Concurrent Fault Detection for Secure QDI Asynchronous Circuits

Konrad J. Kulikowski, Mark G. Karpovsky, Alexander Taubin, Zhen Wang
Boston University, Reliable Computing Laboratory
8 Saint Mary's Street, Boston, MA 02215, USA
{konkul,markkar,taubin,lark}@bu.edu

Adrian Kulikowski
Deloitte & Touche, LLP
2 World Financial Center, New York, NY 10281
akulikowski@deloitte.com

Abstract

Asynchronous micropipelined designs are self-checking against permanent stuck-at faults making the circuits highly resistant against invasive fault attacks. A single transient fault, however, can result in a data token insertion or deletion which can cause a continuous stream of erroneous outputs that can be exploited by a malicious attacker. We propose a system level error detection method targeted for detection of faults that cause data token insertions and deletions. The method is based on minimum distance robust error detecting codes and exploits the repeating errors in linear networks to improve the error detection performance.

1. Introduction

Increased parameter variation, difficulties of predicting timing, high soft error rates, and increased noise are just some of the challenges facing nanoscale electronics. Asynchronous circuits have been shown to solve many of these problems and are a likely candidate to replace clocked implementations for future digital nanotechnologies [9]. Asynchronous circuits have also been shown to have many inherent properties beneficial for secure circuit applications. Clockless designs have been shown to have advantages in providing protection against power, EMI, timing, and fault attacks [6]. Based on these trends in secure and nano circuits one can expect future secure digital devices to be based on asynchronous designs.

In this paper we investigate effects of faults and errors of secure asynchronous quasi delay insensitive (QDI) designs and propose a protection method against fault attacks that exploits the unique fault characteristics of the designs. Although we concentrate our analysis on security applications, the fault models and methods considered are useful in general designs as well.

The unique behavior of fine-grained QDI asynchronous circuits in the presence of transient faults requires special consideration and can be exploited to improve the error detection. We introduce *minimum distance robust codes* which combine the robust and classical minimum distance properties of codes. These codes provide for better detection than the traditional system level methods for detection of transient faults in fine grained QDI asynchronous circuits especially when the faults result in a deletion or a creation of a data token. We apply and analyze our method for the protection of the linear portion of the Advanced Encryption Standard (AES).

2. Faults in QDI Circuits

The effects of faults on QDI circuits has been investigated in [7] [8]. A fault within a QDI asynchronous cell can have one of the following effects

1. deadlock
2. invalid data token ('11')
3. data modification (flipping a value of a data token)
4. data generation (creation of a data token)
5. data deletion (deletion of a data token).

A major benefit of QDI asynchronous circuits is that almost all stuck-at-faults create a deadlock in the system [2] which makes the circuits almost completely self-testing against permanent stuck-at faults. This behavior is especially beneficial in secure fault attack resistant systems since it makes it much more difficult to use invasive techniques to perform a fault attack.

Asynchronous designs also offer good resistance against many transient disruptions. The dual rail encoding of data makes it more unlikely that data modifications due to natural events will occur since a flip in a logical value requires the modification of two signals. The encoding redundancy of the dual-rail data can be used for fine grained checking of data validity for detection of many transient disruptions that result in an invalid data token value (i.e. '11').

Asynchronous circuits, however, are vulnerable to the effects of transient faults due to the final two manifestations. A transient fault which disrupts the handshaking between gates can result in generation or deletion of a data token. A single transient fault which creates or deletes a data token can result in a large number of erroneous outputs.

To illustrate the negative effects of token insertions or deletions consider the circuit in Figure 1 which shows block level data flow for a asynchronous QDI micropipeline. A fault which generates a new data token (marked with X in Figure 1) can result in a misalignment of all the data within the pipeline as well as all future data inputs. The new data token can erroneously take the place of a data token in the data streams thereby offsetting all of the tokens in all of the following streams (see Figure 1). When this occurs the data tokens in the data steams will be mis-synchronized and the final output can be erroneous for each misaligned data stream. The fault, although only transient, will corrupt future data messages as though they all had the same fault. A similar effect can be observed when a data token is deleted due to a fault.

