Circuit Level Defences Against Fault Attacks in Pipelined NCL Circuits

Qingyu Ou, Fang Luo, Shilei Li, and Lu Chen

***Abstract*— As a type of side-channel attack, fault attacks on the hardware implementation of cryptographic algorithms has been widely studied, and has been successfully applied to a variety of ciphers. So far, many hardware countermeasures against fault attacks have been proposed. However, most of them are tailored to a specific cryptographic algorithm and cannot provide a comprehensive level of protection against all possible faults. In this paper, we present a novel, secure and generic framework based on the robustness of null convention logic pipelines and dual-rail encoded systems, which is resistant to fault attacks at the circuit level by implementing technologies such as rail synchronization, maximum delay matching, error-detecting monitors, and self-feedback mechanisms. Both the theoretical analysis and simulation results show that all faults at the output of computational blocks can be fully detected and corrected. The new framework gives us the ability to detect and correct the faults induced in the register.**

***Index Terms*— Fault attack, fault-tolerant, null convention logic (NCL).**

I. Introduction

FAULT ATTACK on the hardware implementations of cryptographic algorithms was firstly studied by Boneh *et al.* [1], and since then, it has been successfully applied to a variety of ciphers [2]–[8]. To resist fault attacks, several methods and architectures have been proposed [9]–[12]. However, because most of the current integrated circuits are synchronous, and their activities are controlled by a global clock, the behavior and sensitivity of synchronous circuits may make the ciphers exposed to fault attacks [13], [14].

Clockless circuits represent a class of circuits which are not controlled by a global clock or data, but by those logical relationships. Because of their specific architecture, clockless circuits have more different behaviors than synchronous circuits in the presence of faults. Monnet *et al.* [15] proposed a reference asynchronous architecture for DES, and developed several hardening techniques for quasi-delay insensitive (QDI) circuits with the rail synchronization and token game [15]. By minimizing distance robust error-detecting codes, Kulikowski *et al.* [16] proposed a system-level detection of faults that cause data token insertions and deletions in QDI asynchronous circuits. Peng *et al.* developed a detection of concurrent failure method for pipelined QDI circuits by achieving fail-stop once permanent or transient errors occur [17]. However, these approaches are not secure against the fault sensitivity analysis (FSA) [18] due to the dependence of the sensitivity data on the critical conditions. Kuang *et al.* [19] proposed a modified null convention logic (NCL) architecture with high single event upset (SEU) tolerance to eliminate all multiple SEUs owing to particle strikes in a steady state of the computational block, and then a technique against particle strikes by using Schmitt trigger circuit and resizing the feedback transistor [20] was proposed. Nevertheless, these approaches are not resistant against faults that are generated in the register. Besides, the logic design of discarding the wrong data token is not presented.

NCL [21] is a type of asynchronous logic, and it is a delay-insensitive (DI) logic. The logically determined property of NCL circuits makes them naturally against some categories of faults and fault sensitivity analysis. Thus, NCL circuits are an attractive design option for fault-tolerant systems.

In this paper, the fault propagation characteristics of NCL pipeline are analyzed, and a new framework called the synchronization self-feedback random tower (SFRT) structure for high fault-tolerance is proposed. To enhance the properties of resistance to fault attacks, the rail synchronization technology and monitoring illegal dual-rail codes are adopted, which enables us to improve the fault tolerance of circuit by transforming the faults into the illegal dual-rail codes. In addition, the SFRT register which provides the ability to detect the soft error with a high spatial and temporal resolution is implemented. Both theoretical analysis and simulation results show that the new framework proposal brings great fault-tolerant improvements.

The rest of this paper is organized as follows. Section II describes the generation and propagation of faults in NCL pipelines. Section III presents the principle of the SSFRT structure. The simulation results are discussed in Section IV. Section V concludes this paper.

II. Overview of NCL Pipeline

A. Architecture

NCL is a QDI asynchronous paradigm, because wires-connecting components have to adhere to the isochronic

Manuscript received October 4, 2013; revised July 12, 2014; accepted August 24, 2014. Date of publication September 19, 2014; date of current version August 21, 2015. This work was supported in part by the National Natural Science Foundation of China under Grants 6110042 and 61202338; in part by the Natural Science Foundation of Hubei Province under Grant 2013CFB441; and in part by the Foundation of Science and Technology on Information Assurance Laboratory under Grant KJ-13-106.

The authors are with the Department of Information Security, Naval University of Engineering, Wuhan 430033, China (e-mail: ouqingyv@163.com; lf_0215@sina.com; listone@163.com; ieucl@163.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2014.2354531

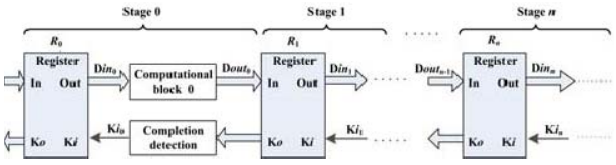


Fig. 1. NCL pipeline architecture.

fork assumption. As shown in Fig. 1, a typical NCL pipeline architecture consists of computational blocks, registers, and completion detection circuits.

The computational block consists of various threshold gates. The threshold gate can be expressed as THmn gate where n is the number of inputs, m is the threshold, and $1 \leqslant m \leqslant n$. The following behavior constraints on the computational block must be satisfied for QDI [20]: 1) its outputs may not transition from all NULL to a complete set of DATA until the input values are completely DATA; and 2) its outputs may not transition from a complete set of DATA to all NULL values until the input values are completely NULL. These constraints are equivalent to Seitz's weak conditions [22].

