.13 that random starting timing of the SET

pulse leads can lead to a large variation on the number of latched SET pulses, thus large variation on soft error rate estimation.

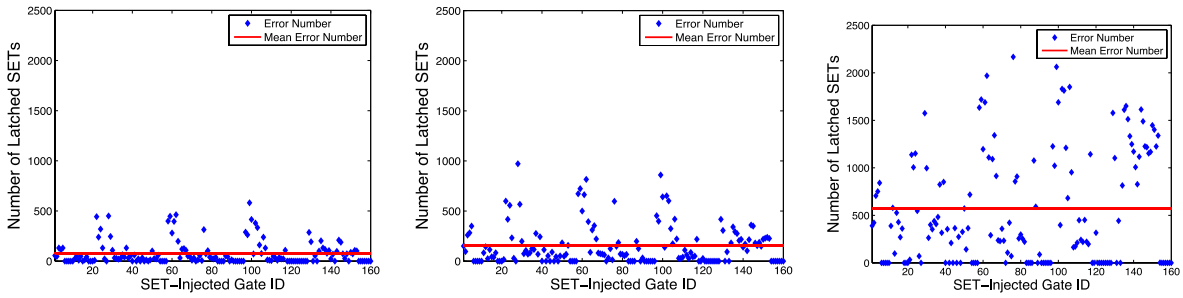


(a) (b)

Fig. 4. 13 The number of latched SETs for SET pluses being injected to different gates on a critical path of c432. (a) δ=100ps, (b) δ=500ps.

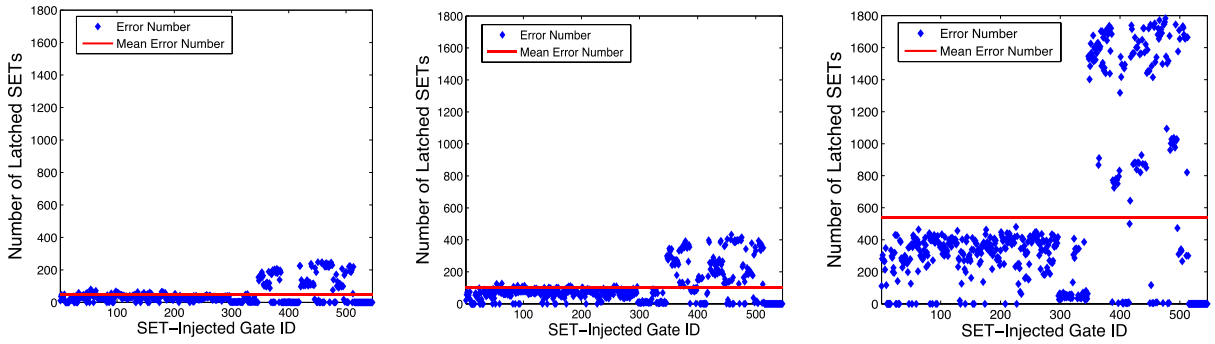
4.5.3.2. Impact of the Ratio of SET Pulse Width over Clock Period

Our derivation indicates that the SET latching probability has a strong dependency on clock period, SET pulse width, and the SET injection timing with respect to the clock period. We chose two benchmark circuits, c432 and c1355, to validate our derivation. The circuit c432 has a large variety on the applied logic gates; 68 out of 160 gates do not have inherent logical masking capability. In contrast, the circuit c1355 only have 67 out of 546 gates that cannot mask SET-induced errors via the gates themselves. Based on the worst-case delay reported by the Synthesize tool Design Compiler, we selected a slightly higher number than the worst-case delay for the clock period of each benchmark circuit and performed post-synthesize simulation in Cadence NCverilog. The clock period used in this experiment is 2.4 ns. Three different SET pulse widths were used to examine the impact of the ratio of SET pulse width over clock period on the number of SETs



(a) (b) (c)

Fig. 4. 14 Impact of the ratio of SET pulse width over clock period on SET latching for c432. (a) δ/TCLK=50/2400, (b) δ/TCLK=100/2400, (c) δ/TCLK =500/2400.



(a) (b) (c)

Fig. 4. 15 Impact of the ratio of SET pulse width over clock period on SET latching for c1355. (a) δ/TCLK=50/2400, (b) δ/ TCLK =100/2400, (c) δ/ TCLK =500/2400.

latched. Single SET pluses were injected to all possible gates in the benchmark circuit. Each data point in Figs. 4.14 and 4.15 was collected after over eight million testing cycles. We randomly chose the SET starting point for each SET pulse injection.Comparing Fig. 4.14(a)-(c), we can see that the number of latched SETs varies more significantly for a larger δ/CLK ratio than a smaller one. This means, for a given clock period, increasing SET pulse width results in increasing the probability of a SET being latched. The scattered data points can be interpreted that random SET injection location leads to a noticeable variation on the SET latching probability, as each gate has a

different logic delay to reach the flip-flop latching window and different logical masking capability. The variation of SET latching probability does not only come from different logical masking capabilities of the gate received the SET pulse, but the delay of that gate in the critical delay path also plays an important role. Figure 4.15 confirms the conclusion above.

We further validate our conclusions by varying the clock period and SET pulse width in the simulation of c432 and c1355. As shown in Fig. 4.16, no matter δ or clock period T_{CLK} is changed, a higher ratio of δ/ T_{CLK} always results in a higher standard deviation of the number of latched SETs in a given simulation time. Figure 4.16 also shows that c432 has a higher standard deviation on the number of latched SETs than c1355. One of the reasons for that is c432 has a larger variety on the implemented gate types and a higher ratio of gates w/ over w/o logical masking capability than c1355.

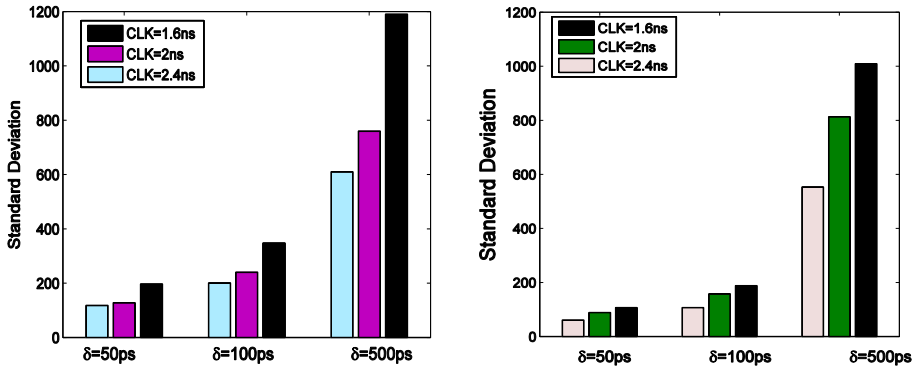


Fig. 4. 16 Standard deviation of the latched SETs by varying clock period and SET pulse width. (a) c432, (b) c1355.

4.5.3.3. Impact of Gate Delay and SET Pulse Width

We examine the impact of gate delay and SET pulse width on the SET latching probability of an XOR chain shown in Fig. 4.7. Again, there is one SET pulse injection per time in this experiment. As shown in Fig. 4.17, the SET latching probability is close to 1 (i.e. 100% latching probability),

because XOR gate does not have inherent logical masking capability. The gate delay shown in Fig. 4.17 represents the position of the XOR gate in the chain. The larger gate delay means being closer to the flip-flop at the end of the XOR chain. As shown in Fig. 4.17, the SET latching probability of the earlier gates in the XOR chain is slightly less than that of the latter gates, when the SET pulse width is just over one clock cycle long. This is reasonable because the earlier gate has a longer path to propagate the SET effect to the latching window and there is still a chance of error masking. If we sum up all possible error conditions listed in Table 4.1, we have the overall SET latching probability, expressed in equation (4.20), for the gate delay within the range between T_H and $2T_{CLK}+T_H- \delta$. In this experiment, T_{CLK}, T_S and T_H are 1.6ns, 19ps, and 19ps, respectively. The inaccuracy of our derived P_{ILW} (=0.97625) is less than 2%, compared to the simulated P_{ILW} of 0.995.

$$P_{ILW} = (T_{CLK} - T_S - T_H)/ T_{CLK} \quad (4.20)$$

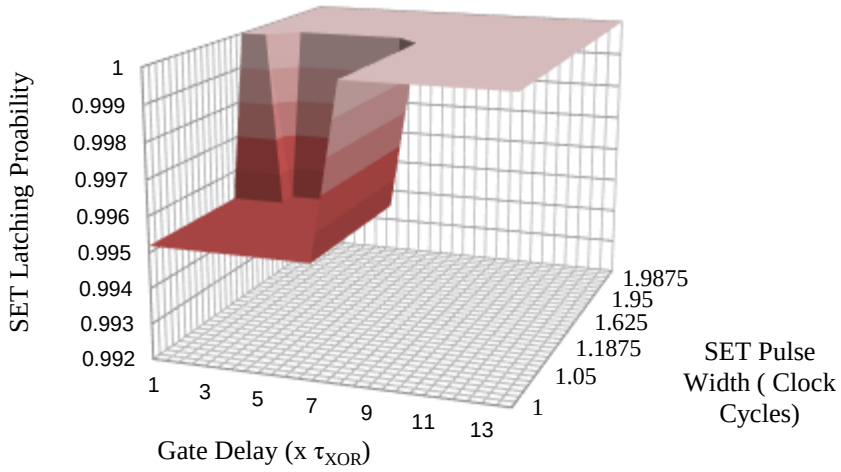
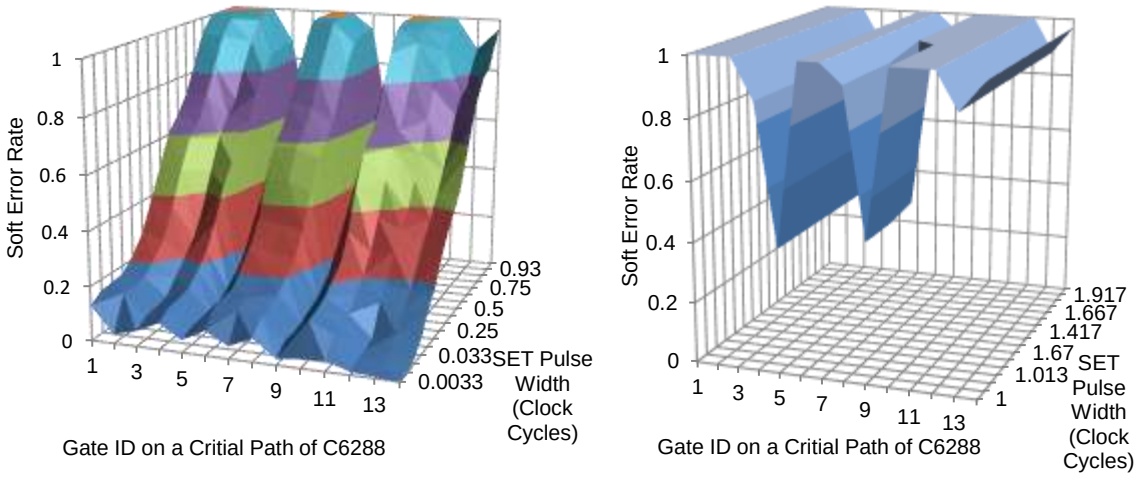


Fig. 4. 17 Simulated SET latching probability for the XOR Chain shown in Fig. 7.4.

Multiple-cycle SET injection in a circuit without inherent logical masking yields a SET latching probability close to 1. Now, we perform the similar experiment on a large-scale benchmark circuit C6288 from ISCAS'85, which has effects of SET pulse broadcast and re-convergence. We combined the SET latching probability and logical masking probability into the overall soft error

rate. We define the soft error rate as the ratio of the number errors detected by comparing the faulty version with a golden model over the total test cases. As shown in Fig. 4.18(a), generally the soft error rate increases with the increasing δ, if δ is less than one clock cycle. For some gates that have strong logical masking capability for the given input pattern, the soft error rate in Fig. 18.4(a) experiences some dents. When δ is greater than one clock cycle, the non-logical masking gate produces a latching probability of 1; two valleys shown in Fig. 4.18(b) are caused by logical masking. To zoom in the impact of multiple-cycle δ on the latching probability, we chose four random gates and randomized SET injection location. As shown in Fig. 4.19, the SET latching probability remains nearly constant when δ is slightly larger than one clock cycle. As δ keeps increasing, the SET latching probability approaches 1. The curve for random gate in Fig. 4.19 shows that the variation on the SET latching probability is up to 3.1%.



(a) (b)

Fig. 4. 18 Simulated SET latching probability for c6288 experiencing SET pulse (a) less one cycle, and (b) greater than one cycle.

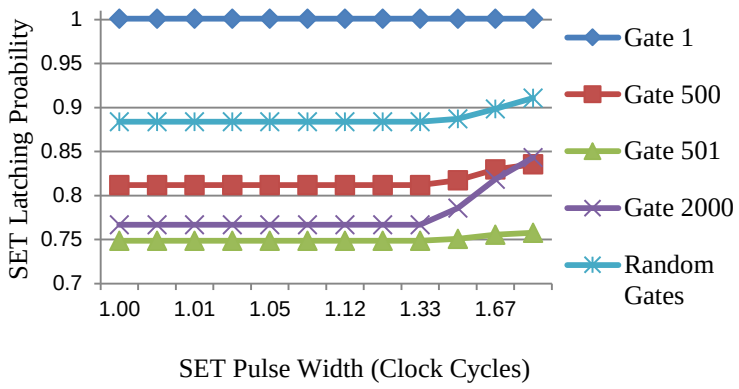


Fig. 4. 19 Latching probability for c6288 experiencing single-cycle SETs.

4.5.4. Simulation Time Reduction

We compared the simulation time for random simulation to reach a stable soft error rate (SER) with that for our semi-random method. Modified c1355 benchmark circuit was used in the experiment performed for this section. In random simulation, the gate receiving a SET pulse and the SET starting timing were random. As shown in Fig. 4.20(a), the SER obtained from Monte-Carlo (MC) random simulation has a variation of 50% (=the second data point 0.014286/ the last stable data point 0.028554. We ignore the first data point). In the proposed method, we used the same random input patterns as we used in the random simulation, and multiplied the measured SER with δ/T_{CLK} (=500/3200 in Fig. 4.20(a)) to predict the average SER. We specified the reasonable stable SER as the variation swing amplitude is within 5% of the final stable SER. The proposed method reaches the stable SER at the time of the 1165th cycle. The random simulation reaches a stable data point at the 2920th cycle. This means that the proposed method can reduce the simulation time by 60%, as the proposed method only need a small amount of simulation time to reach a stable SER. We increased the SET pulse width to 600ps and repeated the same experiment. As shown in Fig. 4.20(b), the proposed reaches the stable SER at the time of the 580th cycle. The random simulation reaches a stable data point at the 2665th cycle. This means that the proposed method can

reduce the simulation time by 78.2%. We repeated the experiments on the modified benchmark circuits c432 and c6288. As shown in Fig. 4.21, the proposed method reduces the deviation on soft error rate significantly faster than random simulation.

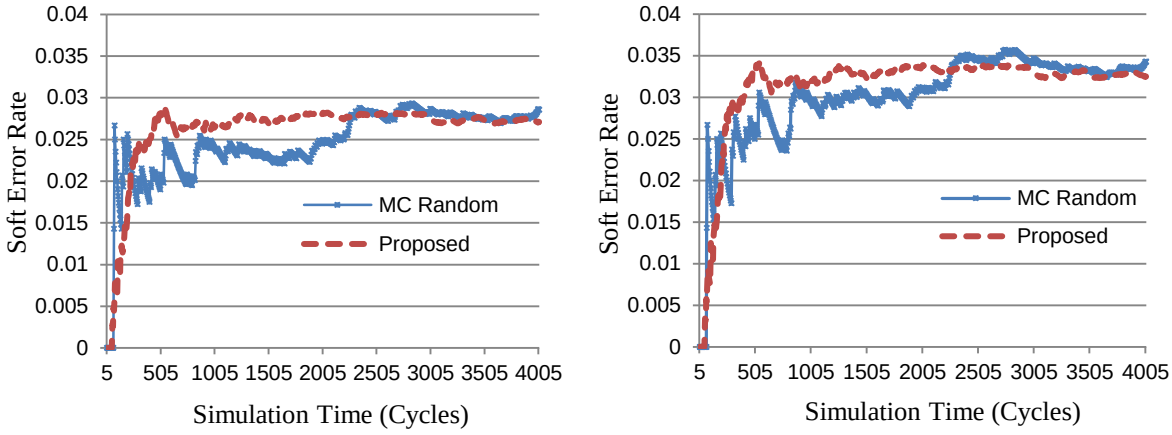


Fig. 4. 20 Reduction of simulation time for c1355. (a) δ=500ps. (b) δ=600ps.

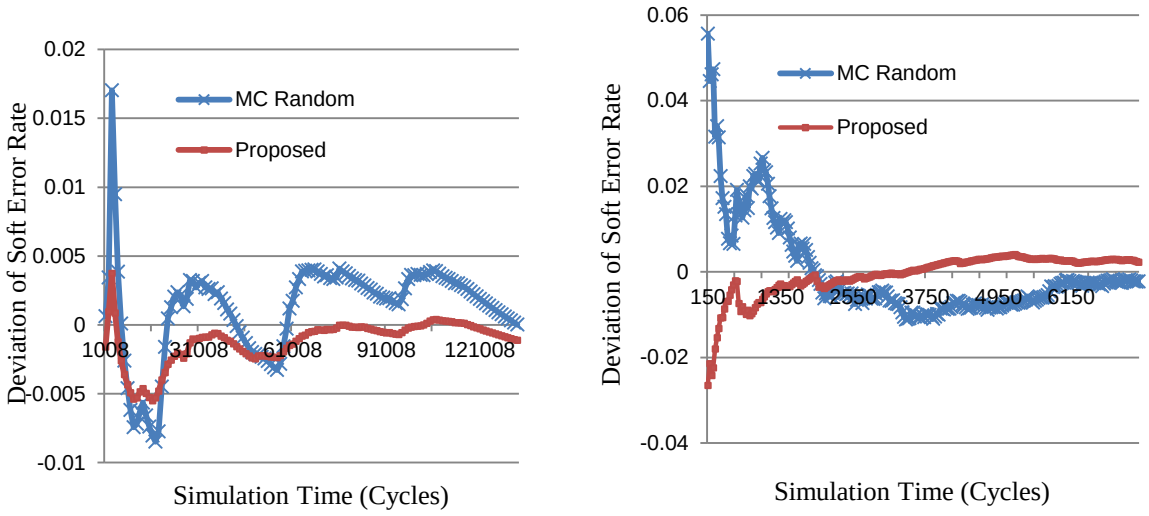


Fig. 4. 21 Deviation reduction achieved by proposed method. (a) c432, (b) c6288.

4.6.Conclusion

As single-event transients are expected to occur more often and remain longer than those in old technology nodes, studies on the SET latching probability are imperative to precisely estimate soft

error rate for integrated circuits in the nanometer regime. In previous work, the model for SET latching probability is simple and derived for singe-cycle SETs. In this work, we propose a systematic analysis method to precisely model the SET latching probability, which additionally considers SET injection location, SET starting time, and the scenarios of multiple-cycle SET injection. Our model is more comprehensive than the existing model of SET latching probability; thus, our model has a potential to improve the accuracy of soft error rate estimation.

Simulation on the circuit without inherent logical masking capability show that the proposed systematic analyses method achieves up to 97% average accuracy for single-cycle SETs, and 98% accuracy for multiple-cycle SETs. Our case studies on ISCAS'85 benchmark circuit show that the SET latching probability has strong dependency on the SET injection location (i.e. the gate delay from the beginning of a critical path) and the SET starting time with respect to a clock latching edge. Our analytical model for new dependent parameters is consistent with the trend obtained from Monte-Carlo simulation. As our semi-random simulation method fully exploits the boundaries and explicit latching probabilities indicated in the proposed model, our method reduces the SET assessment time by up to 78% in the c6288 circuit, compared with Monte-Carlo simulation. Significantly simulation time reductions are also observed during the evaluation of soft error rate for c432 and c1355 benchmark circuits.

Chapter 5. Assessing CPA Resistance of AES with Different Fault Tolerance Mechanisms

5.1. Introduction

Cryptographic algorithms have become essential for the security-critical applications such as military, government, and banking systems. The extremely high computation complexity of ciphers prevents the key recovery through brute-force guessing methods, however advanced security attacks [73-76, 14, 15] are possible to make the cryptosystem fail. Fault attacks (FAs) [73-76] compromise the cipher implementation and produce faulty ciphertexts for cryptanalysts to retrieve the encryption key. FA can be done using deliberate injection of faults into cryptographic devices by means of white light, laser beam, voltage/clock glitch, and temperature control [13]. Faults induced by fault attacks have certain similarity with the random faults caused by radiation environment. No matter what kind of fault (random or intentional) is introduced to the cryptographic device, it is imperative to implement an effective fault tolerance mechanism into the cryptosystem. Traditional fault-detection methods for cryptosystems exploit information redundancy [80, 81, 82, 83, 78], spatial redundancy [16], or time redundancy [84] to detect faults and generate an alert signal to stop the normal operation. We are referring fault detection as fault tolerance because once fault is detected, faulty intermediate state will be recomputed. Side-channel analysis attack (SCA) is another type of attacks that studies the side channel signals of the cryptographic hardware, such as power, delay, and temperature to guess the secret key. SCAs are categorized in simple power analysis (SPA) [15], differential power analysis (DPA) [15], and correlation power analysis (CPA) [14]. As CPA requires far fewer traces for recovering the key than SPA and DPA, it is the most commonly used power analysis method lately.