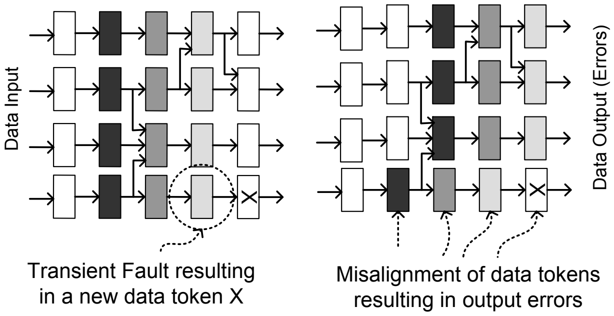
Data Input
Data Output (Errors)
Transient Fault resulting in a new data token X
Misalignment of data tokens resulting in output errors

Figure 1. Example of token insertion

By injecting faults which generate or delete data tokens into an asynchronous implementation an attacker can obtain many erroneous outputs corresponding to that fault without having to reapply the fault or cause permanent damage to the circuit. The ability to obtain multiple erroneous outputs corresponding to a single fault location simplifies the attack as it makes it easier to determine the location of the fault. Other than a possible eventual pipeline stall for some implementations with algorithmic loops, QDI asynchronous circuits have no natural method of detecting such behavior.

The observation that many faults within a QDI asynchronous circuits do not lead to a deadlock is important for traditional reliability considerations and has motivated several specialized circuit-level protection methods [3] [7] [10] [11].

A major disadvantage of the proposed circuit level techniques is that they are designed to stop the propagation of the errors and typically have no methods for detection or they assume only single event transients within the circuit. Detection of faults for secure circuits is crucial since the frequency and number of faults observed can be used to recognize if a device is under an attack. The ability to recognize a fault attack increases the device security since it allows the device to take reactive measures such as clearing sensitive information from memory to prevent possible data exposure. Masking or filtering of transient faults by circuit duplication reduces the possibility of detecting and reacting to an attack. Likewise, assumptions of single transient faults used for natural sources are cannot be guranteed for an active attacker.

We propose a system level protection method which does not mask faults and can detect data token deletions, creation, and modifications. The method uses a concurrent error detection architectures based on minimum distance robust error-detecting codes. The codes are especially suited for detecting token deletions and insertions due to faults in linear networks as the probability of detection for these codes increases as more erroneous outputs are observed. Likewise, the probability of detection for these codes is generally independent of the error multiplicity and provides for a guaranteed level of detection regardless of the error type. These robust codes offer several advantages over classical linear error-detecting codes such as Hamming or parity for the error model considered. The classical methods have a large number of undetectable errors and they do not exploit the repeating nature of the errors associated with token insertions and deletions.

The definition of these codes and a simple construction is presented in the next section followed by their application to AES.

3. Robust Codes

We start with a summary of the relevant definitions and construction from [5] and propose new constructions of *minimum distance robust* codes. These codes have a minimum distance which guarantees the detection of errors of a low multiplicity while preserving robust error detection properties for all other errors.

We define the error-masking probability $Q(e)$, for an error e and a code C as

$$Q(e) = \frac{|\{w|w \in C, w+e \in C\}|}{|C|} \quad (1)$$

where $+$ is addition in the respective field.

A robust code is a code for which there are no undetectable errors. Equivalently, code is robust if there is at least one codeword that does not result in the masking of the error.

Definition 3.1 *The code C is* ***robust*** *with respect to its error-masking probability iff $Q(e) < 1$ for all nonzero errors.*

For a given dimension and redundancy of a robust code the aim is to minimize the maxima of $Q(e)$ or equivalently the number of messages for which an error will be masked over all nonzero errors . To help in the discussion we define *R-robustness* of a code.