Registers and completion detection circuitry are required to coordinate the adjacent computational blocks to ensure the correct data communications [20] Two adjacent registers interact with each other through their request and acknowledge signals K_i and K_o, respectively. The current DATA wavefronts are prevented from overwriting the previous DATA wavefronts by ensuring that the two DATA wavefronts are always separated by a NULL wavefront. When a register (e.g., R_1) detects a complete set of DATA at the output of computational block, it will inform the previous register (e.g., R_0) that the current computation has been done, and then a NULL wavefront is allowed to flow into the computational block by setting Ki_0 low.

B. Hysteresis

In the NCL pipeline, threshold gates are designed with hysteresis state-holding capability, such that after the output is asserted, all inputs must be deasserted before the output is deasserted.

Hysteresis ensures a complete transition of inputs back to NULL before asserting the output associated with the next wavefront of the input data [21], [23].

III. Faults Propagation in NCL Pipelines

Because of the hysteresis of threshold gates, the delay faults will be filtered by NCL pipeline, and therefore only the transient/steady faults, caused by variations of the power supply voltage, temperature, radiation, intense light or EM disturbances, particle strikes, and so forth, will have influence on the behavior of NCL pipelines.

The propagation of a transient reset fault (1-0-1) or steady reset fault (1-0) in NCL pipelines, may lead to an earlier completion of the current computation without affecting the circuit logic function. In contrast, the propagation of a transient set fault (0-1-0) or steady set fault (0-1) may make the error-free signal high and result in a wrong DATA, which may be delivered to the next stage. Table I shows the behavior profile for the NCL pipeline.

TABLE I
Behavior Profile for the NCL Pipeline

The output rails of the current stage		Request Signal	The input rails of the next state		Description
$Dout_0^0$	$Dout_0^1$	Ki_1	Din_1^0	Din_1^1	
X	X	0	0	0	The faults have no influence on the next stage when the signal Ki_1 is low
1	0	1	1	0	The stable wavefront of correct results reaches the input of the next stage when the signal Ki_1 is high
0	1	1	0	1	
▲	X	1	▲	X	The faults is delivered to the next stage when the signal Ki_1 is high
X	▲	1	X	▲	

X = Don't Care
▲ = A transient/steady set fault

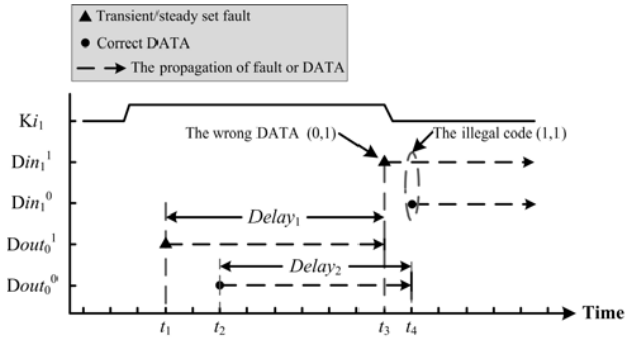


Fig. 2. Fault takes to propagate prior to the correct DATA.

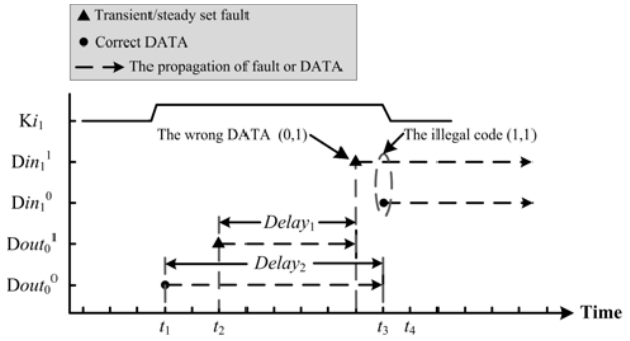


Fig. 3. Fault takes to propagate preceded by the correct DATA.

If a transient/steady set fault appears on the output rails $Dout_0$ ($Dout_0^0$, $Dout_0^1$) before the signal Ki_1 is set to high, the fault will not be delivered to the next stage, and thus the next stage cannot receive any new data at this time. In contrast, if a transient/steady set fault appears on the output rails $Dout_0$ ($Dout_0^0$, $Dout_0^1$) after the signal Ki_1 is set to high, the fault will be delivered to the next stage. However, the influence of NCL pipelines will be different, depending on the fault propagation characteristics.

For simplicity, it is assumed that a transient/steady set fault appears on the output rail $Dout_0^1$ whose error-free signal is low and the signal Ki_1 is set to high. As shown in Figs. 2 and 3, respectively, once the fault arrives at the register in the current

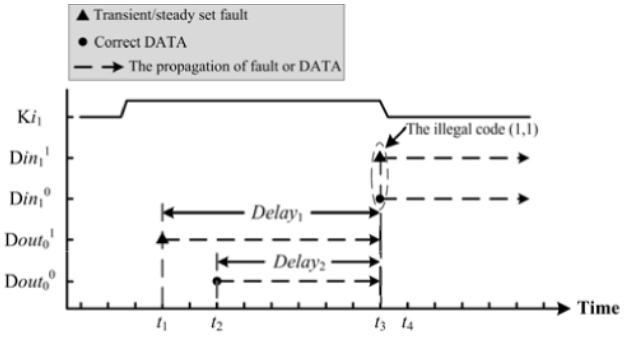


Fig. 4. Fault and the correct DATA arrive at the register at the same time.