Countermeasures to thwart fault attack and side-channel attack are typically investigated in a separate fashion. Unfortunately, thorough investigation on how one countermeasure specifically

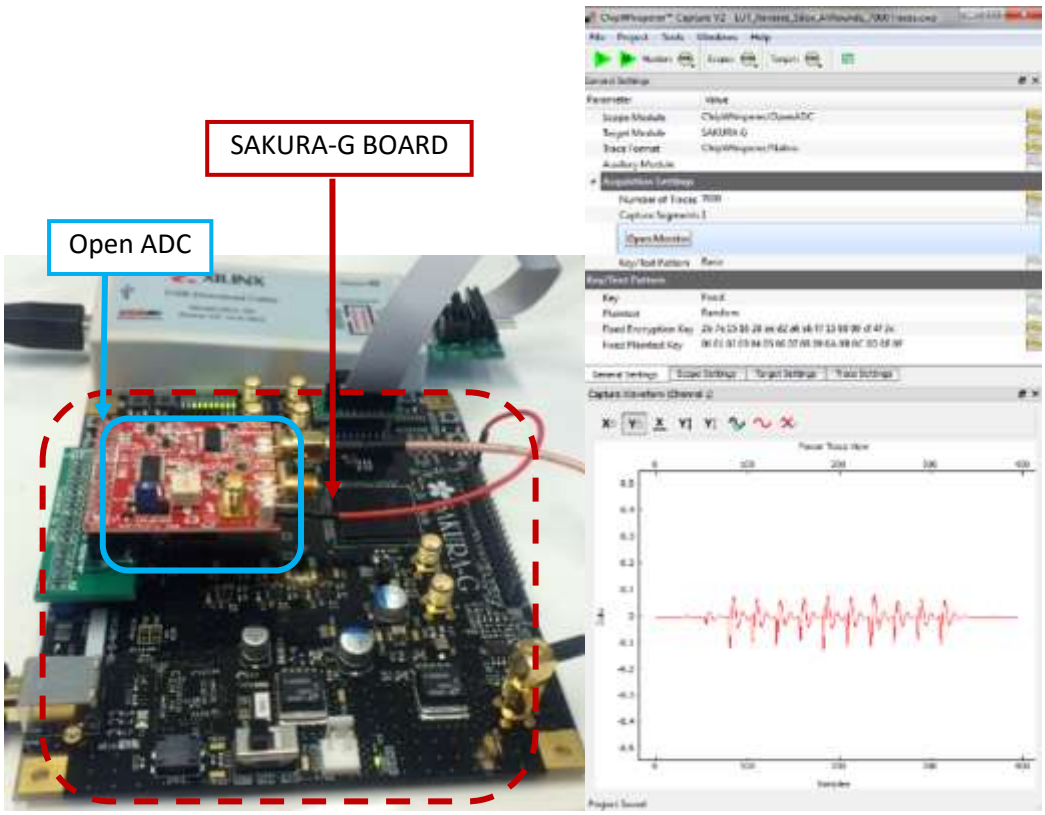
for one attack affects the efficiency of another attack is not explored much. In this work, we focus on the investigation of how fault detection (FD) mechanisms affect the efficiency of CPA attack. As Advanced Encryption Standard (AES) [72] is one of the most widespread cryptographic algorithms, we use AES as the subject in this work. The existing work [20, 66] concludes that the use of parity codes in the S-Box of AES for fault detection makes CPA succeed faster than the S-Box without any fault detection mechanisms. Unfortunately, that observation is based on the gate-level simulation on the S-Box only, rather than a real hardware emulation of the complete AES. Moreover, the power model used in the early work [20, 66] is Hamming weight, not the powerful Hamming distance. Hamming distance model is powerful because it can retrieve more number of subkeys in given number of power traces than Hamming weight model. Next to fill in the gap, we assess the CPA resistance of the AES protected with different types of fault-detection mechanisms on a FPGA platform.

In Section 5.2, we introduce a brief overview of AES structure and representable fault detection methods for AES against random and intentional faults. Section 5.3 provides the experimental setup. We present the emulation results on FPGA in Section 5.4. Finally, we conclude this work in Section 5.5.

5.2. Preliminaries

5.2.1. AES Structure

AES is a symmetric block cipher, which accepts plaintext and key (128, 192, or 256 bits) as input and generates the ciphertext after N_r iterations. The number of rounds N_r depends upon the key size. The AES encryption process is shown in Fig. 5.1. The AES algorithm's internal operations are performed on a two dimensional array of bytes called State, and each byte consists of 8 bits. The key schedule algorithm is used to scramble the key used for each round. While



(a) (b)

Fig. 5. 2 (a) SAKURA-G board with an OpenADC mounted and the Xilinx USB download cable. (b) ChipWhisperer interface capturing one power trace for a fixed key and a random plaintext.

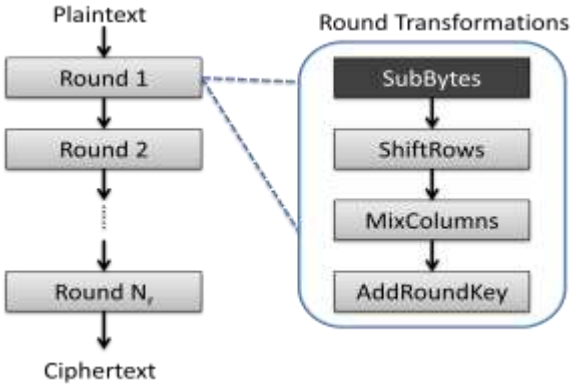


Fig. 5. 1 Encryption structure of AES algorithm [88].

ShiftRows, *MixColumns* and *AddRoundKey* are linear operations, *SubBytes* is non-linear. In this work, we use look up table implementation for *S-Box* [72].

5.2.2. Typical Fault-Tolerance Mechanisms for AES

Traditional fault-tolerance methods for cryptosystems exploit information redundancy, spatial redundancy, or time redundancy to detect faults. In this subsection, we briefly introduce three common FT methods.

5.2.2.1 Hardware Redundancy based FT

The simplest fault tolerance method is double modular redundancy (DMR) [81], in which the original function is duplicated. Comparison of the outputs from the two copies can indicate the inconsistence caused by faults. DMR can be applied at operation level, round level as well algorithm level. The DMR can be extended to triple modular redundancy (TMR), where the original function can be repeated three times followed with a majority of voter.

5.2.2.2 Parity Code based FT

Parity based fault detection techniques have been explored widely for AES cipher [80-83, 78]. The output parity bits of each transformation in every round are predicted from the inputs of the corresponding transformation [80-83]. S-Box and its parity prediction are non-linear and complex and are implemented using look up table [80, 81, 78] as well as logic gates in composite field for higher performance [83]. The approach in [78] is based on storing the one-bit predicted parity of the S-Box in a table and comparing it with the actual parity. The parity schemes presented in [82] are independent of the structures of the S-Boxes and the inverse S-Boxes.

5.2.2.3 Inverse Function based FT

Karri et al. [16] developed a systematic concurrent error detection approach for symmetric block ciphers at the register transfer level. This method exploits the inverse relationship between the encryption and decryption at the individual operation level, round level, and algorithm level.

5.2.3. Correlation Power Analysis (CPA)

Before CPA, we first need to obtain the power history of the cryptographic device consumed during the encryption process. Next, the adversary uses the intermediate state values to determine if there is a correlation between the measured power trace and the power predicted by the adopted power model. The basic equation for a CPA [77] attack is shown in eq. (5.1), where $r_{i,j}$ is the correlation coefficient at point j for the key hypothesis i, $t_{d,j}$ is the power measurement of trace number d at point j, and $h_{d,i}$ is the hypothetical power consumption of hypothesis i for trace number d, with a total of D traces. $h_{d,i}$ in eq. (1.5) is related to the power model either Hamming distance or Hamming weight [18]. A higher correlation coefficient ($r_{i,j}$) indicates a closer key guess.

$$r_{i,j} = \frac{D\sum_{d=1}^{D} h_{d,i}\, t_{d,j} - \sum_{d=1}^{D} h_{d,i} \sum_{d=1}^{D} t_{d,j}}{\sqrt{\left(\left(\sum_{d=1}^{D} h_{d,i}\right)^2 - D\left(\sum_{d=1}^{D} h_{d,i}^{\,2}\right)\right)\left(\left(\sum_{d=1}^{D} t_{d,j}\right)^2 - D\left(\sum_{d=1}^{D} t_{d,j}^{\,2}\right)\right)}} \tag{5.1}$$

Hamming weight power model was first adopted for SCA because of its simplicity. The Hamming weight model predicts the power by assuming the previous intermediate state is a zero string [18]. Later, the Hamming distance power model removes the assumption of zero string and generates a more accurate power prediction by predicting the power based on the previous state.

5.3. Our Objective and Experimental Setup

5.3.1. Study Objective

The object of this work is to assess the CPA resistance of the AES with different fault-detection (FD) methods, double modular redundancy (DMR), inverse function (inverse), and even parity check code (parity). The AES implementation interested in this work is iterative round operation. Fault-detection methods are applied to MixColumns and SubByte (i.e. S-Box hereafter), which are two representable linear and non-linear operation steps in AES.

5.3.2. Hardware and Software for Experiments

We used the SAKURA-G FPGA board that is designed for research on hardware security such

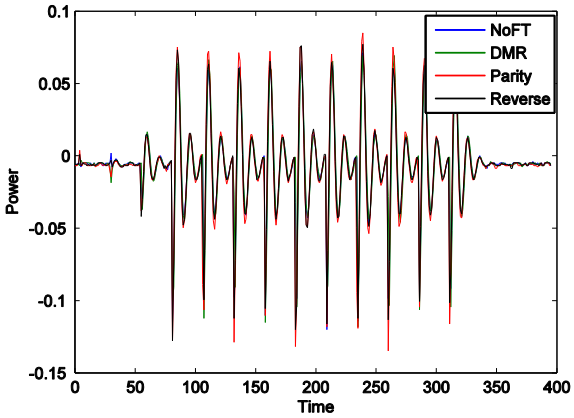


Fig. 5. 3 Power traces for AES with four different fault tolerance methods in S-Box.

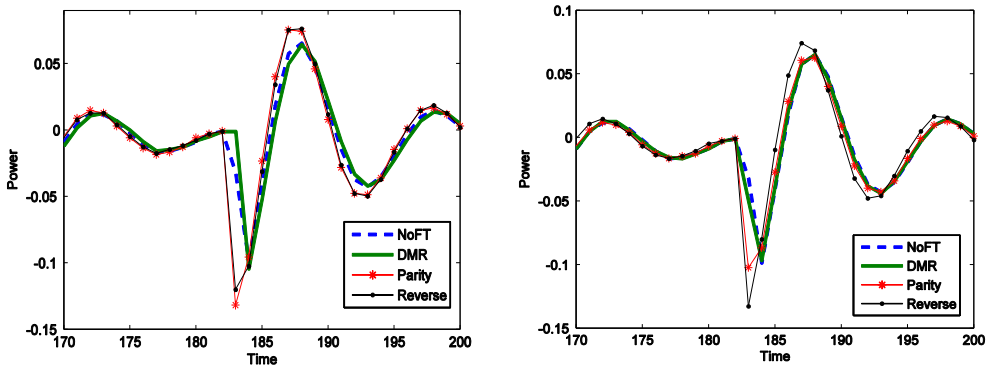


Fig. 5. 4 Zoomed-in Power traces for AES with (a) FT in S-Box, and (b) FT in MixColumn.

as Side-Channel Attacks (SCA) and Fault Attack (FA). This board contains two Spartan-6 FPGAs: one (LX75 FPGA) for a cryptographic implementation and the other one (LX9 FPGA) for power traces capturing. Verilog-HDL code for AES from [79] was downloaded to the SAKURA_G board. A Python-based *ChipWhisperer* [14] software was used to performance power trace capturing and analysis. In order to use the ChipWhisperer software and perform a synchronous sampling for collecting the power traces, we used the hardware OpenADC along with the SAKURA-G board. This Analog-to-Digital Converter was mounted right on top of the SAKURA-G board as shown in Fig. 5.2. The SAKURA-G board is connected to the PC through a USB download cable that is

supported by Xilinx iMPACT downloader to provide a serial connection for programming the FPGAs.

The whole experiments follow the same routine, which is described below: The encryption implementations are done based on a fixed key and 7,000 random plaintexts. According to a preliminary set of results, it was understood that the key could be recovered by around 7,000 power traces for the iterative AES encryption algorithm that was programmed in the SAKURA-G board. In the following experiments, we chose the 128-bit AES configuration with the look-up table for the SubByte that is also called S-Box. The number of power traces required for retrieving all the subkeys in different experiments change. In order to shorten the experiment time we kept the number of random plaintexts fixed and focused on analyzing the number of subkeys that were recovered in 7,000 power traces.

We used Hamming distance as the power model in the attack procedures. In order to prove the strength of Hamming distance over Hamming weight power model, we have provided a set of experiments using Hamming weight and Hamming distance for a same condition. Our results in Section IV.C show that Hamming distance power model is more powerful model than Hamming weight one, as the CPA with Hamming distance model can retrieve the key with less power trace.

5.4. Experimental Results for CPA on AES

5.4.1. Comparison of Single Power Traces of Different FDs

The power traces of the AES implementation with different FD methods are compared in this subsection. The same plaintext and key were applied to all the four different AES implementations: (1) no fault detection (no FD), (2) double modular redundancy based FD (DMR) [70], (3) even parity check code based FD (parity) [80], and (4) round-level inverse function based FD (inverse) [6].

Figure 5.3 shows four different power traces for ten computation rounds of the AES with and without FD in the S-Box. Due to the different fault detection mechanism, the captured power traces of the AES are different. We zoomed in the power trace details of Fig. 5.3 are in Fig. 5.4(a). As can be seen, different fault detection methods lead the occurrence of power transition peaks at the different moments. In addition, the magnitude of peak power also varies with the FD method. We repeated the experiment on the AES with FD schemes on the MixColumn transformation in Fig. 5.4(b). Comparing Figs. 5.4(a) and (b), we found out that, if the same fault detection method is applied to the different operation steps, the power peak magnitude and the time of occurrence will have different offsets. This indicates that different FD methods or different FD locations will add different levels of impact on the key retrieval process.

5.4.2. Key Retrieval Speed Comparison

In this work, we adopt partial guessing entropy (PGE) [28] as an evaluation criterion, which determines the average number of guesses that would take to estimate the correct value of a single subkey byte. The 'partial' is referred to the guessing entropy on each subkey rather than complete encryption key. The PGE of zero indicates that Subkey is perfectly known. As soon as the sum of all PGEs for AES subkey bytes is zero, no more guessing is needed and the complete key bytes are found by the CPA method. Therefore, we use the accumulated PGE (APGE) as a metric to examine when the key retrieval is completed.

5.4.2.1. PGE Variations over Number of Analyzed Power Traces

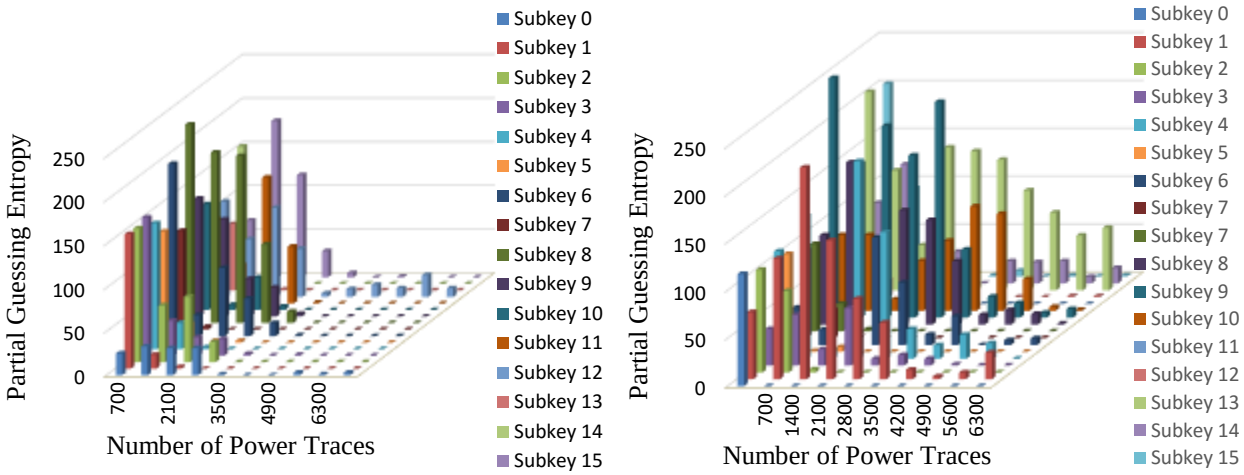
SubByte is the most important non-linear operation in the AES and can be implemented with a lookup table, so called S-Box. Fig. 5.5 shows the PGE for each subbyte key in the process of key retrieving. As each byte has 8 bits, the total number of guess attempts for each subkey byte is up to 256. If the number of required guesses decreases to 0 (i.e. PGE is zero), that means the subkey

byte has been recovered through CPA. It is expected that the PGE of the AES without FD goes to zero faster than the AES with a FD mechanism. In this case, the fault detection mechanism actually makes it more difficult to retrieve the key than the original AES implementation. Figure 5.5 indicates that the PGE for the AES without FD and with FD (parity check code) in S-Box eventually decreases, as the number of power traces used in CPA increases. But, the S-Box with parity FD needs a larger number of power traces to retrieve the total key bytes than the one without FD.

5.4.2.2. Impact of S-Box with FT on Key Retrieval Speed

We examined the impact of different fault-detection methods on the key retrieval speed. As shown in Fig. 5.6, the AES implementation with different FD schemes yields different APGE compared with no FD. This confirms that the fault-detection mechanism do NOT positively affect the efficiency of CPA. Instead, different FD methods increase or decrease the APGE at different degrees. As shown in Fig. 5.6, the parity method leads to the highest APGE at the first 1,400 traces than other methods. This means, if a limited number of power traces are available, the parity FD method makes the key retrieving more difficult than other FD methods and in need of reasonably more subkeys guess. We further observed that the cause of non-zero PGE is sometimes originated from the same subkey byte. Therefore, we used another way to assess the speed of key retrieval and verify our understanding on the impact of adding different FD methods on the strength of CPA. Therefore, we analyzed the number of subkey bytes that were recovered at each power trace interval, as shown in Fig. 5.7. For instance, given 3,500 power traces, 100%, and 75 % more subkey bytes are retrieved from DMR and inverse function based FD methods, respectively compared with no FD case. In case of parity based FD, the number of retrieved subkey are 25%

less than the No FD case. In Fig. 5.7, we can see that DMR and inverse FD are able to retrieve the



(a) (b)

Fig. 5. 5 Partial Guessing Entropy for the AES S-Box (a) without fault detection, and (b) protection with a parity check code.

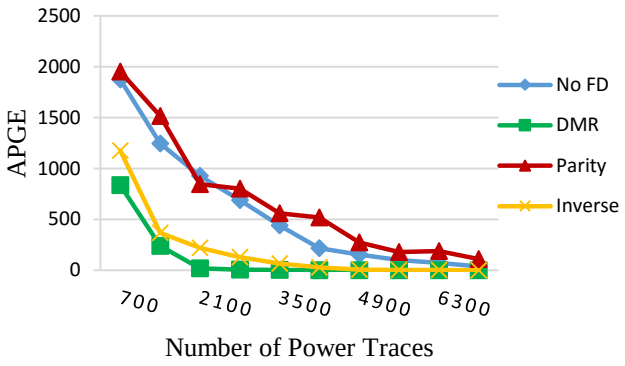


Fig. 5. 6 APGE for different FD methods applied to S-Box

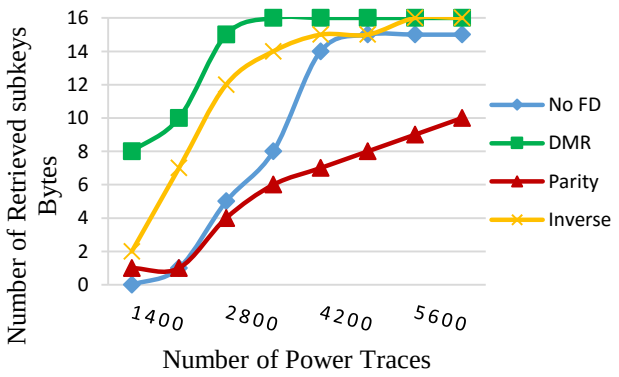


Fig. 5. 7 The number of subkey bytes found over different number of power traces in S-Box.

key faster than no FD case as these operations increase the power consumption and related correlation. DMR is the fastest method and the parity is the slowest one. The total subkey bytes retrieved with parity FD at the 6000 power traces are just 10 compared with other FD methods.

5.4.2.3. Impact of MixColumns with FT on Key Retrieval Speed

We repeated our experiments on the MixColumns transformation as well. No FD, DMR, parity, and inverse functions are applied to the MixColumns, respectively. The prediction of parity bits for the MixColumns is based on the work [8]. To obtain the inverse function of the MixColumns, we converted the System Verilog version of inverse function for MixColumn published on the opencore website [79]. As shown in Fig. 5.8, the APGE for all FD cases for the MixColumns is almost same after 5000 power traces except parity based FD mechanism. Similarly, we checked the number of retrieved subkey bytes over different numbers of power traces. As shown in Fig. 5.9, the key retrieval speed for the parity based FD is slowest among other FD, i.e. DMR and inverse. This conclusion for parity based FD is consistent for MixColumns compared to S-Box transformation. Even though in 6000 power traces, number of subkey bytes in DMR and inverse FD is more than parity FD, the key retrieving process is slower compared with application of same FD schemes in S-Box. The reason for this can be the optimization of design area by Xilinx synthesis tool. Although the MixColumns module uses much less FPGA slices than S-Box, the speed of key retrieving in the MixColumn with different FD methods is as significant as that in the S-Box.