Definition 3.2 *A robust code C where*

$$R = \max_{e \neq 0, \in GF(q^n)} Q(e)|C| \quad (2)$$

is called ***R-robust****.*

A systematic robust code can be defined by the nonlinearity of its encoding function. The nonlinearity of the function can be measured by using derivatives $D_a f(x) = f(x+a) - f(x)$. The nonlinearity measure can be defined by (from [1])

$$P_f = \max_{0 \neq a \in GF(2^k)} \max_{b \in GF(2^r)} Pr(D_a f(x) = b) \quad (3)$$

where $Pr(E)$ denotes the probability of occurrence of event E. The smaller the value of P_f, the higher the corresponding nonlinearity of f.

Theorem 3.1 ***([5])*** *Let f be a function with nonlinearity P_f that maps $GF(2^k)$ to $GF(2^r)$ where $k \geq r$, the set of vectors resulting from the concatenation of a, b : $\{(a, b = f(a))\}$ where $a \in GF(2^k)$ and $b \in GF(2^r)$ forms a $(2^k P_f)$-robust systematic error-detecting code.*

The general constructions of robust codes do not consider a minimum distance as in classical error-detecting codes. However, the construction of systematic robust codes based on nonlinear functions can be applied to classical linear minimum-distance codes to produce *minimum-distance robust codes*.

Definition 3.3 *Let $||e||$ denote the multiplicity of an error e. A R-robust code where the $Q(e) = 0$ for all errors where $||e|| < d$ is a* ***d-minimum distance R-robust code****. A d-minimum distance R-robust code of dimension n with M codewords is denoted by a quadruple (n, M, d, R).*

Minimum distance robust codes can be constructed by appending a nonlinear signature to any systematic minimum distance code.

Theorem 3.2 *Let V be a systematic $(n, 2^k, d)$ code and let $f : GF(2^k) \to GF(2^r)$ be nonlinear function with nonlinearity P_f. The code*

$$C = \{(x, Px, f(x))|(x, Px) \in V\}, \quad (4)$$

where P is the encoding matrix for code V, is a $(n + r, 2^k, d, R \leq 2^k P_f)$ minimum distance robust code.

Proof Appending extra nonlinear bits does not change the minimum distance of the code. Any error which will affect only the redundant bits of V will clearly be immediately detected. Any other error will be detected by the robust code. The robustness R of the code follows from Theorem 3.1.

For the designs in this paper we use robust codes based on the non-repetitive quadratic form as the nonlinear function for the robust code:

$$f(x = (x_0, x_1, ...x_k)) = x_0 x_1 + x_2 x_3 + ... + x_{k-1} x_k \quad (5)$$

where k is odd, $x_i \in GF(2)$, and all operations are over $GF(2)$.

The non-repetitive quadratic function which maps elements from $GF(q^k)$ to $GF(2)$ is a perfect nonlinear function and results in optimum systematic robust codes. Using the function f we define the following two codes.

Construction 3.1

$$C_1 = \{(x, f(x))|x \in GF(2^k)\} \quad (6)$$

where k is odd and f is the function from (5) is a (2^{k-1})-robust code with dimension $n = k + 1$.

Construction 3.2

$$C_2 = \{(x, p(x), f(x))|x \in GF(2^k)\} \quad (7)$$

where k is odd, f is the function from (5), and $p(x)$ is the parity of x is a $(k+2, 2^k, 2, 2^{k-1})$ minimum distance robust code.

The robust and minimum distance robust codes have several benefits which motivated their application for detection of token insertions and deletions in secure circuits. First as described in more detail in [4] and [5], robust codes have uniform error detection. Their error masking probability is bounded by $R/|C|$ regardless of error multiplicity or any subset of errors that is considered. This eliminates weak areas of protection. Even if an attacker injects faults that result in errors of high multiplicities, the protection guarantees a minimum level of error detection. That is not the case for protection based on linear codes. Error detection properties for errors of multiplicity greater than the minimum distance for linear codes can be unpredictable and very low.