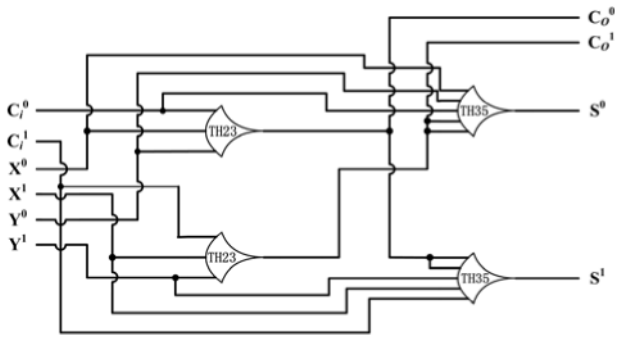


Fig. 5. Optimized NCL fulladder.

stage, prior to the correct DATA on the output rail Dout_0^0. The wrong DATA (0, 1) will be delivered to the next stage prior to the illegal dual-rail code (1, 1). The same situation happens when a transient/steady set fault appears on the output rail Dout_0^1 whose error-free signal is low. More formally, the preconditions that the wrong DATA will take to propagate through the pipeline can be defined as follows:

$$\mathrm{D}delay_T - \mathrm{D}delay_D < t_D - t_T, \quad \text{if } t_D > t_T \tag{1}$$

$$t_T - t_D < \mathrm{D}delay_D - \mathrm{D}delay_T, \quad \text{if } t_D \le t_T \tag{2}$$

where t_T and t_D are the duration of the transient/steady set fault and the correct DATA, respectively, and D$delay_T$ and D$delay_D$ are the propagation delay of the transient/steady set fault and the correct DATA, respectively, before they arrive at the register in the next stage.

However, a special case that the transient/steady set fault and the correct DATA arrive at the register in the next stage at the same time, should be noted. In this case, the illegal dual-rail code (1, 1) will be exhibited on the rails Din_1 (Din_1^0, Din_1^1) as shown in Fig. 4.

More formally, for the transient set fault, the precondition that the illegal dual-rail code (1, 1) will be exhibited immediately in the next stage, can be defined as follows:

$$|\mathrm{D}delay_T - \mathrm{D}delay_D| = |t_D - t_T|. \tag{3}$$

For the steady set fault, because of the hold characteristics of the fault signal, the condition that the illegal dual-rail code will be exhibited immediately in the current stage, can be defined as follows:

$$\mathrm{D}delay_T > t_D - t_T \quad \text{if } t_D > t_T \tag{4}$$

$$t_T - t_D < \mathrm{D}elay_D \quad \text{if } t_D \le t_T. \tag{5}$$

If the transient/steady set fault occurs on the rail whose error-free signal is high, it will only lead to an earlier transition from NULL to DATA without affecting the circuit logic function.

IV. Fault-Attack Resistant Architectures

As a result of the logical determination of NCL circuits, wavefronts will propagate without any requirements of significant timing assumptions. It may make each dual-rail data flow out from the output of the computational block at different times. Furthermore, even if the input DATA is synchronized, there will still be differences of the propagation delay for each data paths. An optimized NCL fulladder is shown in Fig. 5, where the propagation delay from the input to output C_O^0 or C_O^1 is defined as $Delay_{\mathrm{TH23}}$. When $C_i = 1$, $X = 1$, and $Y = 0$, the TH23 and TH35 gate in series will result in the worst case propagation delay of $Delay_{\mathrm{TH23}} + Delay_{\mathrm{TH35}}$.

When the dual-rail values are presented at different times on the two rails, and the temporal or the propagation delay differences are large enough, there will not be a temporal overlap between the wrong DATA and the correct DATA. In this situation, there will be no way to distinguish between the wrong DATA and the correct one, because of the lack of the additional reference information. When there is a temporal overlap between the wrong DATA and the correct DATA on the two rails, the illegal dual-rail code (1, 1) will be exhibited, and it can be easily detected on the logically determined variable boundary [21].

A. Principle of SSFRT

In Section II, we present the conditions that the fault can be delivered to the next stage. In these conditions, we can observe that the fault duration and the fault propagation delay have a huge impact on the fault propagation characteristics.

To improve the resistant against the fault attack, the SSFRT structure is presented in this paper. Its basic idea can be described as follows.

1) Restrict the dual-rail data by synchronizing the input data and delaying the propagation control signal to transform faults into illegal dual-rail codes. Specifically, by reducing the time interval of the dual-rail data and matching the propagation delay between the fault and the correct DATA, the preconditions (1) and (2) are not satisfied.
2) On the basis of the aforementioned idea, implement a multilevel faults/illegal code monitor and immediately start the local flushing once the fault/illegal code is detected.
3) With the ring structure and the dynamic random storage, implement the SFRT register to improve the fault-tolerance by relaxing the spatial restrictions.