5.4.3. Impact of Power Models in CPA on the Efficiency of Key Retrieving

In Section 5.2.3, we have differentiated the Hamming weight and Hamming distance power models in CPA. Regazzoni et al. [20] performed the CPA using the power traces obtained from the transistor-level simulation on the S-Box protected with parity check codes. Their CPA was based

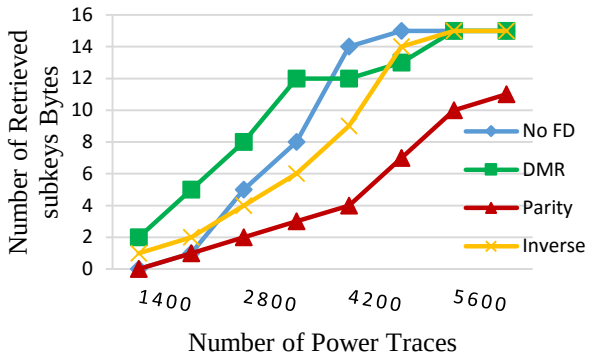


Fig. 5. 8 The number of subkey bytes found over different power traces in MixColumns.

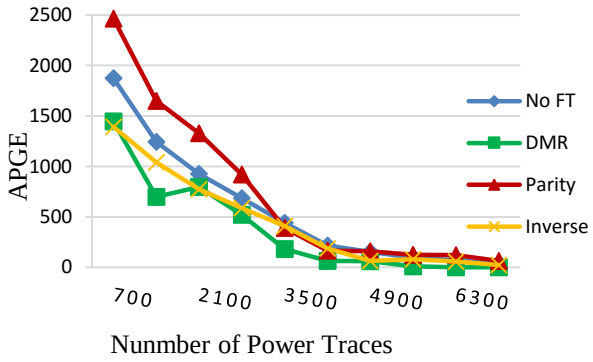


Fig. 5. 9 Accumulated PGE for different FD methods applied to the MixColumns.

on the Hamming weight power model. The result presented in [20] showed that the key retrieval process is faster when the FD is presented in the AES implementation, comparing with the no FD case. In this section, we repeated the same experiment as we did for Fig. 5.6, except using the Hamming weight power model in the CPA instead of using Hamming distance one.

To have a reasonable comparison, we used the same set of power traces that were used in the experiment for Fig. 5.6. As shown in Fig. 5.10, the parity based FD method indeed results in a significant decrease in APGE comparing with the no FD case. This observation matches to the simulation result in the early work [20]. If Hamming weight power model is adopted in the CPA, adding a FD method to an S-Box indeed increases the vulnerability of the AES against CPA except

the parity check code based.

We take the parity method used in S-Box as an example to compare the APGE trend for two CPA attacks with different power models: Hamming distance (HD: AES last round-state) and Hamming weight (HW: AES S-Box output, first round (Enc)). As shown in Fig. 5.11, the CPA using the Hamming weight power model requires more power traces to recover the subkey bytes than the CPA using the Hamming distance model.

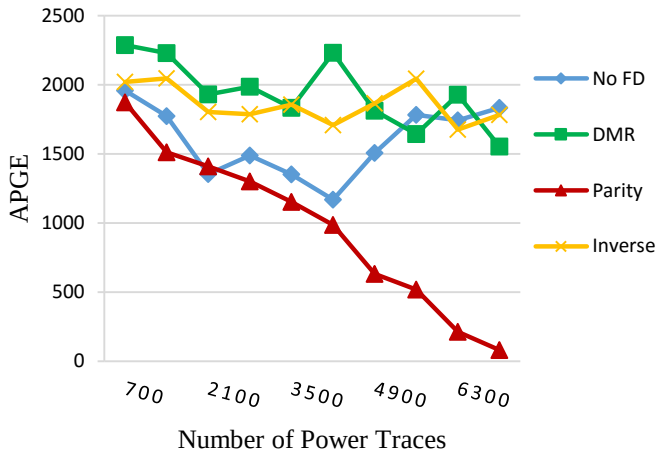


Fig. 5. 10 Accumulated PGE for different FD methods applied to the S-Box with HW power model.

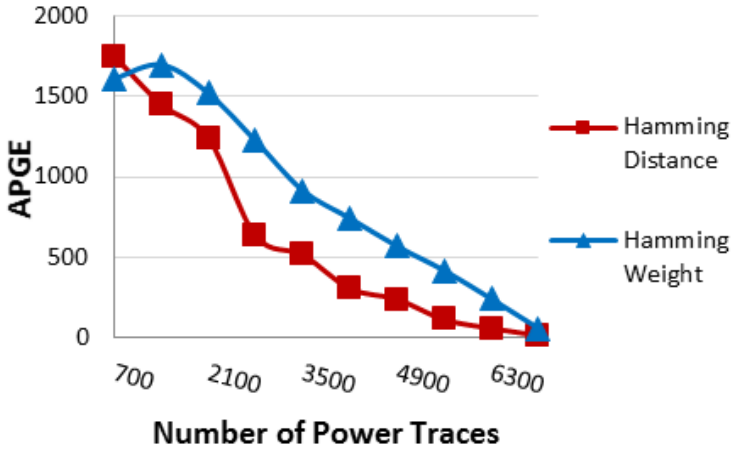


Fig. 5. 11 APGE obtained from different power.

5.4.4. Impact of Multiple FT Methods on the Efficiency of Key Retrieving

The experimental results in the previous section reveal that the FD method on different computation steps has different impact on the key retrieval speed through CPA. In this part we studied the impact of different number of FDs in the AES implementation on the key retrieval speed. As the number of key retrieval analysis increases, the achieved APGE is more consistent.

Fig. 5.12 compares the impact of the number of FD schems on the key retrieval speed. In this experiment, we considered FD as parity check code. We done first experiment with FD only in S-Box, then we added FD in MixColumns and finally in AddRoundKey as well. According to Fig. 5.12, the APGE for all the three cases are dropping as the number of power traces increase. We can conclude that more adding redundancy to all transformation does not help for protecting the AES implementation from the CPA attack. Hence the factor of applying redundancy to all module should be considered while designing the cipher implementation in FPGA platform

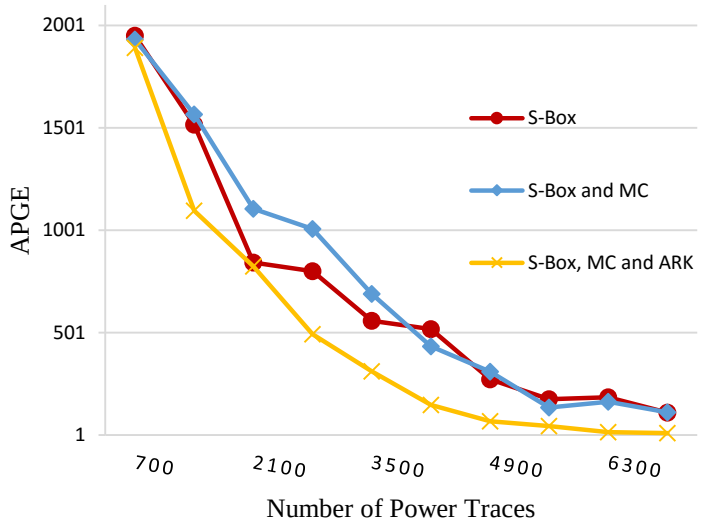


Fig. 5. 12 Impact of multiple FD methods on key retrieval speed.

5.5. Conclusions

Side channel analysis (SCA) and fault analysis (FA) attacks are widely studied for cryptosystems as these are most prominent attacks which can reveal the secret key by the attacker. The cryptographic systems are designed to protect from such attacks, however the countermeasure for The majority of existing efforts suggest using two separate countermeasures to address these two attacks. Unfortunately, one countermeasure for a particular attack can influence the other attack positively or negatively. In this work, we perform systematic assessment on the impact of different fault detection (FD) methods on the CPA resistance of complete AES implementation. Our hardware based experimental results show that the fault detection methods affects the speed of key retrieving through CPA attack. The FD methods can have negative or positive effect on the key retrieval process, for instance - DMR based FD in S-Box ease the process of key retrieval while parity check code make it difficult. We also analyzed the impact of adding more FD methods on the different transformation of AES implementation, however more redundancy do not help to prevent the CPA attack.

Chapter 6. A Systematic FPGA-based Assessment on Fault-Resistant AES against Correlation Power Analysis Attack

6.1. Introduction

Cryptographic algorithms are widely used for the security-critical systems in military, government, and banking applications. Theoretically, the extremely high computation complexity of ciphers prevents the key recovery through brute-force attempts. However, it has been demonstrated that physical attacks [73-46, 14, 15, 19, 39, 51, 55, 56], such as fault analysis and side-channel analysis attacks, can significantly reduce the number of brute-force attempts needed for the secret key retrieval.

Fault analysis (FA) attacks [73-76] and side channel analysis (SCA) attacks [15, 58, 28, 22] are two typical physical attacks to reveal the cipher key. FA can be done using deliberate injection of faults into cryptographic devices by means of white light, laser beam, voltage/clock glitch, and temperature control [13]. The work [56] shows that a FA attack can break the advanced encryption standard (AES) implementation with only pair of fault-free and faulty ciphertexts. The work [55] demonstrates that the FA attack can retrieve the key with the knowledge of faulty ciphertexts only. To thwart FA attacks, hardware-redundancy based [16, 70], information-redundancy based [78, 80, 81, 82, 87], and time-redundancy based [84] fault detection schemes are commonly used. The other class of physical attack is SCA attack [14, 15, 58, 28, 22], which exploits the correlation between the power consumption and the underlying logic switching of the cryptosystem to determine the secret key. Simple power analysis (SPA), differential power analysis (DPA) [15], and correlation power analysis (CPA) [14] are three variants of SCA. CPA has been identified as the most efficient one as it requires the least power traces to find the crypto key.

Countermeasures to thwart FA and SCA attacks are typically developed in a separate fashion. Although, the countermeasures for individual attacks have had a great progress, a new powerful attack is emerging—a combination of FA and SCA attack, i.e. combined attack [55, 51, 39, 19]. The working principle of a combined attack is to inject faults (e.g. stuck-at faults) to a specific location in the cryptosystem, so that a large portion of the cryptosystem does not switch and thus does not consume power. Consequently, the attacker can have more accurate power estimation for the AES using the subkey they guessed. If the estimated power is closer to the real power, the CPA key retrieval will be faster.

Because our method prevents successful fault attacks and meanwhile uses dynamic masking to raise the bar for accurate power prediction, proposed method is superior to a single FA or SCA countermeasure. This work fills in this need by conducting a FPGA-based comprehensive assessment. Furthermore, we propose a corresponding countermeasure for the combined attack. The rest of this chapter is organized as follow. In Section 6.2, we discuss the related work and highlight our contributions. We introduce a brief overview of AES structure and correlation power analysis (CPA) in Section 6.3. Section 6.4 provides the experimental setup, analysis of single power trace, metrics used for CPA attack, and CPA analysis using different power models. We present the FPGA-based comprehensive assessment results in Section 6.5. In Section 6.6, we propose a potential countermeasure to thwart the combined attack. We conclude this work in Section 6.7

6.2. Related Work and Our Contributions

6.2.1. Related work

Traditional fault detection (FD) methods for cryptosystems exploit information redundancy, spatial redundancy, or time redundancy to detect faults. Double modular redundancy (DMR) [81],

is the simplest fault detection method based on hardware redundancy scheme. The duplicated copy can be the same module transformation or its inverse version [82]. Error control code (ECC) based FD scheme have been explored widely for AES cipher [80, 82, 87, 78, 81]. The ECC parity bits of the output of every round are predicted from the inputs of the corresponding transformation [80, 82, 87, 81]. The process of ECC encoding can be implemented using look-up table [80, 78, 81] and logic computation [87]. The parity schemes presented in [82] are independent of the implementation methods of the S-Boxes and the inverse S-Boxes. A systematic concurrent error detection approach [63] is presented for symmetric block ciphers at the register transfer level. This method exploits the inverse relationship between the encryption and decryption at the individual operation level, round level, and algorithm level.

The main objective of a SCA countermeasure is to make the power consumption of a device as independent as possible to the intermediate values of a cryptographic algorithm. Random masking based approaches [63, 22, 97] are extensively studied in this regard. The formal security of masking countermeasure is provided in [63]. In [22], the authors propose to mask the intermediate value of an AES SubBytes implementation, and they basically shift the computation of the finite field inversion in the AES S-box down to GF(4). The scheme in [22] is able to resist zero-value attacks. High order masking based countermeasure against DPA is presented in [97]. The other techniques like insertion of dummy code, power consumption randomization, and balancing of data are used for processor level architectures to thwart power based side channel attacks [64]. Aggressive frequency scaling [98] is proposed to defeat DPA/CPA by reducing the probability of maximum correlation values for correct key.

The countermeasures for FA and SCA attacks are typically implemented separately. However, the additional hardware cost and power consumption induced by the fault detection circuitry to

thwart FA could affect the success speed of CPA. The work [21] shows that the power analysis vulnerability depends on the particular error detection code used for fault detection. Another work [60] studies the impact of fault detection codes on the correlation between the dynamic power consumption and the manipulated data on the simple 8-bit and 16-bit registers, which is much less complicated than an AES implementation. The existing work [20, 66] concludes that the use of parity codes in the S-Box of AES for fault detection makes CPA succeed faster than the S-Box without any fault detection mechanisms. Unfortunately, that observation is based on the gate-level simulation on the S-Box only, rather than a real hardware emulation of the complete AES. Moreover, the power model used in the early work [20, 66] is Hamming weight, not the powerful Hamming distance. Hamming distance model is powerful because it can retrieve more number of subkeys in given number of power traces than Hamming weight model.

The FPGA-based assessments in [58] and [57] indicate that the fault detection methods can have negative or positive effect on the key retrieval speed of the CPA attack. Although the works [58, 57] provide the guidelines to choose a particular fault detection scheme depending on their CPA analysis and fault coverage results, they do not consider the other factors (other than the fault detection methods) that could affect the CPA success rate. There is lack of thorough investigation on other factors that can influence the success rate of CPA attack.

Towards the combined attacks, Clavier et.al [55] suggests using algorithm-level inverse computation, duplicated rounds, purposely introducing data error, and checksums. They further predicted that randomization (time or order) during the execution will be helpful to destabilized the cryptanalysis in the SCA attack. Roche et.al [39] point out that the insertion of random delay may be a possible countermeasure against the combined SCA and FA attacks. Alternatively, they suggest duplicating the computation paths and masking the ciphertexts from two paths with two

masking vectors. Although the works [55, 39] focus on the combined attacks from an attacker's point of view and provide some advices for possible countermeasures, no quantitative experimental results have been provided to prove the feasibility of their suggestions. Our work follows their direction and we perform FPGA-based experiment to demonstrate a feasible countermeasure for a combined CPA and FA attacks.

6.2.2. Our Contributions

This work is the extended version of our early work [57]. Compared to [57], this version provides more comprehensive assessment on the impact of FA countermeasures on the efficiency of CPA attack in the Advanced Encryption Standard (AES). More precisely,

- We systematically evaluated the impact of typical fault detection methods on the key retrieval speed of the CPA attack. Different than other works [20, 66, 58, 57], we categorized the comparison into (1) what kind of redundancy is used for fault detection, (2) how much redundancy is introduced to the crypto implementation, (3) which location is the fault detection mechanism applied, and (4) which power model is used in the CPA attack. Accumulated Partial Guessing Entropy (APGE) and the number of subkey bytes retrieved for a given number of power traces are quantitatively compared among different countermeasures.
- We further provided new assessment on the impact of the S-Box implementation method and FPGA synthesis option on the CPA key retrieval speed.
- To explore the efficient countermeasure for the combined attack, we first examined the feasibility by means of heterogeneous-redundancy fault detection methods. Next, we proposed a method that combines dynamic masking and error deflection techniques to thwart FA and CPA attacks simultaneously. Different than the suggestions provided by [39], the masking vector in our method is changeable over time, instead of a static vector.

6.3. Our Systematic Assessment

6.5.1. Impact of Hardware Redundancy based Fault Detection Mechanisms on CPA Key Retrieval Speed

To assess the impact of hardware redundancy on CPA attack, we applied double modular redundancy (named as DMR) and inverse encryption function (named as inverse) immediately after the module operation to detect faults in two AES modules: SubBytes (S-Box) and Mixcolumns. The results are shown in Fig. 6.1. We observed that hardware redundancy based fault detection methods positively affect the key retrieval speed of the CPA attack. This conclusion is consistent no matter where we apply the hardware-redundancy based fault detection methods. In Figs. 6.1(a) and (b), the APGE of the AES with fault detection methods is less than that of the AES without fault detection capability.

While, the impact of the hardware redundancy based fault detection methods on the key retrieval speed is different. As shown in Fig.6.1 (a), the APGE goes to zero at the power traces of 2800 when DMR is applied to the S-Box. The use of the inverse fault detection in the S-Box results in the need of 2200 more power traces for key retrieving than DMR. Similarly, the application of DMR in the MixColumns module makes it easier to recover the key than the inverse method, as shown in Fig. 6.1(b). We further observed that the cause of non-zero PGE is sometimes originated from the same subkey byte. Hence we used another way to assess the speed of key retrieval, we analyzed the number of subkey bytes that were recovered at each power trace interval, which is shown in Fig. 6.2. We can see that the DMR is the most vulnerable method for CPA attack.

From this set of experiments we can see, more hardware redundancy leads to a less number of power traces needed for key retrieval. The amount of hardware redundancy introduced to the

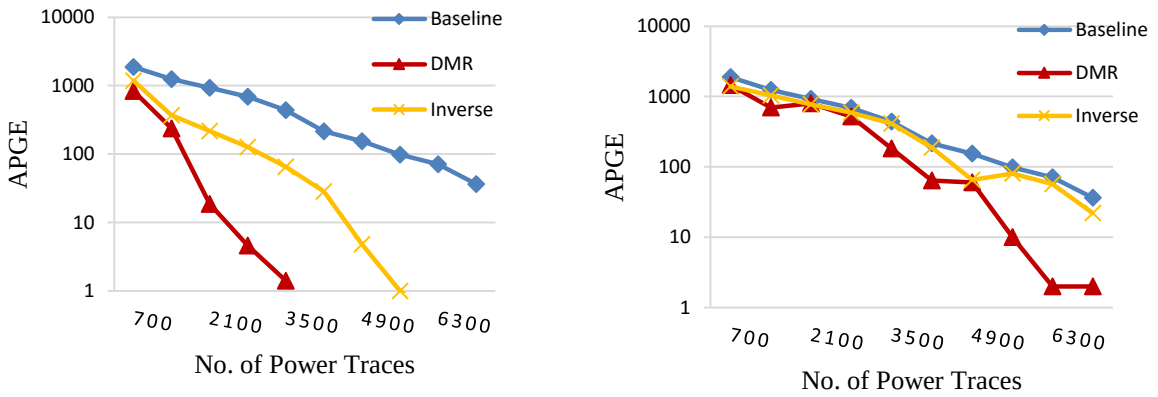


Fig. 6. 1 Average APGE for hardware-redundancy based FD methods applied to (a) S-Box, and (b) MixColumns.

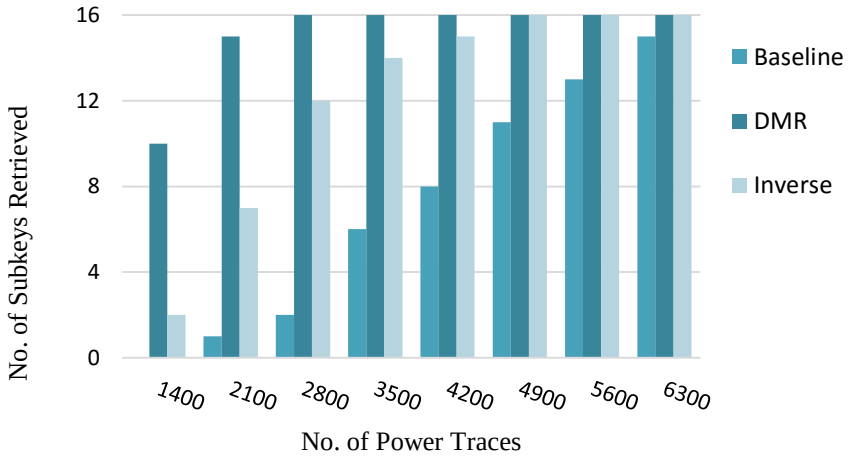


Fig. 6. 2 The number of subkeys retrieved over the number of power traces used in CPA.

overall AES not only depends on what fault detection method, but also depends on where the fault detection method applies.

The hardware cost of the implementation is presented in Table 6.1. Hardware cost and power consumption of S-Box inverse is more than S-Box DMR, but the number of power traces required for CPA in DMR is less than that of the inverse method. This is useful observation as the DMR is

just replication of existing module while the inverse function introduces some non-linearity to the computation. In case of MixColumns, due to the optimization options in Xilinx ISE tool, the hardware cost of DMR and inverse FD didn't increase significantly. However, the key retrieval process is fast in the case of DMR and inverse FD because of compact placement of cells in FPGA.

Table 6. 1 Hardware cost of hardware-redundancy based FDs.

Design	No. of Slice Registers	No. of Slice LUTs	No. of LUT Flip-flops
S-Box DMR	759	3414	582
S-Box Inverse	761	3399	570
MixColumns DMR	769	2675	831
MixColumns Inverse	766	2852	882

6.5.2. Impact of Information-Redundancy based Fault Detection Mechanisms on CPA Key Retrieval Speed

6.5.2.1. PGE Comparison of Fault Detection Methods Applied to S-Box and MixColumns

Figures 6.3 and 6.4 show the PGE for each subbyte key in the process of key retrieving when a parity check code is applied to the S-Box and MixColumns. As each byte has 8 bits, the total number of key guessing attempts for each subkey byte is up to 256. As shown in Figs. 6.3 and 6.4, eventually the PGE decreases as the number of power traces used in the CPA attacks increases.