Secondly, the detection of errors for these codes is data dependent. Each error is masked only for a R subset of possible codewords. When an error distorts several different codewords the probability of detecting the error increases. That is, the chances that at least one of the codewords will not be in the set which masks the error decreases exponentially for a uniform distribution of codewords. This property increases the probability of detecting token insertions and deletions. In linear circuits token insertions and deletions result in a continuous stream of repeating errors.

In the next section we apply these codes to the protection of a round of AES to demonstrate their benefits and usefulness for detection of faults in micropipelined asynchronous circuits.

4. Application to AES

4.1 Implementation

A round of AES can be divided into two types of operations, linear and nonlinear. The field inversion is the only nonlinear transformation. All other operations are linear and can be implemented with XOR gates only. On the round level the 128-bit datapath can be split into four independent and identical 32-bit data streams. We will therefore describe the design with respect to one 32-bit stream. In [4] an efficient method for providing robustness for the nonlinear portion, the field inversion, was presented. We extend the completely robust protection to the linear MixColumn and affine transformations of a round of AES.

For all analysis and experiments we used the linear portion of a typical AES round which consists of one MixColumn transformation and four affine transforms. The linear circuit and all circuits used to add protection were synthesized from an HDL specification using Synopsys Design Compiler. The synthesized linear network which has a 32-bit input and a 32-bit output requires 216 two-input XOR gates.

The error characteristics of the linear circuit due to single token insertions and deletions are depicted in Figure 2. The figure shows the fraction of single token insertions or deletion within the circuit that will result in a given multiplicity of errors at the output of the network. As the histogram shows many of the single token insertions will result in a single bit error or affect an odd number of output bits. This property has previously motivated the use the parity code for protection. However, in addition to the errors which can be detected by parity there are many errors which are undetectable by the parity method. Even weight errors account for 27% of the possible manifestations of a token insertion within the circuit.

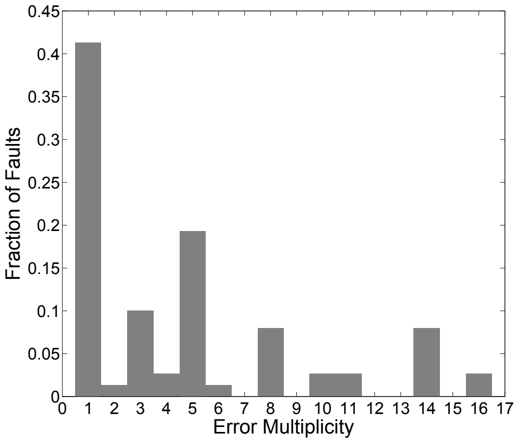

Figure 2. Fraction of single token insertions causing an error of given multiplicity

To evaluate the benefits of using robust and minimum distance robust codes we injected faults into designs with three different protection methods: linear parity, robust parity, and a minimum distance robust codes. In all the three cases the linear network remained the same only the predictor circuit and the error detecting network were changed.

For each protection method, to the linear circuit a redundant predictor (P) is added such that the output of the predictor and the output of the linear network of AES result in a desired codeword. An additional circuit, the error detecting network (EDN), is used to verify the relationship between the output of the linear network and the predicted signature. The general architecture of the protection is shown in Figure 3.

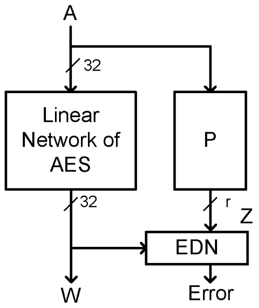

Figure 3. Architecture for concurrent error detection

	predictor	EDN	overhead (%)
linear parity	31	32	30%
robust parity	185	32	100%
min. dist. robust	196	64	120%

Table 1. Gate count and hardware overhead for designs

For the implementation protected by linear parity the one-bit output of the predictor Z is equal to the parity of the error free output W. For the implementation protected by the robust parity, the one-bit output of the predictor is equal to the nonrepetitive quadratic function of W. Finally, for the implementation protected by the minimum distance robust codes the predictor has a two-bit output. One bit is equal to the parity and the other is equal to the non-repetitive quadratic function of the error-free output of the linear network of AES.