If the computational block is composed of an optimized NCL fulladder as shown in Fig. 5, then the implementation of SSFRT can be shown in Fig. 6. Only when all the DATA wavefronts have arrived at the input synchronization block, the control signal $\mathrm{SYN}_{\mathrm{input}}$, produced by the TH33 gate, could

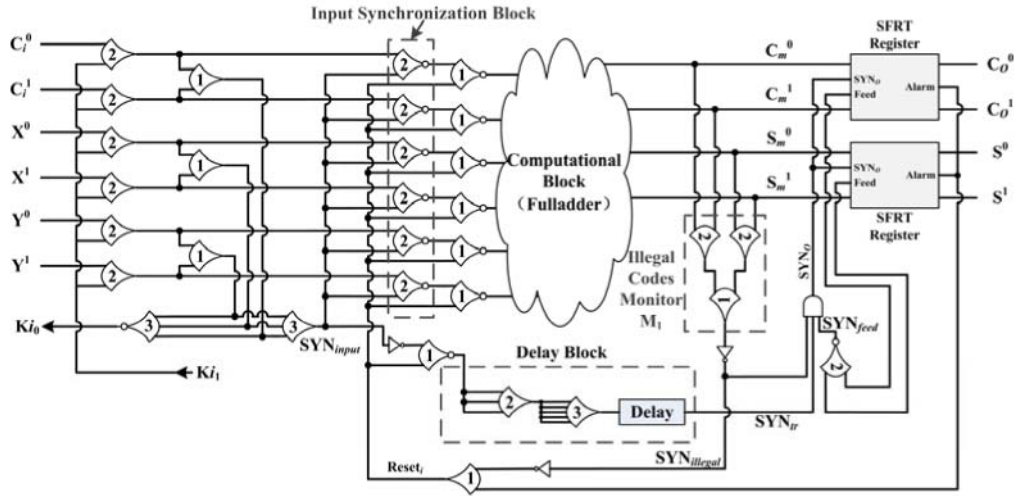


Fig. 6. SSFRT structure.

be set to high to allow the DATA wavefronts to flow into the computational block. Simultaneously, the control signal SYN_{input} will take to propagate through the delay block. As a result, the signal SYN_{tr} will be set to high. Finally, the signal SYN_O will be set to high, to allow the DATA wavefront to be received by the SRFT register. It is important to note that the delay block will result in the worst case propagation delay of the computational block, and a configurable delay module can be connected in series. As a result, the output data have already appeared on the output port of the computational block, before the signal SYN_{tr} transforms from low to high.

The illegal code monitor is used to detect the illegal dual-rail code (1, 1) on the rails. For example, if the illegal dual-rail code is exhibited on the rails (C_m^0, C_m^1), the output of the TH22 gate will be set to high, and then the output of the TH12 gate will be also set to high. Finally, the signal $SYN_{illegal}$ will be set to low through the inverter.

Now, let us assume that the transient or steady set fault is generated on the rail, whose error-free signal is low in the computational block prior to the correct DATA. As the fault is not allowed to flow into the SFRT register until the signal SYN_O is high, and the signal SYN_{tr} is not set to high until all of the DATA wavefronts have arrived at the input port of SFRT register, the illegal dual-rail code (1, 1) will be exhibited before the fault is delivered to the SFRT register, and the signal $SYN_{illegal}$ of the illegal code monitor M_1 will be set to high prior to the signal SYN_{tr} under the influence of the delay block. As a result, the signal SYN_O will still be low, and any data which is generated by the current computational block, will be prevented from flowing into the SFRT register. Moreover, the local reset signal $Reset_i$ will be set to high, to allow the NULL wavefront to overwrite the computational block. Finally, the input DATA will come to the computational block again for recomputation. It is the same when the faults in SFRT register is detected, and the output signal alarm is set to high.

The SFRT register is designed on the basis of the two-level and three-cycle ring structure, as shown in Fig. 7. Once the correct DATA has been delivered to the SFRT register, the self-feedback [19], [20] signal SYN_{feed} will be set to low immediately. It will make the signal SYN_O (in Fig. 6) set to low, and the subsequent wrong DATA that takes to propagate preceded by the correct DATA, will be prevented from flowing into the SFRT register. Once there exists any valid DATA in the two-level and three-cycle ring structure, the signal $Lock_0$/$Lock_1$ will be set to high, to prevent any other input. By the built-in path random swapping (PRS) block, the DATA wavefronts will choose the path randomly in the two-level and three-cycle ring structure. The mechanism of the PRS provides a reliable way to implement a dynamic and random storage of DATA wavefronts.

Assume that a wrong DATA is delivered to the SFRT register (note that this is the worst case assumption), it will be detected by the illegal dual-rail code monitor. Moreover, the internal faults caused by the soft-error can be easily detected using the built-in fault monitor M_2. Once an illegal code/internal fault is detected, the signal alarm will be set to high, and finally the local reset signal $Reset_i$ (in Fig. 6) will be also set to high. This will result in the local reset action, and then the NULL wavefronts will overwrite the computational block. Once the NULL wavefronts flow into the SFRT register, the signal Clear will be set to high, which will force the register to flush.

The structure of the PRS block is shown in Fig. 8. When the DATA wavefront arrives at the input rails (Pi_0^0, Pi_0^1) or (Pi_1^0, Pi_1^1), the illegal code monitor M_3 will estimate the validity of the DATA. If it is an illegal dual-rail code, the signal *invalid* will be set to high, and the signal Clear (in Fig. 7) will be also set to high by the built-in fault monitor M_2 in the SFRT register. This will force the register to flush, and the correct DATA wavefronts will flow into the SFRT register again. If it is a valid DATA, the true random number generator will generate a 1-bit random number on the falling edge of the control signal *ctr* [connected to the signal CD_0, CD_1, CD_2 (in Fig. 7)]. According to the random numbers, the DATA wavefronts will flow out through the data path ($P(i+1)_0^0$, $P(i+1)_0^1$) or ($P(i+1)_1^0$, and $P(i+1)_1^1$).