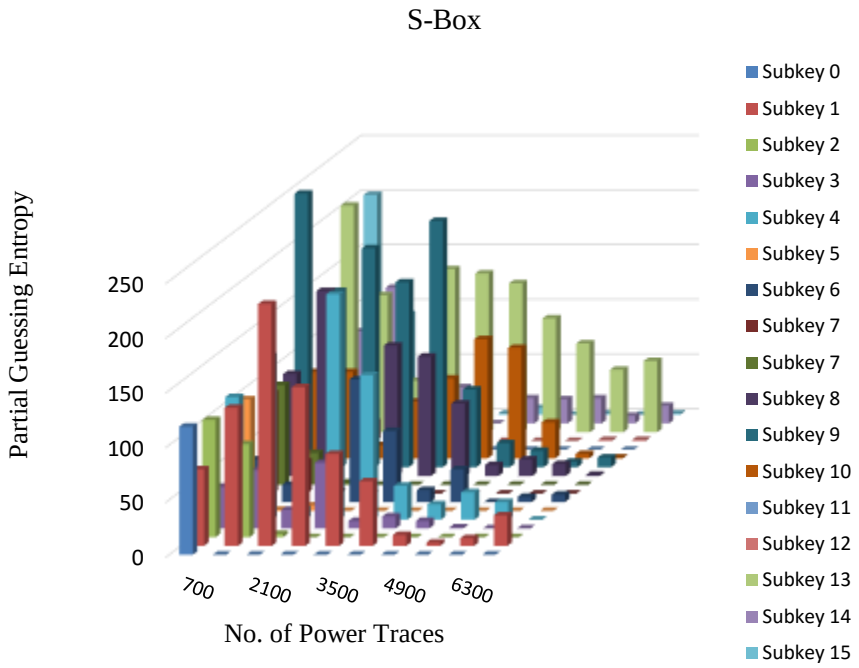


Fig. 6. 3 Partial guessing entropy for the S-Box protected with parity-check code.

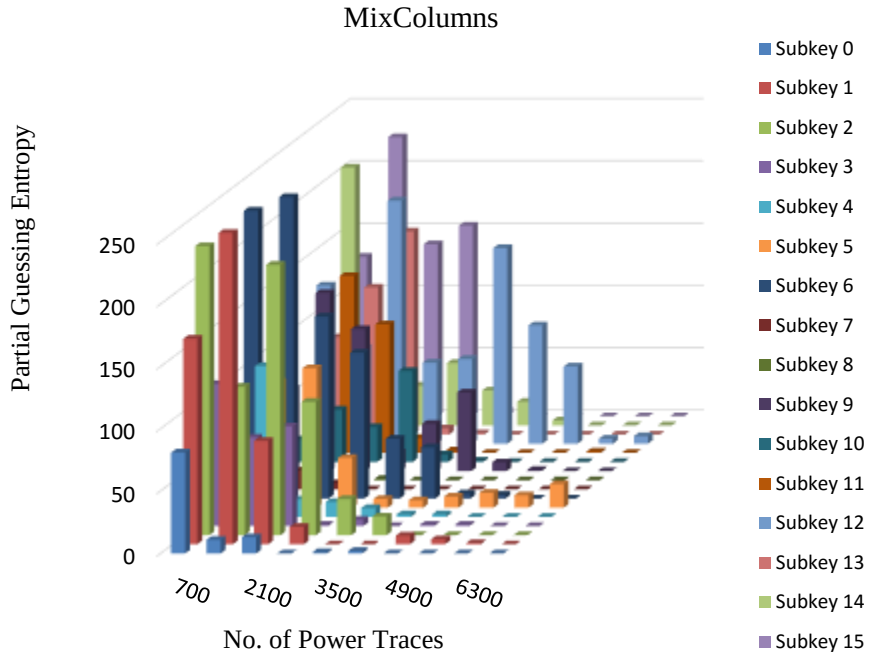


Fig. 6. 4 Partial guessing entropy for the MixColumns protected with parity-check code.

If the limited power traces are available with the adversary, the CPA attack may not be successful. If the S-Box is protected with a parity FD, the attacker needs a larger number of power traces to retrieve the complete key than that of MixColumns protected with same FD method.

6.5.2.2. Impact of the Location of Fault Detection Mechanisms on CPA Key Retrieval Speed

The conclusion for hardware redundancy is straightforward—more hardware makes CPA attack easier. In contrast, the impact of information redundancy on the efficiency of CPA attacks is relatively complicated. Initially, we examine the impact of where the parity check code is used and on the key retrieval speed. Next, we access how the complexity of the parity check code affects the key recovery. First we apply an even parity check code to S-Box and MixColumns module individually. As shown in Fig. 6.5, the key retrieval speed depends upon the module where parity-check code is used. For instance, it is more difficult to find out the key if the parity-check code is used in the S-Box than the same code is applied to the MixColumns. One reason for this observation is the non-linearity of S-Box. The MixColumns operation is linear in nature, so the parity-check codes make it easier to perform the CPA attack.

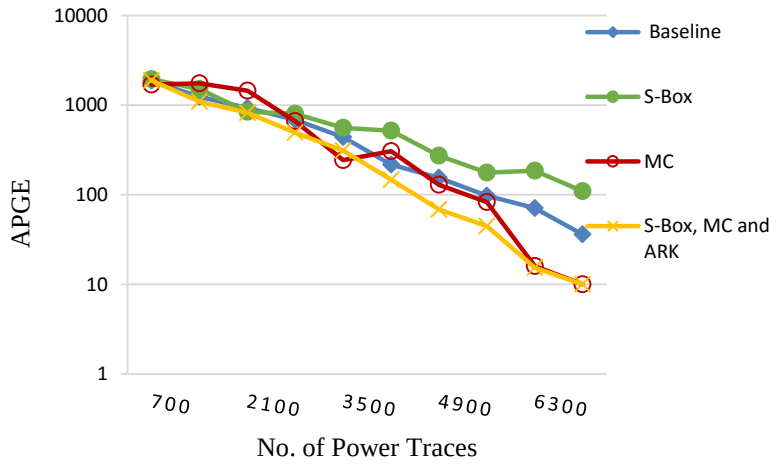


Fig. 6. 5 Average APGE for even parity-code applied at different modules. MC: MicColumns, ARK: Add Round Key.

Moreover, we did experiments by applying parity check code to all modules in the AES. Interestingly, we found that more information redundancy do not further help to prevent the CPA attacks. As shown in Fig. 6.5, the protection on all operation modules in the AES allows the CPA attack to obtain the key with less power traces than the protection on a single module, such as the S-Box and MixColumns. It indicates that, the implementation of information redundant methods unavoidably results in the hardware cost increasing. Therefore, we should consider the non-linear property and hardware overhead simultaneously while assessing the impact of a FD method on the CPA attack efficiency.

6.5.2.3. Impact of the Degree of Information-Redundancy on CPA Key Retrieval Speed

We conducted experiments to evaluate how the degree of information redundancy affects the key retrieval speed. Three CRC codes— CRC2, CRC4, and CRC8—are utilized in this set of experiments. The digits after CRC stands for the number of redundant bits introduced in each message byte. The experimental results are shown in Fig. 6.6. Surprisingly, the more the redundant bits, the easier key retrieval process is. More information redundancy brings in higher correlation between power traces and a suspected key vector. When we compare the hardware cost of each method, we observe that the hardware cost is the primary factor that is responsible for fast key recovery.

As shown in Table 6.2 CRC8 consumes the most area in FPGA, compared to CRC2 and CRC4. Although CRC8 uses less number of slices than CRC4, CRC8 consumes more LUTs than CRC4. This is because the optimization process in the Xilinx synthesize tool utilized the slices in the FPGA more effectively in CRC8 than in CRC4.

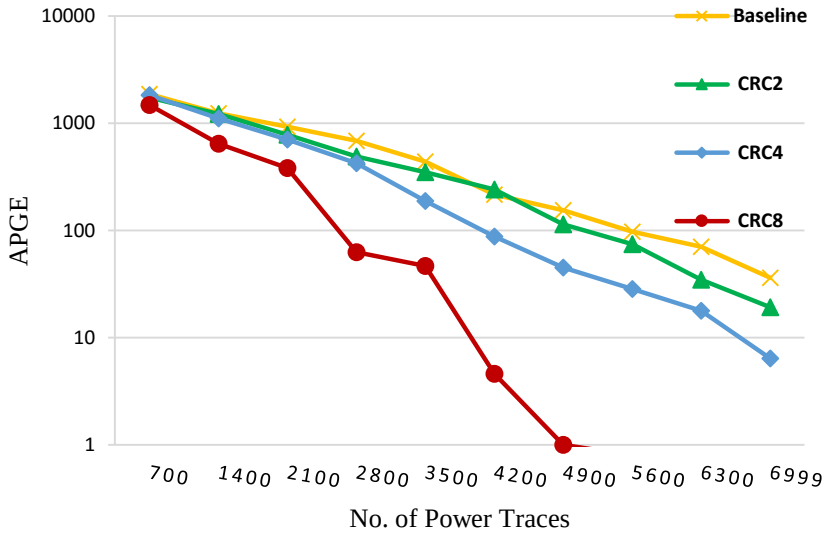


Fig. 6. 6 Average APGE for CRC code applied to S-Box.

Table 6. 2 Hardware cost of CRC based fault detection methods

Design	Number of Slice Registers	Number of Slice LUTs	Number of occupied Slices
Baseline	769	2663	828
CRC2	768	3012	941
CRC4	768	3144	984
CRC8	768	3390	947

6.5.3. Impact of S-Box Implementation Methods on CPA Key Retrieval Speed

S-Box is the most critical operation in AES and approximately consumes one third of total area. It can be implemented in different styles, a look up table (LUT) and the Galois Field (GF(2)) based combinational logic[64]. In this subsection, we compare the impact of different S-Box

implementation methods on the key retrieval speed. We performed CPA attacks on the S-Box implemented with (1) LUT only, (2) GF logic only, (3) LUT and GF with alternative ratio of 1/2:1/2 (i.e.1GF 1LUT) (4) LUT and GF with alternative ratio of 1/4:3/4 (i.e.1LUT 3GF), and(4) LUT and GF with alternative ratio of 3/4:1/4 (i.e. 3LUT 1GF). The impact of different configurations in the implementation of S-Box module on APGE is shown in Fig. 6.7. The corresponding FPGA cost is presented in Table 6.7. As can be seen, the hardware cost of the design directly influences the key retrieval speed of the CPA attacks. The use of more hardware positively affects the CPA attack success speed.

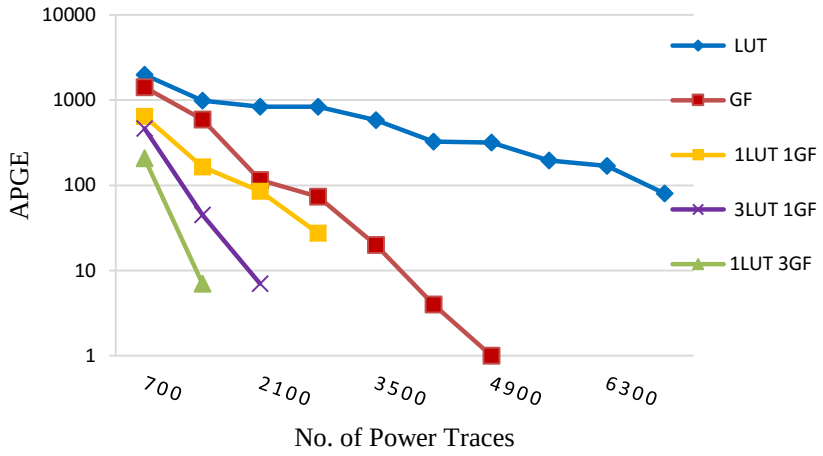


Fig. 6. 7 Average APGE for different implementation of S-Box.

Table 6. 3 Hardware cost of s-box implementation in FPGA

Design	Number of Slice Registers	Number of Slice LUTs	Number of occupied Slices
LUT	769	2663	828
GF	916	3549	1160
1LUT 1GF	876	4077	1310
3LUT 1GF	882	4123	1364
1LUT 3GF	882	4125	1385

6.5.4. Impact of Synthesis Tool Optimization Strategy on CPA Key Retrieval Speed

The optimization strategy offered by the FPGA synthesis tool plays an important role in the key retrieval speed of CPA. We carried the set of experiments to utilize all the synthesis optimization option to see the impact on CPA attack. Figure 6.8 shows the APGE for the protected AES that is synthesized with different FPGA synthesize efforts— normal, high and fast—available in the Xilinx IDE tool.

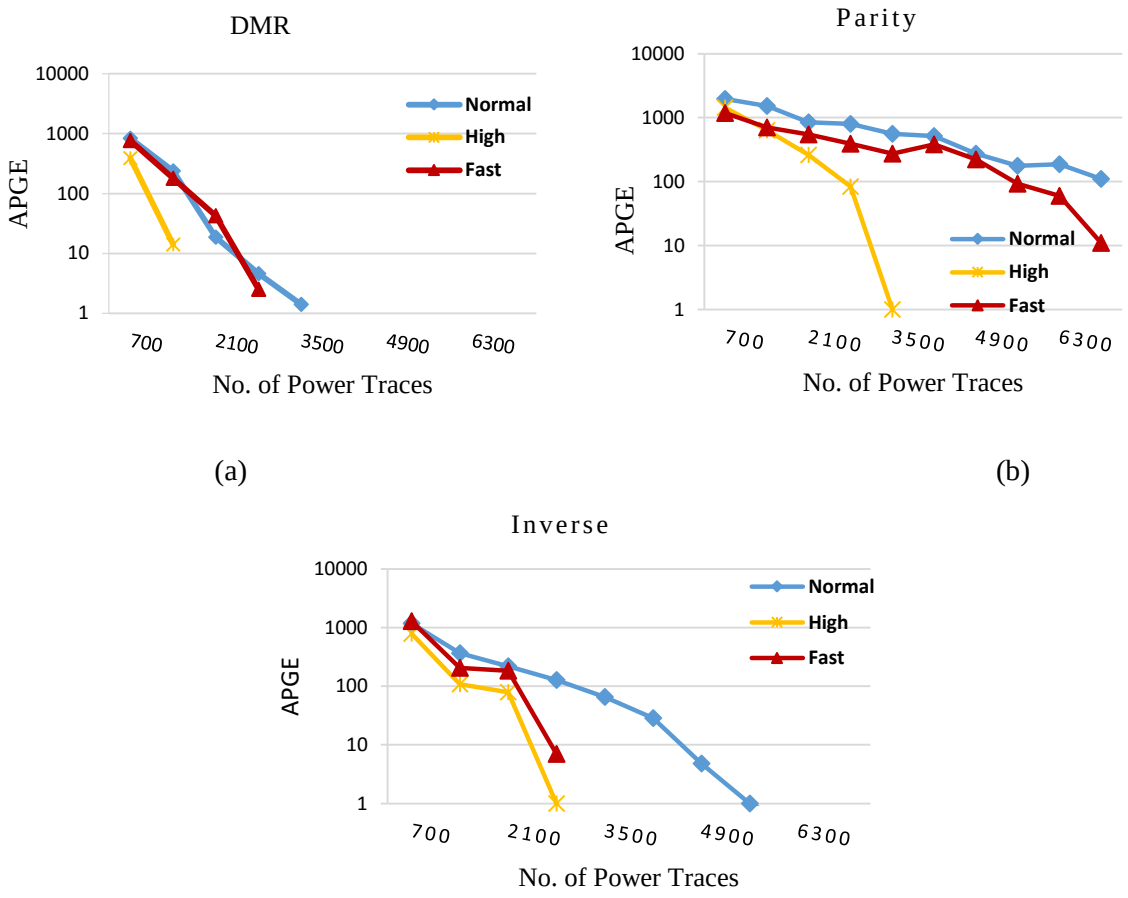


Fig. 6. 8 Impact of synthesis optimization efforts by ISE tool on average APGE for (a) DMR (b) Parity, and (c) Inverse.

For instance, in Fig. 6.9 (a), the number of power traces needed for the key retrieval of the AES S-Box with DMR is 2800, 1400, and 2100 for the normal, high, and fast synthesize options, respectively.

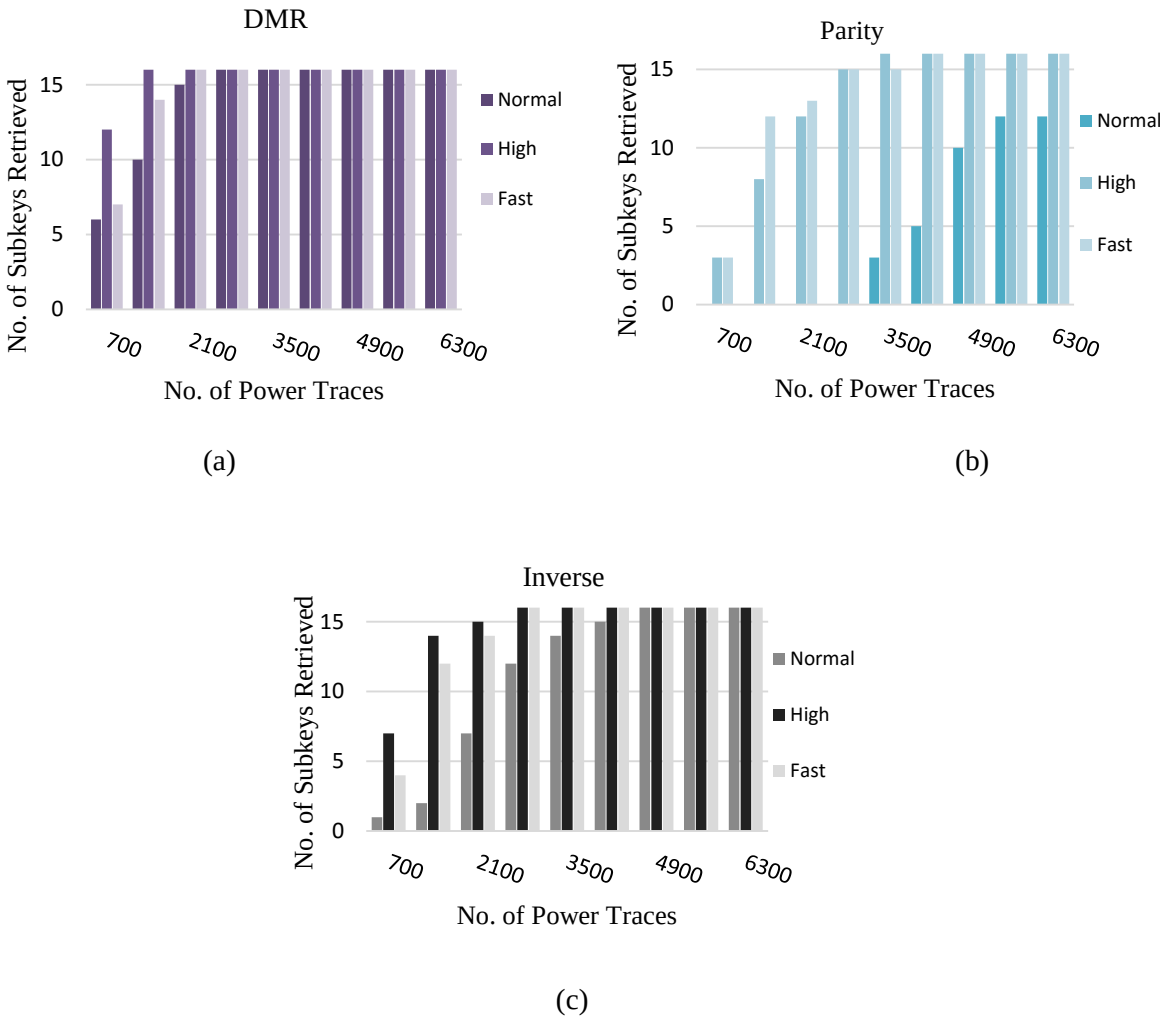


Fig. 6. 9 The number of subkeys retrieved versus the number of power traces used for (a) DMR, (b) parity-check code, and (c) inverse in the S-Box of AES.

As shown in Fig. 6.8(b) and 6.9(b), APGE and key retrieval speed for the case of the S-Box with the even parity check code, the high synthesize optimization can render the complete key with 3500 power traces; while the normal optimization cannot help to retrieve the complete key within 7000 power traces as shown in Fig. 6.5. The impact of synthesize optimization options on

APGE and the key retrieval speed are shown in Figs. 6.8 and Fig. 6.9. As we can see, different fault detection mechanisms for S-Box yield the same effect on CPA. If a high optimization effort option is used during the design synthesis, the key retrieval process will be faster compared with the other two optimization choices (i.e. normal and fast). Therefore, we suggest that the factor of optimization efforts should be considered if we use FPGA to implement the secure and reliable encryption engine for sensitive applications.

6.5.5. Impact of Heterogeneous-Redundancy based Fault Detection Mechanisms on CPA Efficiency

We refer the heterogeneous redundancy to application of different type of FD methods to the specific submodules in the AES. To balance the CPA dependent factors such as hardware cost and non-linearity induced by fault detection methods, we examine whether the heterogeneous redundancy effort can efficiently thwart the CPA attack. We implemented DMR in the MixColumns and varied different fault detection methods in the S-Box. The APGE for different combinations of fault detection schemes is shown in Fig. 11. Recall the experimental result shown in Fig. 6.2, the CPA attack uses 2800 power traces to retrieve the key for the S-Box of the AES protected with DMR. However, if both the S-Box and MixColumns modules are protected with DMR, the number of traces required for key retrieval indeed increases. This may be because the contribution from the hardware redundancy introduced in different operation modules is cancelled out at some degree. The combination of S-Box with parity check code & MixColumns with DMR requires more power traces compared with the case of the S-Box protected with parity FD which cannot retrieve the key in 7000 power traces as shown in Fig.6.10. The arrangement of S-Box with parity & MixColumns with the DMR method constitutes the least FPGA area, i.e. 819 FPGA slices. In contrast, the S-Box with DMR consumes 17% (960) and S-Box with inverse consumes

77% (1451) more FPGA slices, the detail hardware cost is shown in Table 6.4. The number of power traces required to retrieve the key in case of S-Box inverse & MixColumns DMR is 4900 however in case of S-Box DMR & MixColumns DMR is 6300. Hardware cost of combination of S-Box inverse & MixColumns DMR is more expensive than combination of S-Box DMR & MixColumns DMR and S-Box parity & MixColumns DMR which make key retrieval faster. Based on this set of experiments, we can conclude that different types of redundancies and different modules under protection affect the overall success speed of the CPA attack.