The overheads of each of the implementations are summarized in Table 1. The table lists the number of two-input gates required for each implementation and the overhead compared to the unprotected implementation. The linear parity requires very little overhead due to the parity preserving nature of the linear operations. That is, the parity of the inputs is equal to the parity of the outputs which results in a compact parity predictor and a 30% gate count overhead. The robust parity implementation requires the prediction of a nonlinear function of the output and results in a larger overhead. Despite the larger overall overhead, the size of the error detecting network for robust parity is the same size as for the implementation based on linear parity codes. The final implementation which combines the linear and robust parity into one implementation requires slightly more hardware in the predictor and 32 more gates in the EDN compared to robust parity.

We next analyze the benefits of the robust implementation of the AES subcircuit.

4.2 Error Detection Analysis and Simulation

The three different architectures (linear parity, robust parity, and minimum distance robust) were simulated to compare their effectiveness and general protection characteristics in detecting token insertions and deletions. Transient faults which cause data token misalignment were injected and random inputs were applied for each fault. The misaligned outputs as well as the error signal of each architecture was recorded.

The results of the simulations where injection of faults cause only single token insertions and deletions are depicted in Figure 4. The Figure shows the probability of detecting a token insertion or deletion as a function of the number of misaligned outputs that were observed. The curves in the Figure represent the expected probability that a fault which results in a token insertion or deletion will be detected by the circuit. A token insertion or deletion is considered detected if it triggers the error signal at least once for the given number of observed misaligned streams.

A misalignment of data does not always result in errors at the output of the circuit. When data tokens from adjacent data streams have equal values the misalignment of data does not result in a distortion. Not all outputs resulting from the token generations or deletion can be used to detect the fault. As more outputs are observed the probability that the misaligned data tokens do not match and will result in a manifestation of the fault increases.

For each of the architectures the probability of detection grows as more misaligned outputs are observed. For the linear architecture the increase as a function of buffered outputs is due only to the increased probability that the misalignment of data will result in incorrect data. As shown in the histogram in Figure 2, 73% of single token insertions and deletions cause errors of an odd multiplicity. After observation of a single misalignment output, the probability that the fault will be detected can be estimated by taking the product of manifestation and the probability of an error of an odd multiplicity. As more misaligned tokens are observed the probability of a faulty output increases. For the architecture based on linear parity the probability of detection asymptotically approaches the value of 73% which corresponds to the percentage of token insertions which result in an odd error. Regardless of how long the data is observed, the linear architecture will miss 27% of single token insertions or deletions.

For the architecture based on the robust parity code the probability of detection as a function of observed streams increases faster than for the linear architecture. The increase of the probability of detection is not only due to the increased probability of manifestation but also due the robustness of the code. Since the subcircuit of AES is linear, single token insertions and deletions result in a repeated errors whenever they are manifested. When a manifestation corresponding to the token generations or deletions occurs, it always results in the same error pattern at the output of the linear network. Due to the robustness of the codes, the probability of detecting the error, and hence the fault causing the initial token insertion or deletion, increases as more outputs are distorted by the same error. By definition of robust codes, the probability of masking for each error depends on the data. As more codewords are affected by the same error the probability of detection increases.

When all of the outputs of the circuit are possible (as is the case of the linear subcircuit since AES is a one to one function) the probability of detection for the robust codes approaches one as more erroneous streams are observed.

With a single additional bit, all of the token insertions and deletions are detectable. The probability of detection for the robust parity is already greater than that for the linear parity after four streams.

The probability of detection for the robust parity when fewer than four streams are observed is, however, significantly lower than for the linear parity. This can be attributed to the error characteristics of the circuit which favor a code such as parity. The robust parity code does not assume any given error distribution and any error pattern is masked with a probability of at most 50%.