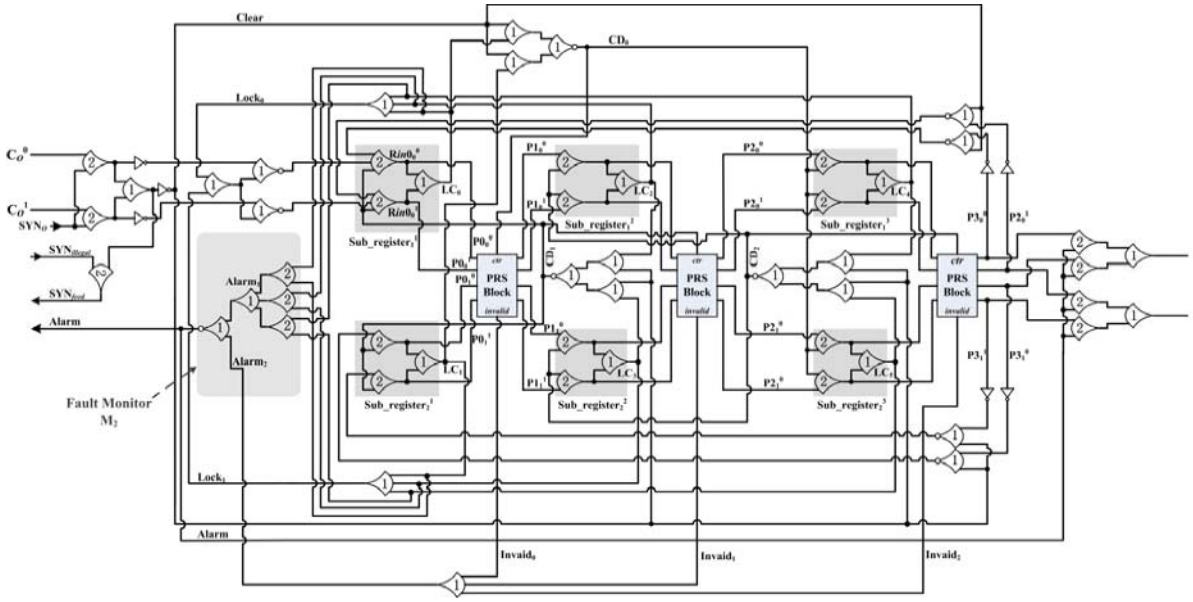


Fig. 7. SFRT register.

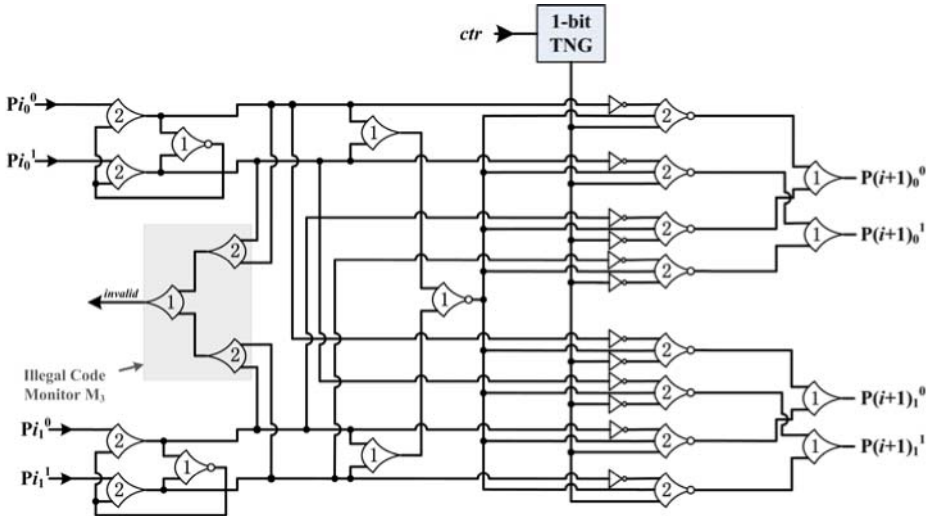


Fig. 8. PRS block.

B. Analysis of the Resistance to the Fault Propagation

Because the analysis of the transient set faults is similar to the steady set faults, for simplicity, it is assumed that only the steady set faults should be considered.

1) Wrong DATA Takes to Propagate Prior to the Correct DATA: As shown in Fig. 9, it is assumed that a steady set fault is generated on the rail C_m^0 (in Fig. 6) during the cycle 1. Furthermore, the wrong DATA (1, 0) takes to propagate prior to the correct DATA (0, 1) on the rails (C_m^0, C_m^1). The processing flow of the SSFRT structure is described as follows.

Step 1: Under the influence of the correct DATA on the rail C_m^1 and the delay between the signal $\mathrm{SYN_{input}}$ and $\mathrm{SYN_{tr}}$, the steady set fault on the rail C_m^0 will be transformed into the illegal dual-rail code (1, 1) before being delivered to the SFRT register, then the illegal dual-rail code will be captured by the illegal code monitor M_1, and the signal $\mathrm{SYN_{illegal}}$ will be set to low.