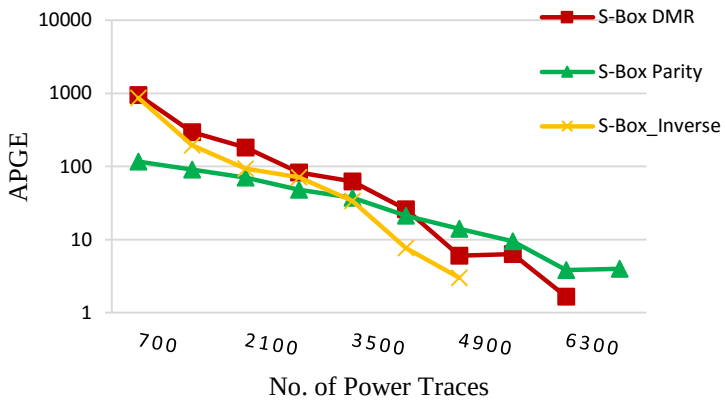


Fig. 6. 10 Heterogeneous redundancy applied to the S-Box and MixColumn modules.

Table 6. 4 Hardware cost mixcolumns DMR FD with different FDs for S-Box

Design	Number of Slice Registers	Number of Slice LUTs	Number of occupied Slices
S-Box DMR	3414	759	960
S-Box Parity	2648	760	819
S-Box Inverse	3784	761	1451

6.4. Proposed Countermeasure against the Combination of CPA and FA Attacks

6.6.1. Proposed Method Description

As single fault detection method cannot completely thwart the combination of CPA and fault attacks, we propose to integrate a dynamic masking technique with an error detection code (ECC) based error deflection mechanism to thwart the combined attacks. The main principle of our method is highlighted in Fig. 6.11.

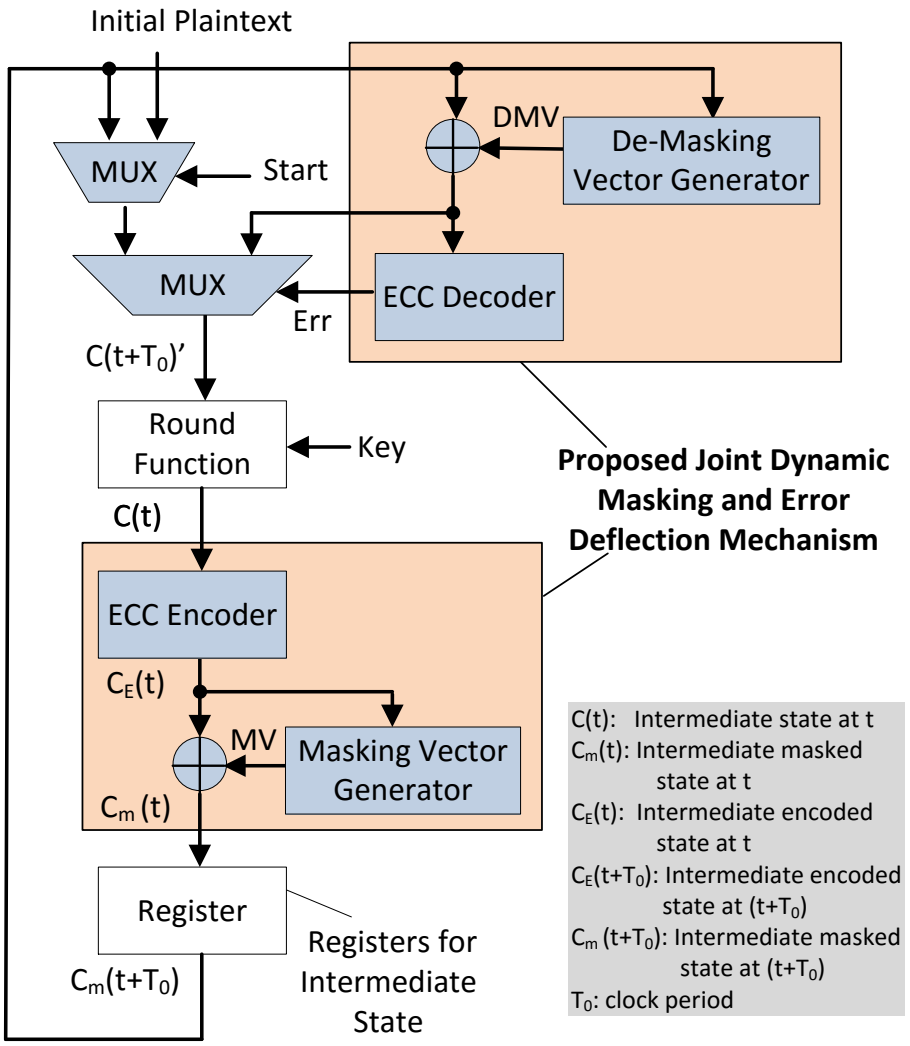


Fig. 6. 11 Concept of proposed joint dynamic masking and error deflection method.

Unlike the fault detection in the previous work encoding the S-Box input [81], we encode the intermediate state value before it reaches the state registers. We argue that it makes more sense to manipulate a register value than modify the S-Box logic to achieve a better correlation of the estimated power consumption and the guessed subkey for the CPA attack.

Another innovation of our method is the dynamic masking. Instead of using a fixed masking vector, we propose to generate the masking vector at runtime using the (de)masking vector generator. The masking vector is a modification of the intermediate state value. As the intermediate state register changes the value over the time, the (de)masking vector is not consistent. Consequently, the power model modification according to a guessed masking vector will fail.

The detailed method is depicted in Fig. 6.11. Before reaching the intermediate state register, the new output from the round function C(t) is first encoded with an error control coding (ECC) encoder – for instance, an 8-bit CRC encoder. Then, the encoded output, CE (t), is masked by a masking vector, MV, provided by the masking vector generator. In our method, the intermediate state register saves the encoded masked intermediate state, rather than the original output from the round function. If the CPA attacker attempts to calculate the correlation between the predicted key and the power measurement, the true correlation will be skewed by ECC and masking process proposed in our method. More importantly, the proposed method can efficiently detect the faults injected in the intermediate register, where the fault attack is typically performed. Our method will first de-mask the encoded and masked intermediate state CE(t+T0). Due to the masking process, the parity check bits recomputed by the ECC decoder may not match to the check bits carried by CE(t+T0). The mismatch on the parity check bits indicates the detection of fault attack. Different than typical fault detection methods, our method continues to feed the round function with a non-

zero input, which is a portion of the masked vector Cm(t+T0). The use of the masked vector deflects the normal power prediction, thus misleading the correlation power analysis.

The algorithm for the dynamic masking vector generation is expressed in eqs. (6.2)-(6.5). Assume xi is the element for the output of the round function. We propose to shift the intermediate state vector CE(t) to dynamically generate the mask vector MV. In the eq. (6.3), we use left shifting by 2, but any number can be applied too. The masked intermediate state vector Cm(t) is expressed in eq. (6.4). The de-masking process is performed in eq. (6.5).

$$C_E(t) = \{x_n, x_{n-1}, x_{n-2}, \cdots, x_i, \cdots, x_1, x_0\} \tag{6.2}$$

$$MV = \{x_{n-2}, x_{n-3} \cdots, x_{i-2}, \cdots x_1, x_0, 0, 0\} \tag{6.3}$$

$$C_m(t) = \{x_n \oplus x_{n-2}, x_{n-1} \oplus x_{n-3}, \cdots, x_1 \oplus 0, x_0 \oplus 0\} \tag{6.4}$$

$$C_m(t+T_0)'[i] = \begin{cases} C_m(t+T_0)'[i], & (i = 0,1) \\ C_m(t+T_0)'[i] \oplus C_m(t+T_0)'[i-2], & (i \geq 2) \end{cases} \tag{6.5}$$

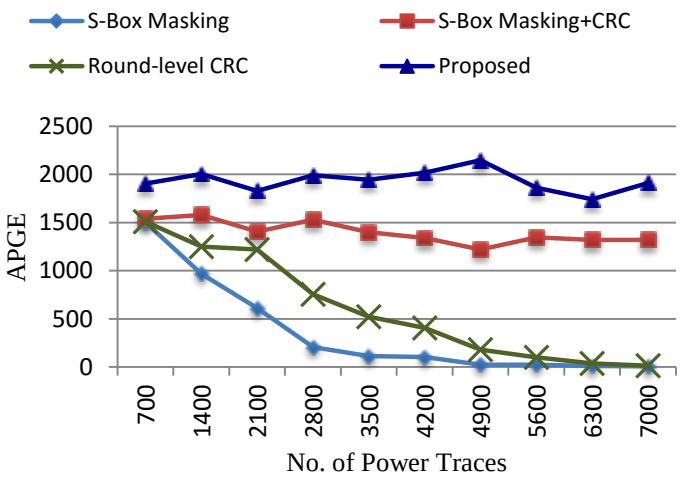
The proposed method also allows the round function including other fault detection methods inside itself, thus thwarting the fault attack aiming for the inner round function.

6.6.2. Evaluation of the Resistance to CPA Attacks

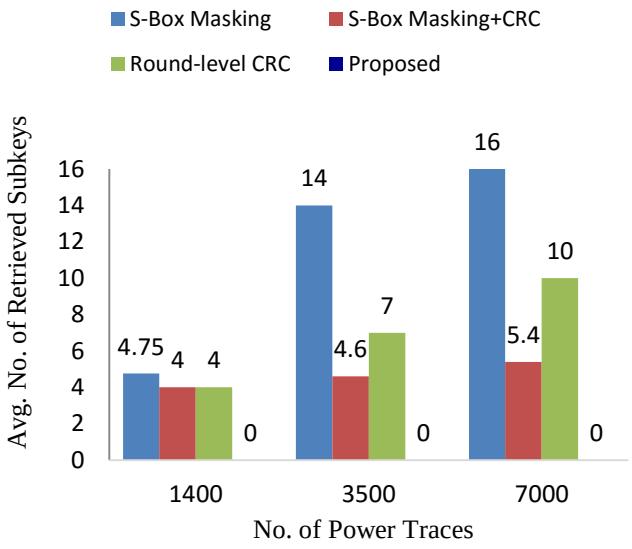
Application of a masking vector is used as a countermeasure to thwart CPA via misleading the power estimation. In Fig. 6.12 (a), our experiments have shown that the simple masking on the S-Box does not effectively strengthen the AES against CPA attack. However, when the masking technique is integrated with CRC, the CPA resistance of the protected AES can be improved.

As shown in Fig. 6.12(a), the joint of Masking and CRC method (hereafter named as masking+CRC) for the S-Box increases the average APGE by 1042 compared to the masking method. As the APGE performance for multiple trials is very close, we took 5 trials to obtain the each average APGE value. The masking process in the joint of Masking and CRC is static, and the

masking vector is a predetermined single vector. The proposed method utilizes dynamic masking and different ECC microarchitecture further improves the average APGE by up to 1577, 535, and 1358 over the masking method, the masking+CRC method, and round-level CRC method respectively.



(a)



(b)

Fig. 6. 12 Comparison of (a) average APGE, and (b) the number of retrieved subkeys in Proposed and other methods.

Figure 6.12(b) shows the improvement achieved by our method from another point of view. The CPA attack on our method does not obtain any subkeys. In contrast, the masking+CRC method leaks approximately 4 or 5 subkeys through the CPA attack.

6.6.3. Evaluation of the Resistance to FA Attacks

We assessed the FA resistance of different methods in Fig. 6.13. As can be seen, if the masking vector is known, the attacker can successfully conduct a FA attack (i.e. FA success rate of 1). When the masking vector is unknown to the attacker, he/she still can perform FA attack by randomly injecting faults to the S-Box output. However, the corresponding FA success rate is reduced due to the imprecise control on the intermediate state registers. The proposed method introduces the masking vector at runtime to resist the attacker from using the brute-force attempt to find the single masking vector. Consequently, the FA success rate of our method is the lowest one compared to the other methods. As shown in Fig. 6.13, our method reduces the FA success rate by 54% over the masking only method, and 90% over the masking+CRC method.

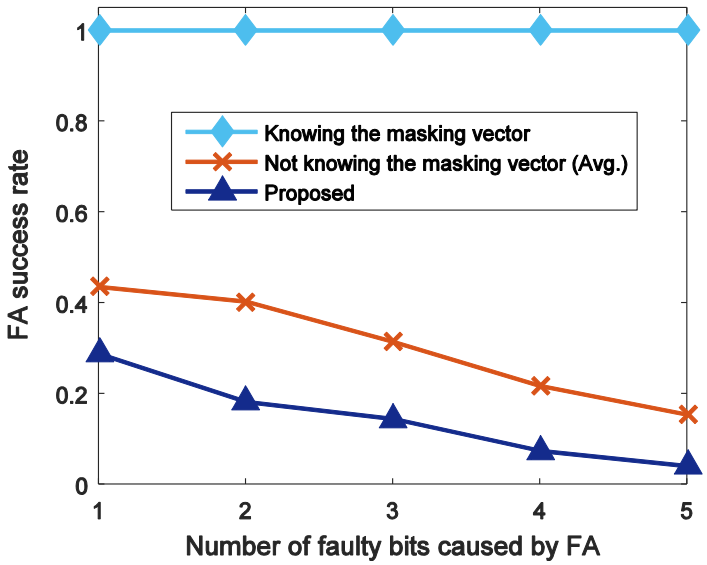


Fig. 6. 13 FA resistance of different methods.

6.6.4. FPGA Cost

Due to the adjustment on the ECC microarchitecture in the AES and the merged masking and ECC circuit, our method reduces the number of slice LUTs (number of occupied slices) cost by 29.5 % (29.2%), 47.1% (44.0%) and 13.7% (14.0%) compared with the S-Box Masking, S-Box masking+CRC, and round-level CRC methods, respectively. Table 6.5 provides the detailed FPGA cost for different methods.

Table 6. 5 FPGA cost for different countermeasures.

Design	Number of Slice Registers	Number of Slice LUTs	Number of occupied Slices
S-Box with Masking	765	2982	914
S-Box with Masking+CRC	757	3975	1169
Round-level CRC	769	2,435	756
Proposed	750	2101	647

6.5.Conclusions

The threats from SCA and FA attacks challenge the integrity and security of cryptographic engine. Although countermeasures for each type of physical attack have been extensively studied in the existing literatures, the impact of the countermeasure for one type of attack on the efficiency of another type of attack has not been well explored. The lack of such study might impede the development of efficient countermeasures for the emerging combined attacks. In this work, we perform a comprehensive and systematic study on the different factors in a FA-resistant AES that affect the key retrieval speed of the CPA attack. Together with the use of different fault detection codes, we assess the impact of power model, hardware cost (FPGA slices), the S-Box implementation method, the linearity (or non-linearity) property of the module under protection,

and FPGA synthesis optimization options on the key retrieval speed. Moreover, we propose a unified countermeasure to thwart the combined CPA and FA attack. Our method integrates a dynamic masking technique with an ECC-based error deflection mechanism to mask the intermediate state value with different masking vector at runtime, thus preventing the accurate power prediction in the CPA attack. Our FPGA-based experimental results show that the proposed method can successfully thwart the CPA attack for given 7000 power traces and reduce the FA success rate by 54% and 90% over the masking only and the joint of masking and CRC methods. In future work, we will compare our method with more combinations of the existing FA countermeasure and SCA countermeasures.

Chapter 7. Dynamic CRC for Reliable and Secure Systems

7.1. Introduction

On-chip communication system uses cryptographic units to prevent the credential information from leaking that results in information security vulnerability [89]. As a minor change on the encrypted message leads to a significant modification on the decrypted plaintext, error control coding (ECC) is typically appended after the cipher text. The ECC codec detects or/and corrects the noise-induced error possibly caused by natural faults in the form of single, multiple, and burst error(s) on the cipher text [90]. Adding the ECC to the communication channel adds reliability in addition to the security provided by the cryptographic algorithm [91]. Exploiting non-systematic ECC methods increases the hardiness of the code against data corruption [92]. For instance, a non-systematic CRC encoding inherently permutes the message and check bits that results in thwarting passive attacks [93].

As mentioned in [94] more than 10% of the attacks target the memories. Therefore, the existing work [88] proposes to use an extra ECC before the memory in addition to the error correcting codes that are incorporated in the memory to obfuscate the data. The existence of ECCs for detecting and possibly correcting errors in the memory is a motivation for the attacker to target the ECC and counteracts the impact of it. One way to cancel out the ECC is implementing Hardware Trojans that make the memory untrusted [95]. The Hardware Trojan has to be designed to be able to activate when an error is injected into the system. Therefore, the Hardware Trojan can camouflage the impact of ECC. As the location of memory contents are remapped, the ECC encoder and decoder naturally hardens the system against hardware Trojan insertion [88]. Unfortunately, the work [88] has not provided a thorough analysis on their security vulnerability in terms of reverse engineering attack time, complexity and success rate. This work fills in this gap.

The main contributions of this work as follows.

- We analyze the impact of the degree and irreducibility of CRC generator polynomials, and the message length on the cost and speed of the generator polynomial retrieval. We found that the use of irreducible polynomials will significantly reduce the CPU time for reverse engineering the applied generator polynomial.

- We propose a method that dynamically alternates multiple polynomials for CRC encoding. Through our analysis, we identify two dependent factors for the efficiency of the proposed method: the ratio of the number of irreducible polynomials to that of reducible polynomials, and the stabilization period used in the process of reverse engineering attacks.

- We compare the polynomial retrieval time for a single polynomial CRC and the proposed dynamic CRC method. In addition, we provide a trend for the reverse engineering time cost over the polynomial degree.

The remainder of this work is organized as below. In Section 7.2, we introduce the system interested in this work and the targeted security attack model. In Section 7.3, we analyze the security vulnerability of the existing work [88]. In Section7.4, we propose a dynamic polynomial alternation method. We analyze the dependent factors for the security vulnerability of our method in Section 7.5 and assess the hardware cost in Section 7.6. We conclude this work in Section 7.7.

7.2. Preliminaries

7.2.1. Abstract of Target System

We aim to protect the on-chip communication system as shown in Fig. 7.1. The data from the processing intellectual property (IP) core are encrypted and the cipher text is further protected with a CRC encoder. We assume the processing IP core, the encryption/decryption unit, and the CRC

encoder/decoder are inside the trusted zone. The attacker can access the CRC codewords from the untrusted zone, which is composed of the on-chip interconnect and the memory IP core. The attacker will collect the codewords and execute passive attacks to, for instance, retrieve the polynomial used in the CRC codec (thus performing fault attack). We also assume that the cipher applied in the cryptographic modules is a block cipher, for example, Advanced Encryption Standard (AES).

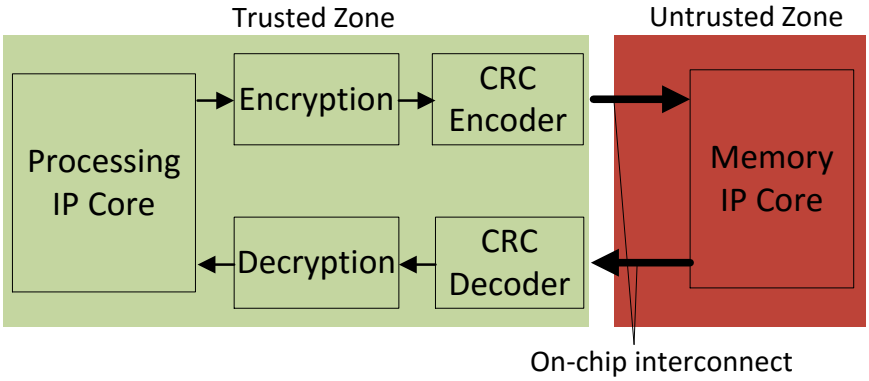


Fig. 7. 1 Target system interested in this work.

7.2.2. Symbols and CRC Encoding Algorithms

The symbols used in the following discussion are listed in Table 7.1.

Table 7. 1 Symbols Used in This Work.

C(x)	Codeword polynomial
N	Codeword length
M(x)	Message polynomial
m_j	The j^{th} bit in the message M(x)
G(x)	Generator polynomial
g_j	The j^{th} coefficient in G(x)
HpG(x)	hypothesized generator polynomial
R(x)	Remainder polynomial
Deg[·]	Polynomial degree
k	G(x) degree

The CRC encoding can be executed non-systematically (Eq. (7.1)) or systematically (Eq. (7.2)).

$$C(x) = \sum_{j=0}^{N-k} \left(\left(m_j x^j \right) G(x) \right) \tag{7.1}$$

$$C(x) = x^k M(x) + R(x) \tag{7.2.a}$$

$$x^k M(x) = Q(x) G(x) + R(x) \tag{7.2.b}$$

The non-systematic CRC reorders the message bits and check bits. Without the knowledge of the generator polynomial, the positions of message bits are unknown. In contrast, the systematic CRC encoding clearly separates the message bits and check bits due to the multiplication of x^k in Eq. (7.2). From the attacker's point of view, the non-systematic encoding is more difficult to be reverse engineered.

7.2.3. Attack Model

This attack model is based on the assumption that the adversary can access the on-chip interconnect and memory IP core to obtain the codeword from the untrusted zone. The goal of the attacker is to recover the CRC generator polynomial, which is the first step before attacking the crypto modules in the trusted zone. The assumed attack model is based on exhaustive search, which means systematical checking all possible polynomials until find the exact G(x) that applies to all of the collected codewords. The detailed flowchart for the reverse engineering process is shown in Fig.7.2. The only difference between non-systematic and systematic CRC is highlighted in the gray area.