Due to the distribution of errors associated with the linear subcircuit of AES (see Figure 2) the architecture based on a minimum distance robust provides the best performance. The architecture combines the linear parity and robust parity methods to ensure eventual detection of all token deletions and insertions but also provide lower detection latency for the nonuniform error profile of the network. As shown in Figure 4, the probability of detection of single token insertions and deletions for the minimum distance robust codes approaches one as the number of observed streams increases.

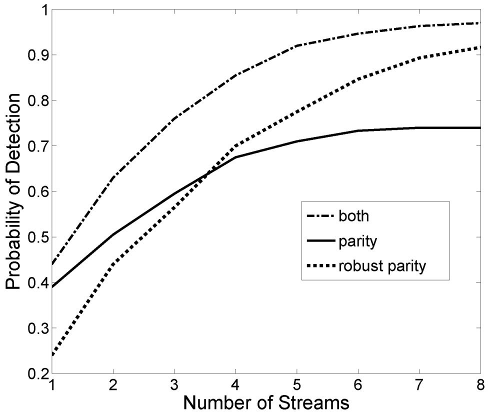

Figure 4. Probability of detecting token insertions and deletions versus number of buffered streams

5. Conclusions

The hardware overhead of the method based on robust error detecting codes is on the order of the previously proposed circuit level methods but has several advantages. The method

1. detects all possible token insertions or deletions,
2. detects data token modifications and manifestations of invalid data tokens,
3. does not mask or prevent propagation of soft errors allowing detection of an attack.

The use of minimum distance robust codes allows guaranteed detection of most probable errors while providing robust detection for all remaining errors. For the robust architectures, the probability of error detection can be configured by adjusting the number of buffered message streams. The method is especially useful for use in high level HDL specifications where the exact circuit topology depends on the synthesis engine and the final library or technology to which the circuit is mapped. Although the method was analyzed specifically with respect to secure AES, the method can be used in any general asynchronous design and can be useful in asynchronous nanocircuits which are susceptible to high error rates.

References

[1] C. Carlet and C. Ding. Highly Nonlinear Mappings. *J. Complex.*, 20(2-3):205–244, 2004.

[2] I. David, R. Ginosar, and M. Yoeli. Self-timed is self-checking. *J. Electron. Test.*, 6(2):219–228, 1995.

[3] W. Jang and A. Martin. SEU-Tolerant QDI Circuits. In *ASYNC '05*, pages 156–165, 2005.

[4] M. Karpovsky, K. J. Kulikowski, and A. Taubin. Differential Fault Analysis Attack Resistant Architectures for the Advanced Encryption Standard. In *CARDIS*, 2004.

[5] K. J. Kulikowski, M. G. Karpovsky, and A. Taubin. Robust Codes and Robust, Fault-Tolerant Architectures of the Advanced Encryption Standard. *J. Syst. Archit.*, 53(2-3):139–149, 2007.

[6] K. J. Kulikowski, A. Smirnov, and A. Taubin. Automated Design of Cryptographic Devices Resistant to Multiple Side-Channel Attacks. In *CHES '06*, pages 399–413, 2006.

[7] C. LaFrieda and R. Manohar. Fault Detection and Isolation Techniques for Quasi Delay-Insensitive Circuits. In *DSN '04*, page 41, 2004.

[8] R. Leveugle, A. Ammari, V. Maingot, E. Teyssou, P. Moitrel, C. Mourtel, N. Feyt, J.-B. Rigaud, and A. Tria. Experimental Evaluation of Protections against Laser-induced Faults and Consequences on Fault Modeling. In *DATE '07*, pages 1587–1592, 2007.

[9] A. J. Martin and P. Prakash. Asynchronous Nanoelectronics: Preliminary Investigation. In *ASYNC '08*, 2008.

[10] Y. Monnet, M. Renaudin, and R. Leveugle. Designing Resistant Circuits against Malicious Faults Injection Using Asynchronous Logic. *IEEE Transactions on Computers*, 55(9):1104–1115, 2006.

[11] S. Peng and R. Manohar. Efficient Failure Detection in Pipelined Asynchronous Circuits. *DFT '05*, pages 484–493, 2005.