Steps 2–4: When the signal $\mathrm{SYN_{illegal}}$ is set to low, the signal SYN_O will be forced to low, and the steady set fault cannot be delivered to the SFRT register. At the same time, the local reset signal Reset_i will be set to high to allow the NULL wavefronts to overwrite the computational block. It should be noticed that the local reset action has no influence on the current state of NCL pipeline, and thus it will not result in the deadlock or the reset of the NCL pipeline. So, with the behaviors of the dual-rail encoded systems as described in [24], the SFRT register can resist the fault attacks.

Step 5: When the NULL wavefronts arrive at the output of the computational blocks, the signal $\mathrm{SYN_{illegal}}$ will be set to high, and then the signal Reset_i will be set to low.

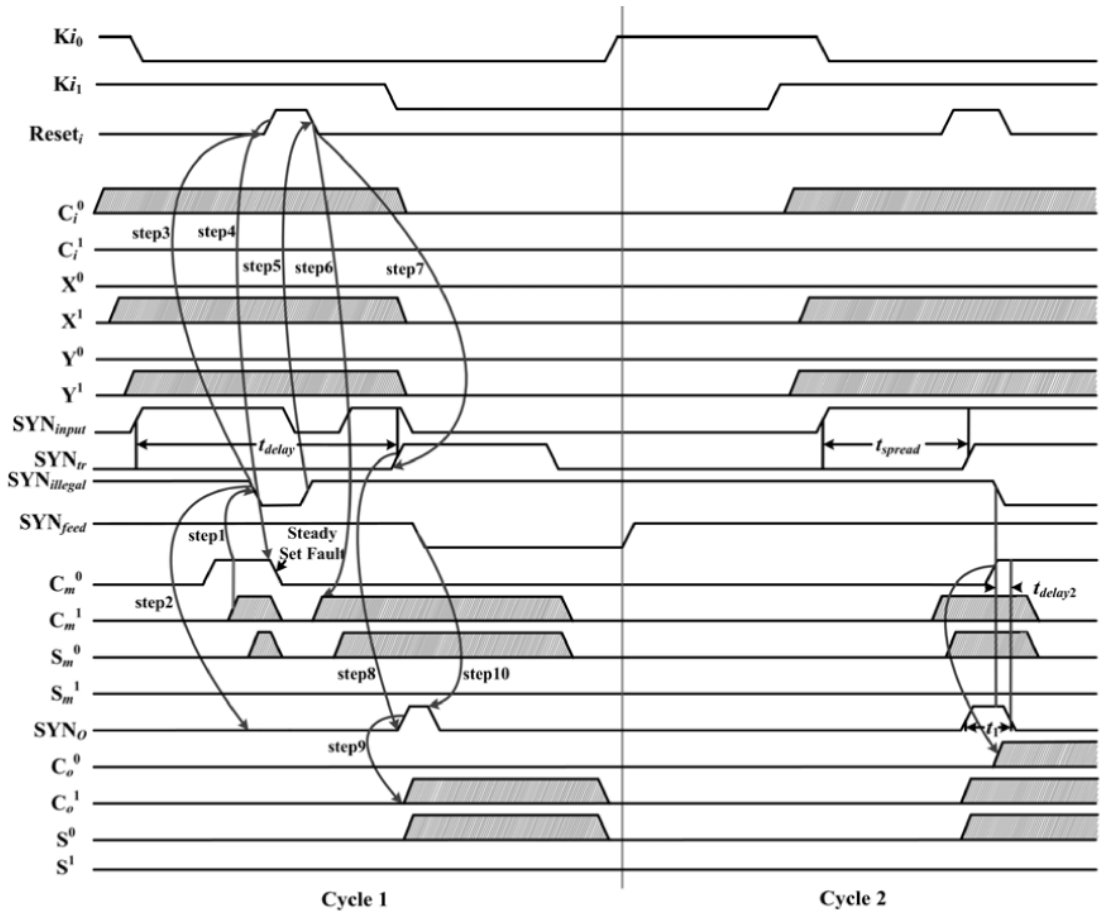


Fig. 9. Analysis of the resistance to the fault propagation of SSFRT.

Steps 6 and 7: The input DATA flows into the computational block for the recomputation. Simultaneously, the signal $\mathrm{SYN_{input}}$ is set to high again and propagates through the delay block. Consequently, the signal $\mathrm{SYN_{tr}}$ will be set to high. It is assumed that the propagation delay caused by the delay block is t_{spread}, and the delay between the rising edge of the signal $\mathrm{SYN_{input}}$ and $\mathrm{SYN_{tr}}$ is defined as t_{delay1}. Then as the signal $\mathrm{SYN_{illegal}}$ is set to high prior to the signal $\mathrm{SYN_{tr}}$ in the Step 1, the relationship that $t_{\mathrm{delay1}} > 2 \times t_{\mathrm{spread}}$ can be satisfied.

Steps 8 and 9: Because the signal $\mathrm{SYN_{illegal}}$, $\mathrm{SYN_{tr}}$, and $\mathrm{SYN_{feed}}$ are all high, the signal SYN_O will be also set to high, and the current DATA waveforns will be delivered to the SFRT register.