The first attempt to recover G(x) in the systematic CRC starts by extracting the hypothetical message from the observed codeword. The first few LSBs of the codeword are considered as the remainder and the rest as the message. The Table 7.2 shows how the message is extracted at each attempt. Consider the case where the codeword is: $[c_1\ c_2\ c_3\ \ldots\ c_{n-3}\ c_{n-2}\ c_{n-1}\ c_n]$.

Table 7. 2 A presentation of extracting the check bits for reverse engineering.

# of Attempts	Message	Check bits	Degree of G(x)
1	$[c_1\ c_2\ c_3 \ldots\ c_{n-3}]$	$[c_{n-2}\ c_{n-1}\ c_n]$	3
2	$[c_1\ c_2\ c_3 \ldots]$	$[c_{n-3}\ c_{n-2}\ c_{n-1}\ c_n]$	4
:	:	:	:
n-3	$[c_1]$	$[c_2\ c_3 \ldots\ c_{n-3}\ c_{n-2}\ c_{n-1}\ c_n]$	n-1

For each extracted message the degree of G(x) can be found from the length of the remainder. For instance, if the length of the remainder is 4 bits the degree of G(x) is 4. Once the degree of G(x) is found we can list all the possible polynomials and pair them with the extracted message at a time. The hypothetical codewords will be generated by using each pair of the extracted message and G(x). All the hypothetical codewords have to be compared with the original codeword. If the hypothetical and original codewords are same it means that there is a possibility that the chosen G(x) is the correct polynomial. Since, the combination of message and G(x) generates the codeword in CRC, there could exist multiple combinations of message and G(x) polynomial that result in the same codeword. In addition our results show that even for a fixed message there exist multiple generator polynomials that give the same codeword. Therefore, if the whole process is run multiple times for different codewords, then we can retrieve the correct G(x) or at least determine a set of possible polynomials.

7.3. Security Vulnerability of the Existing Work

In the previous work [88], a single generator polynomial is used to address both reliability and hardware Trojan issues. In this section, we assess the security vulnerability of that method.

7.3.1. Theoretical Analysis

Without losing the generality, we take $G(x) = x^6 + x^4 + 1$ (i.e. 81 in decimal) as an example to explain how the reverse engineering can retrieve the G(x) *without knowing the message length and*

the degree of the applied generator polynomial. We implemented the reverse engineering attack algorithm shown in Fig. 7.2 in MATLAB. The tentative G(x) list is updated through multiple codewords. As shown in Table 7.2, after two codewords, we can narrow down the G(x) list to two possible generator polynomials, 13 and 81 in decimal. No matter how many new codewords are used, we cannot further filter out one of the possible G(x) in Table 7.3.

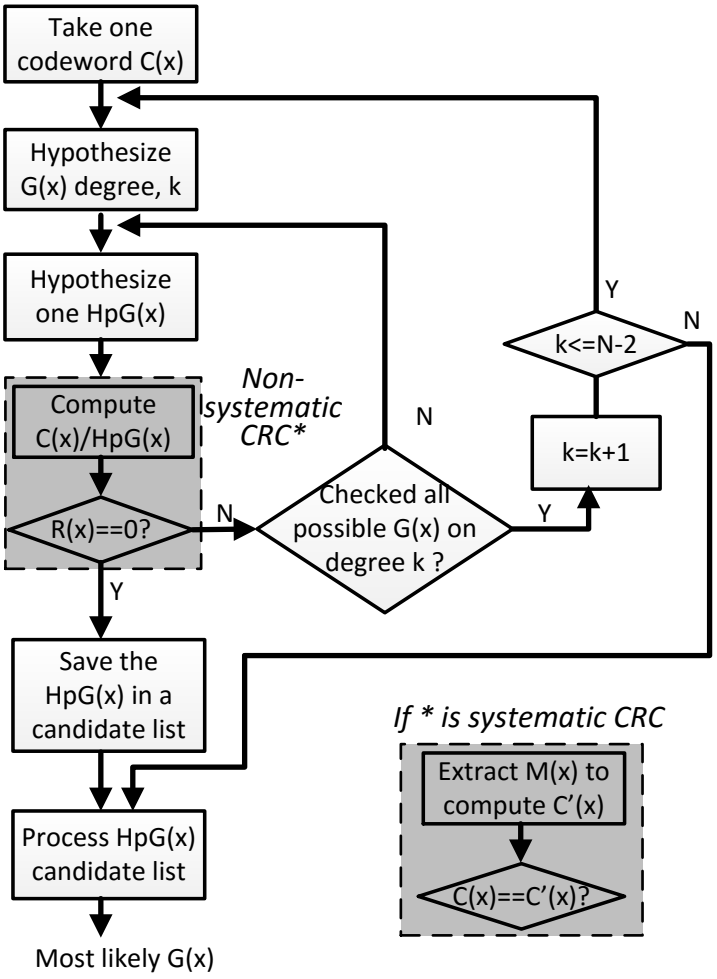


Fig. 7. 2 Flowchart of retrieving the possible G(x) by reverse engineers.

This is because the $G(x) = x^6 + x^4 + 1$ is a reducible polynomial. As expressed in Eq. (7.3), this G(x) can be further broken into a square of the same sub-polynomial $x^3 + x^2 + 1$ (13 in decimal).

$$G(x) = x^6 + x^4 + 1 = (x^3 + x^2 + 1)(x^3 + x^2 + 1) \tag{7.3}$$

Table 7. 3 Retrieval of reducible G(x) using brute-force method.

Codeword / Deg[G(x)]	C1	C2	C3	C4
2	0	0	0	0
3	13	13	13	13
4	0	0	0	0
5	35, 59	0	0	0
6	81	81	81	81
7	161	0	0	0
8	439, 271		0	0
9	0	0	0	0
10	1837	0	0	0
11	3403	0	0	0
12	0	0	0	0
13	0	0	0	0
14	0	0	0	0

Therefore, any C(x) that can be divided by G(x) (81 in decimal) can also be divided by any polynomial equal to 13. While other polynomials that fit for one codeword do not work for other codewords. As more codewords are under examination, any G(x) that is not equal to 13 or 81 will be filtered out. Thus, the attacker can at least have a small list of possible G(x). Interestingly, the G(x) with the largest decimal number is the correct polynomial used in the CRC encoder. This is because the largest polynomial is composed of smaller polynomial terms.

We repeated the same experiment with another G(x) expressed in Eq. (7.4). As this polynomial is an irreducible one, the G(x) retrieval is completed successfully after two codewords, as shown in Table 7.4.

Table 7. 4 Retrieval of irreducible G(x) using brute-force method

Codeword / Deg[G(x)]	C1	C2	C3	C4
2	0	0	0	0
3	11	0	0	0
4	0	0	0	0
5	0	0	0	0
6	97	97	97	97
7	217		0	0
8	0	0	0	0
9	939	0	0	0
10	3403	0	0	0
11	1087	0	0	0
12	0	0	0	0
13	0	0	0	0
14	0	0	0	0

$$G(x) = x^6 + x + 1 \tag{7.4}$$

The examples shown in Tables 7.3 and 7.4 indicate that the choice of the generator polynomial determines whether the exact G(x) can be recovered by analyzing several codewords. **The use of an irreducible polynomial for CRC encoding leads the system to be vulnerable to reverse engineering attacks**. The reducible polynomial could improve the resistance against reverse engineering attacks at some degree, but the attacker is still able to narrow down the list of possible generator polynomials and successfully recognize the exact G(x) used in the CRC encoder.

7.3.2. Number of Codewords Needed for G(x) Retrieval

Now, we examine the average number of codewords one needs to retrieve the G(x) applied in the CRC encoder through brute-force attempts. As non-systematic encoding has better permutation performance, we examine the non-systematic encoding case. One hundred random generator polynomials for each degree were used in the following experiments. If the reverse engineering process cannot narrow down the possible polynomial list to a single option, we took the largest polynomial in decimal (based on the observation in Table 7.3). As shown in Fig. 7.3, we only need 3 or 4 codewords to retrieve the exact polynomial used in the CRC encoder. The standard deviation

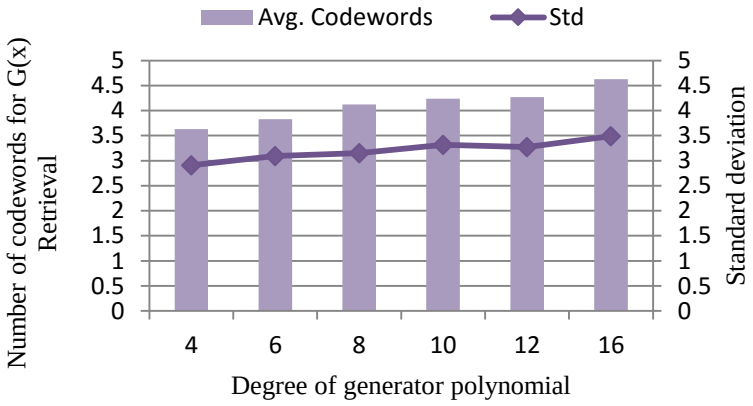


Fig. 7. 3 Average number of codewords for recovering the G(x) applied in the CRC encoder for message length of 8.

is below 4. Next, we changed the message length. As shown in Fig. 7.4, the varied message length only slightly changes the average number of codewords needed for reverse engineering. However, it indeed changes the total simulation time needed for G(x) retrieval. Because the message length is unknown to the attacker, the longer message length means more degrees of hypothesis G(x) should be examined, thus more brute-force time cost.

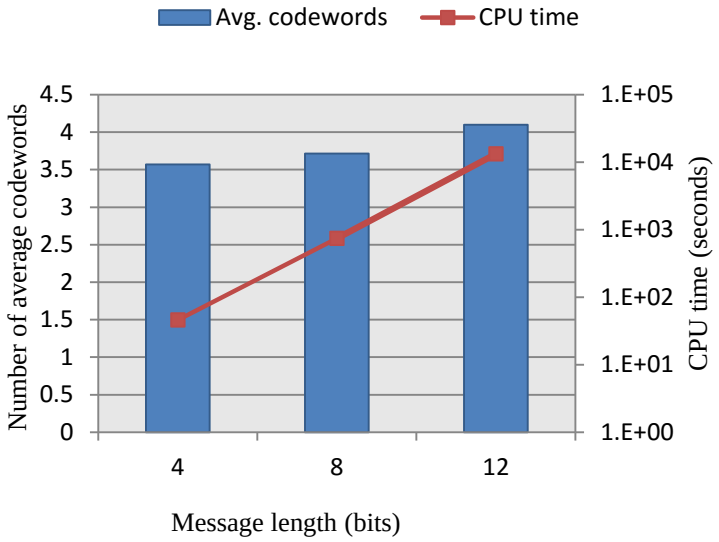


Fig. 7. 4 Average number of codewords to recover G(x) (degree 8) for message lengths of 4, 8, 12

7.4. Proposed Dynamic Polynomial Alternation Method

7.4.1. Method Overview

To enhance the system's resistance against the reverse engineering attack from the untrusted zone shown in Fig. 7.1, we propose a dynamic polynomial alternation method to the CRC encoder. A non-systematic CRC encoding inherently provides a bit permutation capability to reorder the original message bits. Thus, without knowing the applied CRC generator polynomial, the message bits are uncertain. The proposed method further improves the uncertainty through alternating the generator polynomials at runtime. As shown in Fig. 7.5, we use a generator polynomial selector (*GSel*) to dynamically select one G(x) from the polynomial candidates ($G_1(x),\ldots G_n(x)$). A user-

specified algorithm (only available to the trust zone designer) is applied in the *GSel*. The encoder

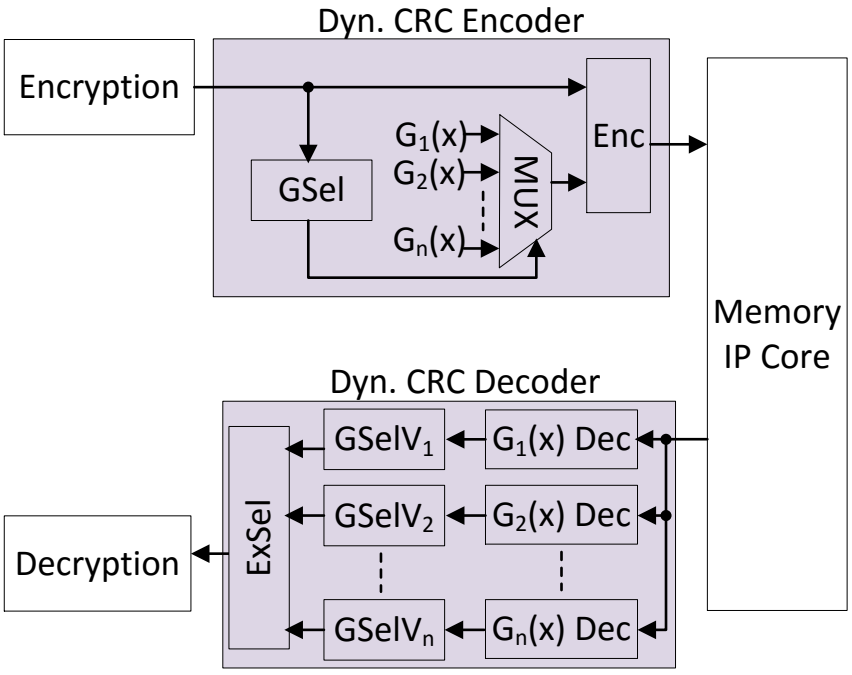


Fig. 7. 5 Proposed dynamic polynomial alternation method for CRC codec.

(*Enc*) is configured based on the G(x) selected at runtime. For a serial CRC encoder implementation, the G(x) selection can be implemented as a switched controlled exclusive OR on each shift register [96]. If parallel CRC generator is implemented, the polynomial selection in Fig. 7.5 could be designed as multiple encoder, one G(x) for each *Enc* unit. The output of the *GSel* unit depends on the incoming message.

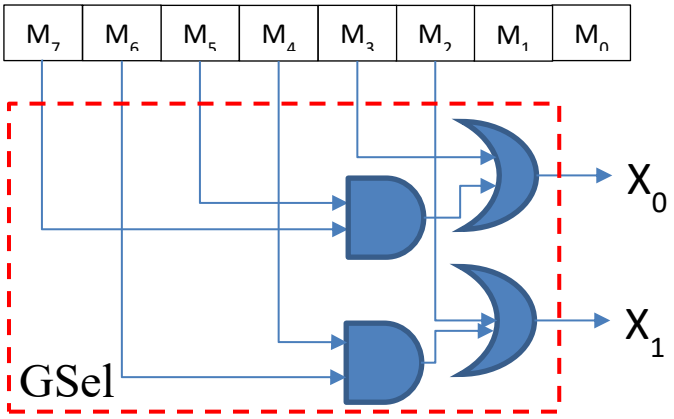


Fig. 7. 6 An example of the GSel block.

The dynamic G(x) polynomials are selected through a selection function shown as GSel in Fig. 7.5. The selection function utilizes the message bits to generate the select signal. Figure 7.6 illustrates an example of GSel block with the assumption of a 2bit select line. By using the message bits in the polynomial selection process an unpredictable select signal is generated. Therefore, the chance of having a fixed pattern reduces that adds another layer of security in addition of using multiple generator polynomials.

Due to the unknown selection pattern of the generator polynomial, every codeword has to be decoded by using all the dynamic generator polynomials used in the encoder side. According to the number of the polynomials used in the dynamic CRC encoder there are decoder blocks in the decoder side. Each decoder block uses one of the generator polynomials for decoding the received codeword from the memory IP. For both systematic and non-systematic CRC the remainder of the long division (codeword as the dividend and G(x) as the devisor) is investigated. A zero remainder in both systematic and non-systematic CRC indicates that the syndrome is zero which means there is no error in the codeword and the message can be extracted safely. In case of having a systematic CRC the message is the remaining bits when the check bits are removed. If a non-systematic CRC is used the quotient is simply the message.

The simulation result on a systematic CRC is shown in Fig. 7.7 indicates that the possibility of having a zero remainder for an incorrect G(x) is below 10% when the dynamic generator polynomials are chosen with/without common factors and from the irreducible polynomials. In addition, if one of the dynamic generator polynomials be the common factor of the rest of G(x)s the chance of multiple zero remainders is 100% and this G(x) should be avoided.

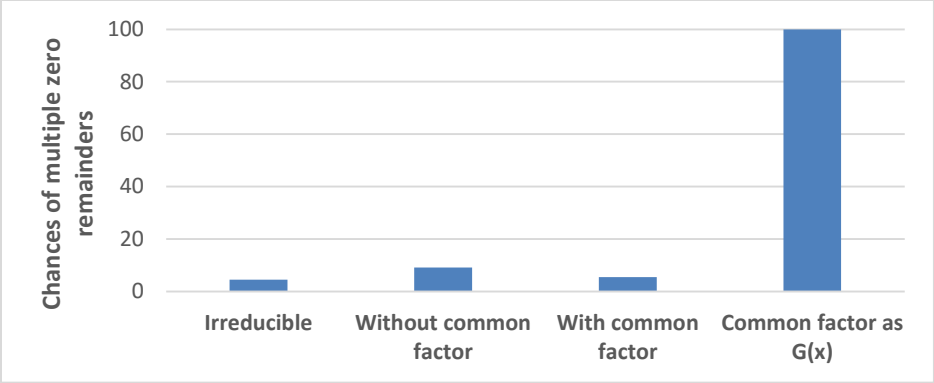


Fig. 7. 7 Chances of incorrect multiple zero remainders in the dynamic CRC.

As the G(x) selection signal depends on the real-time message, the GSel output is dynamic without a pre-determined pattern. Clearly, it is not realistic to record either the selection signal for each CRC codeword or the dynamic pattern used in the CRC encoding process. To correctly retrieve the message encoded by the dynamic CRC encoder, we will decode the codeword with all applied generator polynomials and select the recovered message from the decoder resulting in a zero remainder. For a rare situation, two or multiple decoder ($G_i(x)$ *Dec*) will simultaneously generate zero remainders. To address this issue, we propose to use the $GSelV_i$ unit and verify whether the reconstructed the generator polynomial $G_i(x)$ matches to the one used in the $G_i(x)$ *Dec*. The exclusive selector (*ExSel*) will provide the single recovered message. If ExSel cannot conclude a single message (due to errors injected in the untrusted zone), that message will be dropped.

To defeat the proposed countermeasure, the attacker will confront three questions: (1) How many generator polynomials are used in the system? (2) What are the polynomial degrees? (3)

How do different polynomials switch? These three questions provide a new barrier to prevent the attacker from attack success, compared to the single CRC method in [88].

In the next subsection, we first evaluate the proposed method with a modified reverse engineering attack model, and then assess the dependent factors that affect the efficiency of our method.

7.4.2. Selection of Multiple Generator Polynomials

If multiple polynomials at the same degree are alternatively applied to encode the message, the attacker can retrieve a list of possible G(x), which becomes stable after a few codewords. The assumption on *multiple polynomials at the same degree* is made to simplify the reverse engineering process. We extend the attack flowchart shown in Fig. 7.2 to retrieve a ranked polynomial list. Different than the process for the single G(x) scenario, we use a G(x) matrix to accumulate the possible polynomial that results in a zero reminder on different degrees of hypothesized G(x). Because of the assumption on *multiple polynomials at the same degree*, we eliminate some rows of the G(x) matrix. The true polynomial set for dynamic polynomial application in CRC should be applied to all messages. Therefore, if two codewords suggest two polynomials on different degrees, we can conclude that these two degrees are not the one used in the dynamic CRC encoding.

In this example, we alternatively use one of the generator polynomials in Eqs. (7.5) and (7.6), as well as Eq. (7.4).

$$G(x) = x^6 + x^5 + x^3 + 1 \tag{7.5}$$

$$G(x) = x^6 + x^5 + x + 1 \tag{7.6}$$

Table 7.5 shows the retrieval process of dynamic polynomial alternation method. At the end, only one polynomial degree is left for future accumulation. As soon as the entire reverse engineering process is complete, we calculate the occurrence frequency of each generator polynomial in that G(x) degree row, and set a priority for each possible G(x). Although in the list itself, we cannot tell the alternative frequency of each polynomial, we are able to narrow down the G(x) list and stabilize the possible G(x). Thus, this example proves that the adversary can successfully retrieve the dynamic G(x) list. The reverse engineering for single G(x) retrieval checks the occurrence of a unique G(x) in the G(x) matrix. In contrast, the process for multiple G(x) retrieval cannot use such condition as a convergence criterion.

Instead, we use a *stabilization period* to indicate how many consecutive codewords use the same multiple G(x) list. This metric also indicates our confidence level on the retrieved G(x) results. In Section 7.5, we discuss the impact of the stabilization period length on the attack success rate.

7.5. Dependent Factors of Proposed Method against Reverse Engineering Attack

7.5.1. Number of Trails in G(x) Examination

To save simulation time, we chose a small degree of a G(x) to assess how the number of trails adopted in the G(x) examination process affects the G(x) retrieval success rate. We varied the number of trails from 20 to 200 per each test, and compared the average number of CRC codewords needed for G(x) retrieval. As shown in Fig. 7.8, the required number of codewords varies from 19.15 to 19.43 (which is more than that needed for single G(x) method), and retrieval success rate changes from 25% to 42%. After the number of trials exceeds 100, the improvement on G(x) retrieval success rate is < 5%.

Table 7. 5 G(x) Retrieval Process for the Proposed Multiple Polynomials through four codewords.

Codeword / Deg[G(x)]	C1	C2	C3	C4
2	3	3	3	0
3	5	0	0	0
4	0	0	0	0
5	0	0	0	0
6	33	33, 35	33, 35, 39	0
7	99	99, 81, 101	99, 81, 101, 105, 87	**99**, **81**, 101,**105**, 87
8	0	0	0	0
9	341	341, 439	341, 439, 267	0
10	0	0	0	0
11	1025	0	0	0
12	0	0	0	0
13	0	0	0	0
14	0	0	0	0

7.5.2. Stabilization Period

In the previous subsection, we stopped the G(x) retrieval if the G(x) list becomes stable for 10 consecutive codewords. As indicated in Fig. 7.9, the stabilized G(x) list is not always the correct list of the multiple generator polynomials adopted in the CRC codec, we examine the impact of the stabilization period length on the G(x) retrieval speed and success rate. In dynamic CRC, we cannot narrow down the G(x) list to a single decimal number. The best we can do is to prioritize the possible G(x) decimal and recommend the most likely one. We vary the length of stable period from 10 to 30. As shown in Fig. 7.9, the G(x) retrieval success rate for G(x) degrees on 4, 6, and 10 increases with the length of the stable period. The exceptional case is for degree 8. This is because the majority of the G(x) polynomials are reducible.

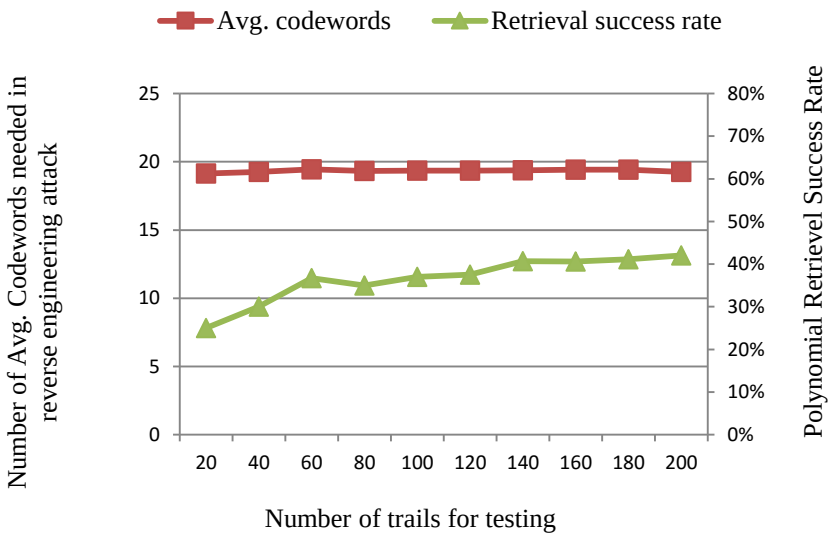


Fig. 7. 8 Impact of the number of test trails on G(x) retrieval success rate.

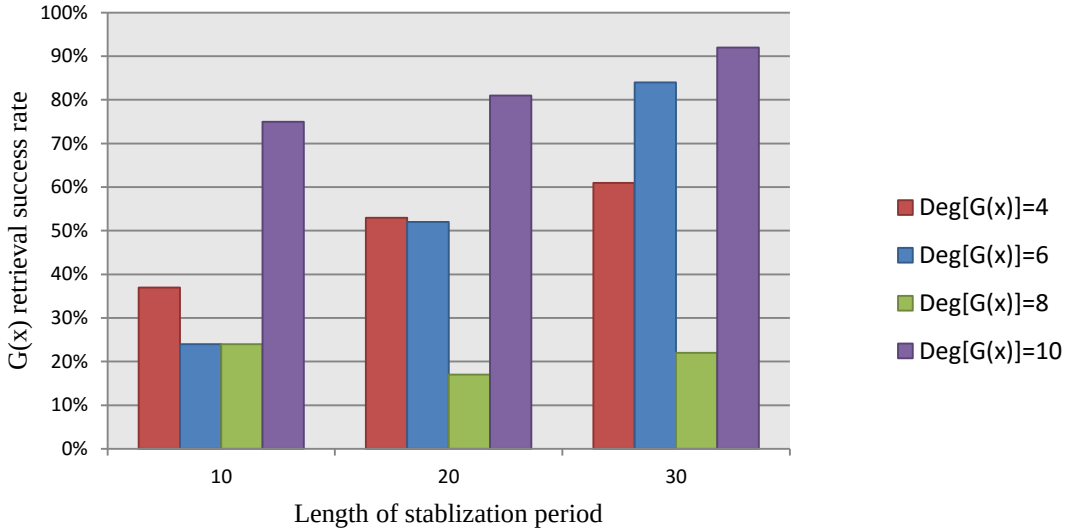
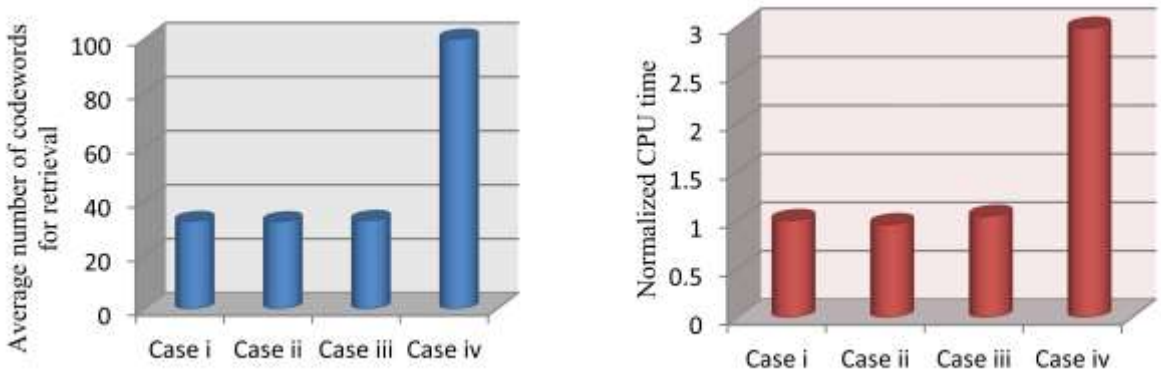


Fig. 7. 9 Impact of stabilization period on the retrieval success rate.

7.5.3. Combination of Different Irreducible Polynomials

The relationship between the G(x) retrieval success rate and the degree of G(x) depends on the polynomial we chosen for the dynamic G(x) scheme. Now, we categorize the polynomials for the dynamic G(x) scheme as (i) all irreducible G(x), (ii) 2/3 irreducible and 1/3 reducible G(x), (iii) 1/3 irreducible and 2/3 reducible G(x), and (iv) all reducible G(x). From Fig. 7.10, it can be seen that the polynomial combinations for the dynamic list indeed matters. As the case iv uses all reducible G(x), the exact G(x) list cannot be retrieved (we set the maximum number of messages for each case to 100). Meanwhile, the case iv also consumes 2.98X simulation time than the case i. This experiment explains why the trends observed in Fig. 7.9 are not monotonically increasing with the polynomial degree and stabilization period.

7.5.4. Dynamic Reducible and Irreducible Generator Polynomials



(a) (b)

Fig. 7. 10 Impact of percentage of irreducible G(x) in the dynamic generator polynomial list on (a) the retrieval success rate, and (b) normalized CPU time.

Table 7.5. Possible recovered generator polynomials for three different cases.

Reducible G(x)s w/common factor		Reducible G(x)s w/o common factor		Irreducible G(x)s	
A	A	A	D	A	B
B	D	B	E		
C	C	C	F		
AB	AD	AB	DE		
BC	DC	BC	EF		
AC	AC	AC	DF		
ABC	ADC	ABC	DEF		
ABC	ADC	ABC	DEF	A	B

After filtering out the non-repeated degrees in the hypothetical G(x) matrix, the possible number of remaining elements for each codeword depends on the number of irreducible polynomials that

each G(x) is composed from. For example, a generator polynomial like ABC that is composed of three primitive polynomials; A, B, and C could have A, B, C, and their combinations (AB, AC, and BC) as the hypothetical G(x) polynomials in addition to the original polynomial ABC. This type of dynamic generator polynomials are defined as reducible G(x)s with common factor. In contrary, the type of dynamic generator polynomials that are reducible but have no common factors are defined as reducible G(x) without common factors. Table 7.5 indicates three examples for three different cases: (1) the dynamic G(x) polynomials are all reducible and have common primitive polynomials, (2) the dynamic G(x) polynomials are reducible and have no common factors, and (3) all the G(x) polynomials are irreducible. In the reducible with common factor category, the two G(x) polynomials ABC and ADC are considered to have two primitive polynomials in common (A and C). As the number of common primitive polynomials in the G(x) increases, the chance of getting a higher rank for incorrect generator polynomials increase as well. Therefore, the chance of finding the right set of dynamic G(x) polynomials decreases. The G(x) recovery probability of irreducible polynomials are the highest that is due to the uniqueness of the polynomials in this category. Since the reducible G(x) polynomials with common factors are composed of unique primitive polynomials, the chance of retrieving the correct G(x) set is less than the irreducible and more than the reducible G(x) polynomials with common factors. Figure 7.11 shows the recovery chance for message lengths of 8 and indicates the fact that the reverse engineering recovery chance decreases for dynamic G(x) with reducible polynomials.

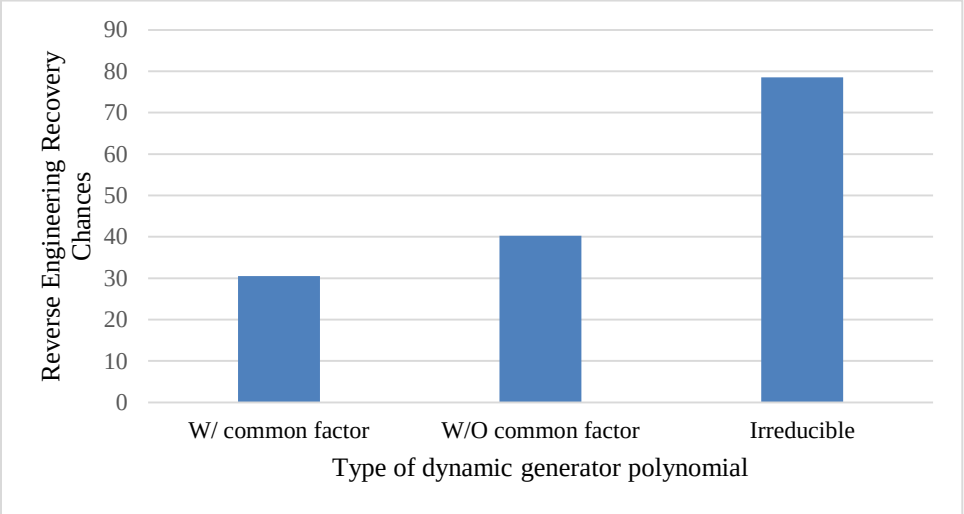


Fig. 7. 11 Success rate of reverse engineering the dynamic G(x) polynomials for three cases.

According to Fig. 7.11 in order to decrease the G(x) recovery chance by the attacker, the designer has to choose multiple polynomials that have common factors between them.

7.6. Error Detection Rate of the Proposed Dynamic CRC

In this section the reliability of the proposed dynamic CRC is investigated. Figure 7.12 presents the simulation results when 1-bit, 2-bit, and 3-bit (burst and random) error(s) were injected between the encoding and decoding stages. This location includes the memory IP that is assumed untrusted and could be vulnerable to natural faults. The message length is considered 8 bits and the simulation results have been repeated 1000 times to diminish the impact of the chosen random message and increase the accuracy of the results. The dynamic generator polynomials for the error detection of 1-bit and 3-bit errors were kept same since such G(x) can detect all the odd number of error bits. In addition, the error detection rate for 1-bit and 3-bit error(s) is 100% that shows the

Fig. 7. 12 Error detection rate in the proposed dynamic CRC.

proposed dynamic CRC did not affect the error detection rate. Another set of dynamic generator polynomials were used to detect 2-bit errors. In contrary, the error detection rate for 2-bit errors was affected by the dynamic CRC and reduced by 1% comparing with the case when a single generator polynomial was used. Although, the error detection rate for 2-bit errors has decreased but the reduction is not significant and the proposed dynamic CRC can still be considered as a reliable detection codec.

7.7. Time Cost and Hardware Overhead

7.7.1. Time Cost

We used a Windows desktop with an Intel Core i5-2400 CPU@3.10GHz and 4GB RAM to run the reverse engineering attack presented in Fig 7.15 on the single G(x) CRC and the proposed dynamic CRC. Each point on the graph in Fig. 7.26 represents the averaged CPU time required for a successful reverse engineering attack for a fixed generator polynomial degree. Figure 7.13 shows that, compared to the method in [88], our method improves the CPU time that the attacker needs to retrieve the G(x) by 27X for the generator polynomial of degree 16. The ratio presented in Fig.

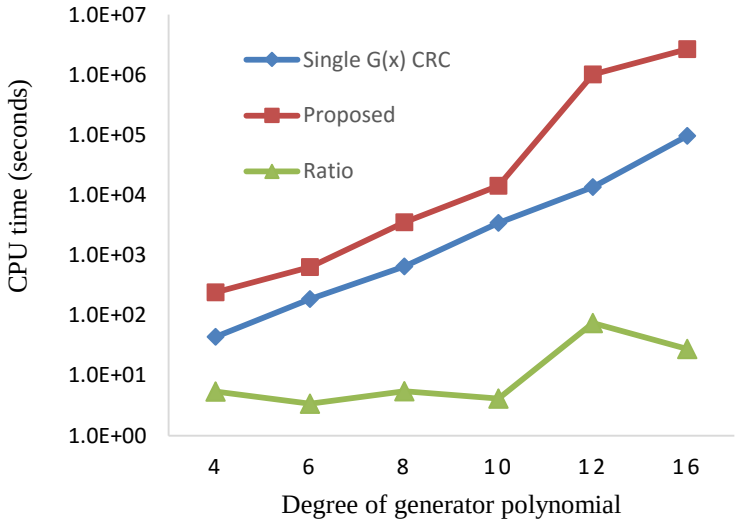


Fig. 7. 13 Simulation time comparison for single G(x) CRC and proposed dynamic CRC.

7.13 is the CPU time of the proposed method over the single G(x) CRC. The ratio of the required CPU time gives a more accurate understanding of the extent of the increase in the CPU time. We used a trend line to predict the CPU time for degree 32. As shown in Fig.7.14, it will take the attacker 6.3 months to retrieve the G(x) list from our method.

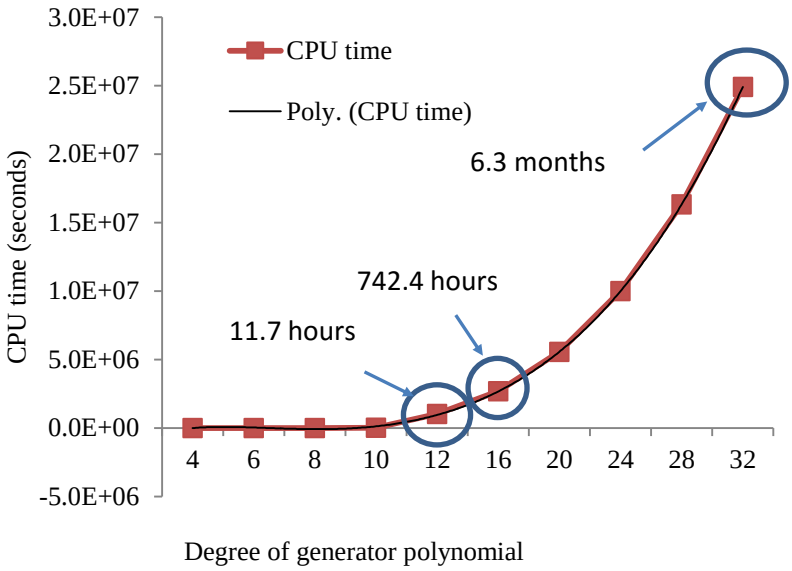


Fig. 7. 14 CPU time for reverse engineering the proposed method.

7.7.2. Hardware Cost Comparison

We implemented a pipelined AES-128, a CRC encoder with single G(x) and a proposed dynamic CRC encoder with Verilog HDL. The source codes were synthesized in Synopsys Design Vision with a 65nm TSMC library. Due to the long delay on CRC32, we set the clock period to 5ns for all the comparisons in this section. Three polynomials are used in the dynamic CRC. As shown in Table 7.6, the proposed dynamic G(x) CRC only increases the area overhead by 0.6% and 2.9% over the single G(x) CRC on degree 8 and degree 32, respectively. Our method consumes 0.9% and 3.8% more power than the single G(x) CRC on degree 8 and degree 32, respectively.

Table 7.6. Area and Power Consumption Comparison.

Design vs. Cost	Area (μm^2)		Total Power (mW)	
Design	Single G(x) CRC	Dynamic G(x) CRC	Single G(x) CRC	Dynamic G(x) CRC
AES128+CRC8	157478.6 (100%)	158477.9 (100.6%)	13.4833 (100%)	13.6003 (100.9%)
AES128+CRC32	159147.9 (100%)	163767 (102.9%)	13.7029 (100%)	14.2206 (103.8%)

7.8. Conclusion

Reliability and security are both important for systems-on-chip. The integration of cryptographic module and error control coding module could be a promising method to jointly address reliability and security. In this work, we analyze the security vulnerability of using a single generator polynomial in CRC, and provide a polynomial selection guideline. Furthermore, we propose a dynamic polynomial alternation method to improve the system capability against reverse engineering attacks on the CRC polynomials through codeword analysis. Simulation results show that our method increases the reverse engineering time by 27X over the single polynomial CRC method at the cost of 2.9% and 3.8% more area and power.

Chapter 8. Conclusion and Future Work

8.1. Error Latching Probability Assessment by a Systematic Analyses Method

The impact of SETs have increased over the time due to the smaller technology size, higher frequency, and lower operating voltage. Therefore, it is imperative to consider the effect of SETs along with SEUs in reliability assessment of a circuit under test. Investigation on the impact of SETs on soft error rate is more complicated than that on SEUs. This is because of the randomness of SET pulse width, SET injection timing and the affected logic type. One of the major factors that have a huge influence on assessing the reliability of a circuit is the soft error rate (SER). The SER due to SETs was typically studied via measurements on test chips by radiating high energy charged particles on the circuit under test in a controlled environment. Although accurate, physical measurements are expensive and time consuming. To save cost, Monte-Carlo simulations on switch-level, circuit-level and system-level have been extensively used at the cost of long simulation time. To reduce the cost and simulation time, a systematic analysis on SER due to SET is proposed.

In this work the analytical model considers logic gate delay and setup/hold time of the memory element in addition to the SET pulse width to provide a systematic approach for evaluating the SER. The systematic approach is based on determining the possible SET pulse width range, the boundaries for the logic gate delay of the circuit and considering the clock period and setup/hold time. As a result, by exploiting the boundaries a set of closed-form expressions for the latching probability under different SET pulse width and logic delay conditions is found. The simulation results show that the accuracy of the proposed analytical model is up to 97.1% for single cycle and 98% for multiple cycle SET pulse widths. Another advantage of the proposed method in addition to accuracy is the speed of the procedure. First based on the environment condition and the location

of the circuit that is wished to evaluate the SER due to SET, the proper boundaries for the SET pulse width and logic gate delay are found. Next, the probability of an SET latching in the memory element is chosen based on the boundaries and is calculated. According to the fact that the probability of error latching is determined through an analytical method and the time consuming random Monte Carlo simulation is avoided, the simulation time reduces by up to 78% in the c6288 circuit.

In future work, it will be interesting to validate our analytical expressions by physical measurements in a controlled environment, where the radiation of high energy charged particles is performed in the form of ion beams.

8.2. CPA Resistance Assessment of AES with Different Fault Detection methods

The combination of Side channel analysis (SCA) and fault analysis (FA) offers a stronger attack for cryptosystems by the adversary and result in compromising the cryptographic algorithms. Our hardware based experimental results show that using multiple countermeasures to address both mentioned types of attacks affects the total power consumption and could possibly result in a more effective CPA attack. Consequently, the key retrieval speed through CPA attack is affected by the use of fault detection methods. We performed a systematic assessment on the impact of different fault detection (FD) methods on the CPA resistance of complete AES implementation. In this work, the impact of using different power models in the CPA attack, the FPGA synthesis optimization choices, and the different S-Box implementations on the key retrieval speed were studied. Furthermore, a unified countermeasure was introduced to thwart the impact of the combined CPA and FA attack. The proposed countermeasure uses the masking technique along with an error control code (ECC). The combination of the masking technique and error deflection ECC creates a situation that prevents the attacker to achieve an accurate power prediction in the CPA attack.

Our FPGA-based experimental results show that the proposed method can successfully thwart the CPA attack for a given 7000 power traces. In addition, the proposed unified countermeasure reduces the FA success rate by 90% comparing with the case when masking is only used as the countermeasure.

As an extension of this work, one can compare our method with more combinations of the existing FA countermeasure and SCA countermeasures. In this work, the impact of the existence of fault detection methods were studied on the effectiveness of CPA. It is interesting to know how the unified countermeasure behaves when fault is introduced to the system and what the success rate of CPA is.

8.3. Dynamic CRC to Thwart Reliability and Security Vulnerability

The weakness of the reliability and security of systems-on-chip need to be addressed simultaneously to avoid possible system failure or compromising the system. One of the common ways to protect the data in memories from random faults and possible hardware attacks is Cyclic Redundancy Code (CRC). CRC scrambles and obfuscates the data before it is written into the memory. According to the results presented in this work the generator polynomial can be recovered by an average of 3 different codewords for either systematic or non-systematic CRC. Therefore, it is imperative to address the reverse engineering threat in CRC and propose a stronger countermeasure. This work, has analyzed the security vulnerability of a usual CRC codec by using one generator polynomial. It has been shown that the CRC security against reverse engineering can be improved by using multiple generator polynomials. Furthermore, a dynamic polynomial alternation method was proposed to increase the attackers challenge in reverse engineering the intended CRC generator polynomial. Simulation results show that the proposed method increases the reverse engineering time by 27X over the single polynomial CRC method at the cost of 2.9%

and 3.8% more area and power consumption. In addition, the error detection rate of the CRC for odd number of errors was 100% for a certain set of generator polynomials. In contrast, the error detection rate for detecting 2-bit errors slightly decreased by 1% when another set of G(x) was used while each generator polynomial in the set had a 100% chance of detecting 2-bits of errors in a single G(x) CRC.