Step 10: After the DATA wavefronts are delivered to the SFRT register, the signal SYN_O will be set to low by the signal $\mathrm{SYN_{feed}}$ to stop the subsequent data input.

2) Wrong DATA and the Correct DATA Propagate at the Same Time: Similarly to 1), if both the wrong DATA and the correct DATA propagate at the same time, the signal $\mathrm{SYN_{illegal}}$ will be set to low prior to the signal SYN_O under the influence of the transition delay between the signal $\mathrm{SYN_{input}}$ and $\mathrm{SYN_{tr}}$. As a result, the wrong DATA cannot be delivered to the SFRT register, and then, the NULL wavefronts will overwrite the computational block.

3) Correct DATA Propagates Prior to the Wrong DATA: Let us assume that a steady set fault is injected, and the wrong DATA (1, 0) will not be generated on the rails (C_m^0, C_m^1) until the correct dual-rail DATA (0, 1) takes to propagate. Then the resistance to the fault propagation will be different depending on the duration of the steady set fault.

As shown in Fig. 9, it is assumed that a steady set fault is generated on the rail C_m^0 during the cycle 2, and the correct DATA on the rail C_m^1 takes to propagate prior to the steady set fault. If the steady set fault is generated during the period that the signal SYNO is high, the wrong DATA will be delivered to the SFRT register, because there exists the delay t_{delay2} during the period between the signal $\mathrm{SYN_{illegal}}$ is set to low and the signal SYN_O is set to low. More formally, if the steady set fault is generated in the interval t_1 during which the signal SYN_O is high, the wrong DATA will be delivered to the SFRT register. The interval t_1 depends on the time consumed in the duration and propagation of the signal $\mathrm{SYN_{feed}}$, as follows:

$$t_1 = 2 \times t_{\mathrm{TH22}} + t_{\mathrm{TH12}} + t_{\mathrm{TH22inv}} + t_{\mathrm{and}} + t_{\Delta 1}. \quad (6)$$

The wrong DATA is allowed to flow into the SFRT register in the time window t_2, which is in fact the propagation delay of the signal $\mathrm{SYN_{illegal}}$, as follows:

$$t_2 = t_{\mathrm{TH22}} + t_{\mathrm{TH12}} + t_{\mathrm{inv}} + t_{\mathrm{and}} + t_{\Delta 2}. \quad (7)$$

The parameters in (6) and (7) are defined as follows:

1) t_{TH22} is the signal propagation delay in the threshold gate TH22;
2) t_{TH12} is the signal propagation delay in the gate TH12;
3) t_{inv} is the signal propagation delay in the inverter;
4) t_{and} is the signal propagation delay in the AND gate;
5) $t_{\Delta 1}$ the maximum wire delay in the propagation path of the signal $\mathrm{SYN_{feed}}$;
6) $t_{\Delta 2}$ is the maximum wire delay in the propagation path of the signal $\mathrm{SYN_{illegal}}$.

As described, if the steady set fault on the rail C_m^0 is generated during the period that the signal SYN_O is high, the steady set fault will be delivered to the SFRT register. After that, the fault propagation characteristics will be different, depending on the temporal overlap between the fault on the rail $P0_0^0$ and the correct value on the rail $P0_0^1$.

It is assumed that the steady set fault occurs on the rail C_m^0, and preceded by the correct value on the rail C_m^1 (note that the case that the fault is delivered to the SFRT register prior to the correct value, will never occur under the influence of the transition delay between the signal $\mathrm{SYN_{input}}$ and $\mathrm{SYN_{tr}}$). Then, if the interval between the occurring of the fault on the rail $P0_0^0$ and the correct value on the rail $P0_0^1$ are large enough, the processing flow is as shown in Fig. 10(a).

Step 1: The correct value on the rail $P0_0^1$ takes to propagate to the rail $P1_0^1$ through the PRS block.

Step 2: The completion detection signal LC_2 (in Fig. 7) is set to high.

Step 3: The signal CD1 (in Fig. 7) is set to low.

Step 4: The NULL wavefronts overwrite the previous stage.

Obviously, the wavefronts steering [21] mechanism of NCL pipelines will prevent the fault from coming into the next stage.

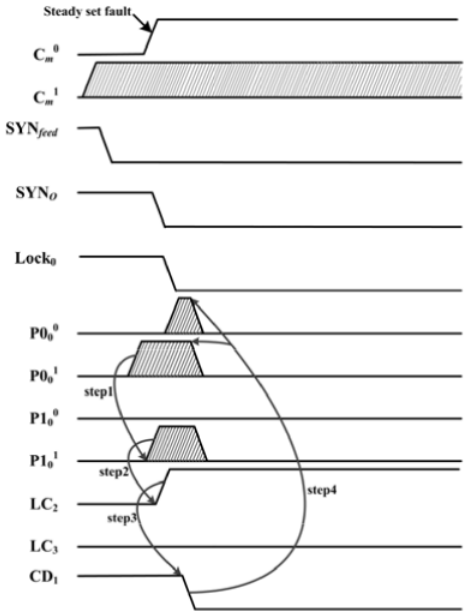
If the interval between the occurring of the fault on the rail $P0_0^0$ and the correct value on the rail $P0_0^1$ are not large enough, the processing flow is as shown in Fig. 10(b).

Step 1: Because there exists a temporal overlap between the fault and the correct DATA in the PRS block, the illegal dual-rail code (1, 1) will be exhibited, and it will be captured by the illegal code monitor M_2 of the PRS block. This will make the signal $\mathrm{Invalid_0}$ (in Fig. 7) set to high.