To improve this work, a more detailed guideline on the generator polynomial is needed to provide a good balance of protection against reverse engineering and offer a maximum error detection rate. In addition, it is worth investigating the susceptibility of the system with dynamic CRC against side channel analysis attack.

References

[1] D. Binder, E.C. Smith, and A.B. Holman, "Satellite anomalies from galactic cosmic rays", *IEEE Trans. Nucl. Sci.*, vol. NS-22, no. 6, pp. 2675–2680, 1975.

[2] M. Nicolaidis (Ed.), "Soft Errors in Modern Electronic Systems", *Springer*, 2011.

[3] T. C. May and M. H. Woods, "A new physical mechanism for soft error in dynamic memories," *in Proc. 16th Int. Reliability Physics Symp. (IRPS), IEEE EDS*, pp. 33–40, 1978.

[4] T.J. O'Gorman, "The effect of cosmic rays on the soft error rate of a DRAM at ground level", *IEEE Trans. Electron Devices*, vol. 41, no. 4, pp. 553–557, 1994.

[5] B. Narasimham, et al., "Characterization of digital signal event transient pulse-widths in 130-nm and 90-nm CMOS technologies," *IEEE Trans. on Nuclear Science*, vol. 54, no. 6, pp. 2506–2510, Dec. 2007.

[6] R. Baumann, "Soft Errors in Advanced Computer Systems", *in IEEE Design & Test of Computers*, 2005.

[7] N. Cohen, T.S. Sriram, N. Leland, D. Moyer, S. Butler, and R. Flatley, "Soft error considerations for deep-submicron CMOS circuit applications", *in Int'l Electron Devices Meeting (IEDM) Tech*. Dig., pp. 315–318, 1999.

[8] P. Shivakumar, M. Kistler, S. W. Keckler, D. Burger, and L. Alvisi, "Modeling the effect of technology trends on the soft error rate of combinational logic," *in Proc. IEEE Dependable Systems and Networks Conf.*, pp. 389–398, Jan 2002.

[9] M. A. Aguirre, V. Baena, J. Tombs and M. Violante, "A new approach to estimate the effect of single event transients in complex circuits," *IEEE Trans. on Nuclear Science*, vol. 54, no. 4, pp. 1018–1023, Aug. 2007.

[10] S. Patranabis, D. B. Roy, and D. Mukhopadhyay, "Using Tweaks To Design Fault Resistant Ciphers", *29th International Conference on VLSI Design* , 2016.

[11] M. Tunstall, D. Mukhopadhyay, and S. Ali, "Differential Fault Analysis of the Advanced Encryption Standard using a Single Fault", *Proceedings of the 5th IFIP WG 11.2 international conference on Information security theory and practice: security and privacy of mobile devices in wireless communication*, pp: 224-233, June 2011.

[12] G. Bertoni, L. Breveglieri, I. Koren, P. Maistri, and V. Piuri, "A Parity Code Based Fault Detection for an Implementation of the Advanced Encryption Standard," *Proc. IEEE Int'l Symp. Defect and Fault Tolerance in VLSI Systems (DFT '02)*, pp. 51-59, Nov. 2002.

[13] H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan, "The Sorcerer's apprentice guide to fault attacks," *Cryptology ePrint Archive, Report 2004/10*, 2004.

[14] E. Brier , C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in Lecture Notes in Computer Science, vol. 3156, pp. 16–29. Springer, Berlin, 2004.

[15] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," *In Proceedings of 19th International Advances in Cryptology Conference. CRYPTO'99*, pp.388-397, 1999.

[16] R. Karri, K. Wu, P. Mishra, and Y. Kim, "Concurrent Error Detection Schemes for Fault-Based Side-Channel Cryptanalysis of Symmetric Block Ciphers," *IEEE Trans. TCAD*, vol. 21, no. 12, pp. 1509-1517, Dec. 2002.

[17] C.H. Yen and B.F. Wu, "Simple Error Detection Methods for Hardware Implementation of Advanced Encryption Standard," *IEEE Trans. Computers*, vol. 55, no. 6, pp. 720-731, June 2006.

[18] T.G. Malkin, F.X. Standaert, and M. Yung, "A Comparative Cost/Security Analysis of Fault Attack Countermeasures," Proc. Int'l Workshop Fault Diagnosis and Tolerance in Cryptography (FDTC '06), pp. 159-172, Oct. 2006.

[19] F. Dassance and A. Venelli, "Combined Attacks on the AES Key Schedule," in FDTC, Workshop on, pp. 63–71, 2012.

[20] F. Regazzoni, T. Eisenbarth, L. Breveglieri, P. Ienne, and I. Koren, "Can knowledge regarding the presence of countermeasures against fault attacks simplify power attacks on cryptographic devices," *in Proc. DFT*, pp. 202–210, 2008.

[21] V. Maingot and R. Leveugle, "Error detection code efficiency for secure chips," *in Proc. ICECS*, pp. 561–564., Dec. 2006.

[22] E. Oswald, S. Mangard, N. Pramstaller, and V. Rijmen, "A side-channel analysis resistant description of the AES S-box," in Proc. Fast Software Encryption (FSE). Lecture Notes in Computer Science, vol. 3557 (Springer, Berlin), pp. 413–423, 2005.

[23] National Institute of Standards and Technology (NIST) of U.S. Department of Commerce, "FIPS 197: Advanced Encryption Standard," Nov. 2001.

[24] F.-X. Standaert, "Introduction to Side-Channel Attacks," in Secure Integrated Circuits and Systems, pp. 27–44, Springer, 2009.

[25] K. Tiri, M. Akmal, I. Verbauwhede. "A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards," in Proc. Eur. Solid-State Circuits Conf. (ESSCIRC), Florence, Italy, 2002, pp. 403–406.

[26] M. Rajaram and J. Vijaya, "A Defense Mechanism for Differential Power Analysis Attack in AES", *Journal of Computer Science*, 2015, vol. 11, no. 2, pp: 291-296, 2015.

[27] J. L. Massey, "Guessing and entropy". *In Proceedings of the 1994 IEEE International Symposium on Information Theory*, page 204, 1994.

[28] C. O'Flynn and Zh. (D.) Chen, "Side Channel Power Analysis of an AES-256 Bootloader", IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 750 – 755, 2015.

[29] W. Hnath, J. Pettengill, "Differential Power Analysis Side-Channel Attacks in Cryptography," *Major Qualifying Project, Worcester Polytechnic Institute,* April 2010.

[30] K. Tiri and I. Verbauwhede, "A Logic level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation," *Proceedings of DATE*, pp. 246–251, 2004.

[31] R. Velegalati, J. Kaps, "DPA resistance for light-weight implementations of cryptographic algorithms on FPGAs*," International Conference on Field Programmable Logic and Applications (FPL)*, pp.385-390, 2009.

[32] S. Shah, R. Velegalati, J. Kaps, D. Hwang, “Investigation of DPA Resistance of Block RAMs in Cryptographic Implementations on FPGAs,” *International Conference on Reconfigurable Computing and FPGAs (ReConFig)*, pp.274-279, 2010.

[33] C. Hu, "IC reliability simulation", *IEEE J. Solid-State Circuits*, vol. 27, pp.241 -246, 1992.

[34] A. Narsale, M. C. Huang, “Variation-tolerant hierarchical voltage monitoring circuit for soft error detection”, *in Proc. ISQED, IEEE*, pp. 799-805, 2009.

[35] M. Zhang and N. R. Shanbhag, “Soft-error-rate analysis (SERA) methodology,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 10, pp. 2140–2155, Oct. 2006.

[36] J. Benedetto, et al. “Heavy ion-induced digital single-event transients in deep submicron processes,” *IEEE Trans. on Nuclear Science*, vol. 51, no. 6, pp. 3480–3485, Dec. 2004.

[37] N. George and J. Lach, “Characterization of logic masking and error propagation in combinational circuits and effects on system vulnerability,” in *Proc. IEEE/IFIP Dependable Systems & Networks*, pp. 323–334, 2011.

[38] M. Grosso, H. Guzman-Miranda, M. Aguirre, “Exploiting fault model correlations to accelerate SEU sensitivity assessment,” *IEEE Trans. On Industrial Information*, vol. 9, no. 1, pp. 142–148, Feb. 2013.

[39] T. Roche, V. Lomne, and K. Khalfallah, “Combined Fault and Side- Channel Attack on Protected Implementations of AES,” in Proc. Prouff, pp. 65–83, 2011.

[40] S. Sayil, A. B. Akkur, and III. N. Gaspard, “Single event crosstalk shielding for CMOS logic,” *Microelectronic Journal*, vol. 41, pp. 506–522, 2010.

[41] M. J. Gadlage, et al., “Alpha-particle and focused-Ion-Beam-Induced single-event transient measurements in a bulk 65-nm CMOS technology,” *IEEE Trans. on Nuclear Science*, vol. 58, no. 3, pp. 1093–-1097, Jun. 2011.

[42] G. I. Wirth, M. G. Vieira, E. H. Neto, and F. L. Kastensmidt, “Generation and propagation of single event transients in CMOS circuits,” in *Proc. Design & Diagnostics of Electronic Circuits and Systems*, pp. 196–201, 2006.

[43] M. G. Valderas, et al, “SET emulation under a quantized delay model,” in *Proc. 22nd IEEE Intl. Symp. Defect and Fault-Tolerance in VLSI Systems (DFTS)*, pp. 68–78, Sept. 2007.

[44] L. Entrena, et al., “SET emulation considering electrical masking effects,” *IEEE Trans. on Nuclear Science*, vol. 56, no. 4, pp. 2021–2025, Aug. 2009.

[45] G. Asadi and M. B. Tahoori, “An accurate SER estimation method based on propagation probability,” in *Proc. DATE’05*, pp. 306–307, 2005.

[46] T. Rejimon and S. Bhanja, “An accurate probabilistic model for error detection,” in *Proc. 18th Intl. Conf. on VLSI Design*, pp. 717–722, 2005.

[47] F. Wang, and V. D. Agrawal, “Soft error rates with inertial and logical masking,” in *Proc. 22nd Intl. Conf. on VLSI Design*, pp. 459–464, 2009.

[48] L. Chen, and M. Tahoori, “An efficient probability framework for error propagation and correlation estimation,” in *Proc. IOLTS’12*, pp. 170–175, 2012.

[49] M. J. Gadlage, et al., "Single event transient pulse widths in digital microcircuits," *IEEE Trans. on Nuclear Science*, vol. 51, no. 6, pp. 3285–3290, Dec. 2004.

[50] R. Ramanarayanan, et al., "Modeling soft errors at the device and logic levels for combinational circuits," *IEEE Trans. on Dependable and Secure Computing*, vol. 6, no. 3, pp. 202–216, Jul.-Sept. 2009.

[51] F. Amiel, K. Villegas, B. Feix, and L. Marcel, "Passive and active combined attacks: Combining fault attacks and side channel analysis," in FDTC, Workshop on, pp. 92–102, 2007.

[52] D. Alexandrescu, L. Anghel, and M. Nicolaidis, "Simulating single event transients in VDSM ICs for ground level radiation," *J. Electronic Testing: Theory and Applications*, vol. 20, pp. 413–421, 2004.

[53] N. Miskov-Zivanov, and D. Marculescu, "Multiple transient faults in combinational and sequential circuits: a systematic approach," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vo. 29, no. 10, pp. 1614–1627, Oct. 2010.

[54] S. S. Rathod, A. K. Saxena, S. Dasgupta, "Alpha-particle-induced effects in partially depleted silicon on insulator device: with and without body contact", IET Circuits Devices Systems, vol. 5, no. 1, pp. 52–58, 2011.

[55] C. Clavier, B. Feix, G. Gagnerot, and M. Roussellet, "Passive and active combined attacks on AES: Combining fault attacks and side channel analysis," in Proc. FDTC, Workshop on, pp. 10–19, 2010.

[56] G. Piret and J. J. Quisquater, "A Differential Fault Attack Technique against SPN Structures, with Application to the AES and Khazad," in CHES 2003, pp.77–88, LNCS 2779.

[57] H. Pahlevanzadeh, J. Dofe, Q. Yu, "Assessing CPA Resistance of AES with Different Fault Tolerance Mechanisms", ASPDAC, 2016.

[58] P. Luo, Y. Fei, L. Zhang, and A. A. Ding , "Side-channel power analysis of different protection schemes against fault attacks on AES," in Proc. ReConFigurable Computing and FPGAs (ReConFig), pp. 1–6, 2014.

[59] H. Liu, M. Cotter, S. Datta, et al., "Soft Error Performance Evaluation on Emerging Low Power Devices", IEEE Transactions on Device and Materials Reliability, no. 99, 2014.

[60] V. Maingot and R. Leveugle, "On the use of error correcting and detecting codes in secured circuits," in Proc. PRIME, pp. 245–248, 2007.

[61] V. S. Veeravalli, A. Steininger, U. Schmid, "Measuring SET pulse widths in logic gates using digital infrastructure", 15th International Symposium on Quality Electronic Design (ISQED), pp. 236–242, 2014.

[62] R.R. Rao, K. Chopra, D.T. Blaauw, et al., "Computing the Soft Error Rate of a Combinational Logic Circuit Using Parameterized Descriptors", IEEE Transactions on Computer-Aided Design of Integrated Circuits And Systems, vol. 26, no. 3, pp. 468–479, Mar. 2007.

[63] E. Prouff and M. Rivain, "Masking against side-channel attacks: A formal security proof," in Proc. EUROCRYPT, LNCS, vol. 7881, pp. 142–159. Springer, Heidelberg, 2013.

[64] J. Daemen and V. Rijmen, “Resistance against Implementation Attacks: A Comparative Study of the AES Proposals”, in Proc. Second Advanced Encryption Standard (AES) Candidate Conference, pp. 1–11, 1999.

[65] N. Kehl, W. Rosenstiel, “An Efficient SER Estimation Method for Combinational Circuits”, IEEE Transactions on Reliability, vol. 60, no. 4, pp. 742–747, Dec. 2011.

[66] F. Regazzoni et al., “Power Attacks Resistance of Cryptographic S-boxes with added Error Detection Circuits,” in Proc. DFT , pp. 508 – 516, 2007.

[67] H. Pahlevanzadeh and Q. Yu, “Systematic Analyses for Error Latching Probability of Single Event Transient”, in Proc. 15th International Symposium on Quality Electronic Design (ISQED), pp. 442–449, 2014.

[68] F. Wrobel, L. Dilillo, A. D. Touboul, et al., “Determining Realistic Parameters for the Double Exponential Law that Models Transient Current Pulses”, IEEE Transactions on Nuclear Science, no. 99, 2014.

[69] Y. Sun, C. Song, Y. Zhao, et al., “FAST: A framework of accurate SER-estimation at transistor-level for logic circuits” in Proc. 10th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT), pp. 1707–1709, 2010.

[70] R. Karri and X. Guo, "Invariance-based concurrent error detection for advanced encryption standard," in Proc. DAC, pp. 573-578, 2012.

[71] M. Fazeli, S. N. Ahmadian, S. G. Miremadi, et al., “Soft Error Rate Estimation of Digital Circuits in the Presence of Multiple Event Transients (METs),” in Proc. Design, Automation & Test in Europe Conference & Exhibition (DATE), pp.1-6, 2011.

[72] National Inst. Of Standards and Technology, "Federal Information Processing Standard Publication 197, the Advanced Encryption Standard (AES)," Nov. 2001.

[73] Barenghi, et al., “Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures,” in Proc. IEEE, vol. 100, no. 11, pp. 3056–3076, 2012.

[74] V. Lomne, T. Roche, and A. Thillard, “On the need of randomness in fault attack countermeasures—Application to AES,” in Proc. FDTC, pp. 85–94, Sep. 2012.

[75] Moradi, M. T. M. Shalmani, and M. Salmasizadeh, “A generalized method of differential fault attack against AES cryptosystem,” in Proc. CHES, pp. 91–100, 2006.

[76] M. Joye, P. Manet, and J. Rigaud, “Strengthening hardware AES implementations against fault attacks,” IET Info Security, vol. 1, no. 3, pp. 106-110, 2007.

[77] C. O’Flynn and Z. Chen., “Chipwhisperer: An opensource platform for hardware embedded security research”, in Proc. Prouff, pp. 243–260, 2014.

[78] L. Breveglieri, I. Koren, and P. Maistri, “Incorporating Error Detection and Online Reconfiguration into a Regular Architecture for the Advanced Encryption Standard,” in Proc. DFT, pp. 72-80, Oct. 2005.

[79] https://github.com/freecores/aes_decrypt_fpga/tree/master/rtl/ verilog

[80] G. Bertoni, L. Brevegelieri, I. Koren, P. Maistri, and V. Piuri, “Error Analysis and Detection Procedures for a Hardware Implementation of the Advanced Encryption Standard,” IEEE Trans. Computers, vol. 52, no. 4, pp. 492-505, Apr. 2003

[81] J. Mathew, et al., “On the design of different concurrent EDC schemes for s-box and gf(p),” in Proc. ISQED, pp. 211-218, 2010.

[82] M. Mozaffari-Kermani and A. Reyhani-Masoleh, “Concurrent Structure-Independent Fault Detection Schemes for the Advanced Encryption Standard,” IEEE Trans. Computers, vol. 59, no. 5, pp. 608–622, May 2010.

[83] Canright, D.,” A Very Compact Rijndael S-box,” Technical Report: NPS-MA-05-001, Naval Postgraduate School (2005).

[84] P. Maistri and R. Leveugle, “Double-Data-Rate Computation as a Countermeasure against Fault Analysis,” IEEE Transactions on Computers, vol. 57, no. 11, pp. 1528 – 1539, Nov 2008.

[85] K. Meritt, “Differential Power Analysis attacks on AES”, May 2012, Available at: http://people.rit.edu/kjm5923/DPA_attacks_on_AES.pdf.

[86] K. Smith Jr., “Methodologies for power analysis attacks on hardware implementations of AES,” *Master's thesis, Rochester Institute of Technology*, 2009.

[87] A. A. Kamal and A. M. Youssef, “An area-optimized implementation for AES with hybrid countermeasures against power analysis,” *In Proc. Int. Symp. Signals Circuits Syst. (ISSCS’09)*, pp. 1–4, 2009.

[88] S. Kan, M. Ottavi, and J. Dworak. "Enhancing embedded SRAM security and error tolerance with hardware CRC and obfuscation." IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS), 2015.

[89] . Guillaume Duc, Ronan Keryell, "CryptoPage: An Efficient Secure Architecture with Memory Encryption, Integrity and Information Leakage Protection", Annual, Computer Security Applications Conference (ACSAC,), pp. 483-492, 2006

[90] K. R. Malakapalli, and KP. Tsang. "Mass storage error correction and detection system, method and article of manufacture." U.S. Patent No. 6,467,060. 15 Oct. 2002.

[91] H. Okano, "Information processing system using error-correcting codes and cryptography." U.S. Patent No. 5,504,818. 2 Apr. 1996.

[92] W. H. Radke, S. Swaminathan, and B. L. Keays. "Non-systematic coded error correction." U.S. Patent No. 7,444,579. 28 Oct. 2008.

[93] F. R El-Din, R. M. El-Hassani, and S. H. El-Ramly. "A novel high-speed systematic encoder for long binary cyclic codes." Communications Letters, IEEE 17.5 pp. 984-987 (2013):

[94] “Memory Corruption Attacks The (almost) Complete History,” http://https://media.blackhat.com/bh-us-10/whitepapers/Meer/BlackHatUSA-2010-Meer-History-of-Memory-Corruption-Attacks-wp.pdf, 2010, [Online; accessed 23-Apr-2015].

[95] M. Tehranipoor and F. Koushanfar, “A survey of hardware trojan taxonomy and detection,” Design Test, IEEE, vol. PP, no. 99, pp. 1–1, 2013.

[96] S. Lin and D.J. Costello, “Error control coding,” (2nd Edition), Prentice-Hall, Inc. Upper Saddle River, NJ, USA.

[97] K. Schramm and C. Paar, “Higher Order Masking of the AES,” In D. Pointcheval, editor, Topics in Cryptology-CT-RSA 2006, volume 3860 of Lecture Notes in Computer Science, pp. 208-225. Springer, 2006.

[98] Avirneni, Naga Durga Prasad; Somani, Arun K. “Countering power analysis attacks using reliable and aggressive designs”, IEEE Transactions on Computers, vol. 99, pp. 1. 2013.

[99] C. O’Flynn and Z. Chen, “Chipwhisperer: An opensource platform for hardware embedded security research,” in Proc. Prouff, pp. 243–260, 2014

[100] Colbourn, Charles J., ed. CRC handbook of combinatorial designs. CRC press, 2010.

[101] Normoyle, Kevin B., and Robert G. Hathaway. "Subsystem and method for encoding 64-bit data nibble error correct and cyclic-redundancy code (CRC) address error detect for use in a 76-bit memory module." U.S. Patent No. 8,099,651. 17 Jan. 2012.

[102] Parrilla, Luis, et al. "Hardware Activation by Means of PUFs and Elliptic Curve Cryptography in Field-Programmable Devices." Electronics 5.1 (2016): 5.

[103] Capiluppi, Andrea, Paolo Falcarin, and Cornelia Boldyreff. "Code defactoring: Evaluating the effectiveness of java obfuscations." Reverse Engineering (WCRE), 2012 19th Working Conference on. IEEE, 2012.

[104] Wang, Huaijun, et al. "Software Attack Modeling and Its Application." High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC), 2013 IEEE 10th International Conference on. IEEE, 2013.

[105] Udupa, Sharath K., Saumya K. Debray, and Matias Madou. "Deobfuscation: Reverse engineering obfuscated code." Reverse Engineering, 12th Working Conference on. IEEE, 2005.