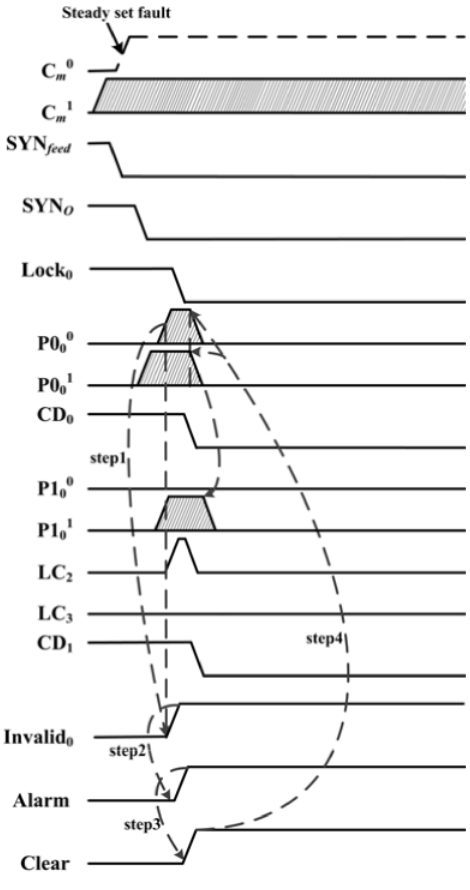
Step 2: The signal alarm (in Fig. 7) is set to high under the influence of the signal $\mathrm{Invalid_0}$, and finally the local reset signal Reset_i (in Fig. 6) will be also set to high. This will result in the local reset action, and then the NULL wavefronts will overwrite the computational block.

Steps 3 and 4: After the NULL wavefronts flow into the SFRT register, the signal Clear is set to high. It will force the register to flush.

If the steady set fault on the rial C_m^0 does not occur during the period that the signal SYN_O is high, the fault will be prevented from delivering to the SFRT register by the self-feedback signal $\mathrm{SYN_{feed}}$.



(a)



(b)

Fig. 10. Analysis of the fault propagation in the SFRT register. (a) Faults are suppressed by the wavefronts steering mechanism. (b) Faults are suppressed by the self-feedback mechanism.

C. Soft Error Detection

As a bridge between the pipeline stages, the SFRT register temporarily saves the computational results that are delivered by the previous stage. When the data in the SFRT register is tampered, the fault would not be detected by the next stage. Therefore, the SFRT registers tend to be an easy target of the

fault injection, such as soft errors. With the SSFRT in Fig. 6 for example, after delivered to the SFRT registers, the DATA wavefront will take to propagate in a way of random in the two-level and three-cycle ring structure. It is assumed that the attacker can induce the SEU with a precise control on the location, the correct DATA can be tampered (token insertion or deletion) only when the correct DATA resides in the same subregister at the same time. Because there are six subregisters in the two-level and three-cycle ring structure, and the DATA wavefront will randomly choose the path in the two-level and three-cycle ring structure under the influence of the PRS block, the probability that the SEU and the correct DATA reside in the same subregister will be reduced to one-sixth, compared with the solution in which there is only one subregister. Further improvements on the soft error resistance can be achieved by increasing the number of levels and cycles in the ring structure. It can be assumed that the number of level is n and the number of cycle is m (m must be an odd number), the success rate in a fault attack is P_{flip}

$$P_{\text{flip}} = \frac{1}{n} \times \frac{1}{m}. \tag{8}$$

Once a SEU is induced at an improper location, there will be two different DATA wavefronts in the SFRT register. As shown in Fig. 11, the soft error will be detected immediately by the built-in fault monitor M_2 in the SFRT register.

As shown in Fig. 11, it is assumed that after the DATA wavefront is delivered to the rails ($P0_0^0$, $P0_0^1$) in the SFRT register, it takes to propagate through the rails ($P1_0^0$, $P1_0^1$) and ($P2_1^0$, $P2_1^1$) one by one. Then, the processing flow is as follows.

Step 1: If a SEU is induced immediately at an improper location (e.g., $P1_0^0$) when the DATA wavefront arrives at the rails ($P2_1^0$, $P2_1^1$), the signal transformation from low to high on the rail $P1_0^0$ will be exhibited. This will lead to the action that the signal L C_2 is set to high at time t_1.

Step 2: The signal CD1 is forced to keep low, and then the new DATA wavefronts will be prevented from flowing into the previous stage. This will cause the monotonic transitioning to cease resulting in the deadlock.

Steps 3 and 4: The wrong DATA wavefront is delivered to the rails ($P2_0^0$, $P2_0^1$). Moreover, both the signal LC_4 and LC_5 will be high at time t_2, which will make the signal alarm set to high. Furthermore, the signal Clear will be set to high. As a result, the local flushing will be started to allow the NULL wavefronts to overwrite the computational block.

D. Comparison With Previous Designs

In [20], a built-in soft error correction (BISEC) technique has been proposed. By exploiting the handshaking protocol and dual-rail encoding, soft errors in the computational blocks are detected and corrected. In [20], it is assumed that the NCL registers and the completion detection circuitry are soft error free, because they are relatively small compared with the computational blocks and therefore less likely to be struck by particles. However, when facing a powerful adversary

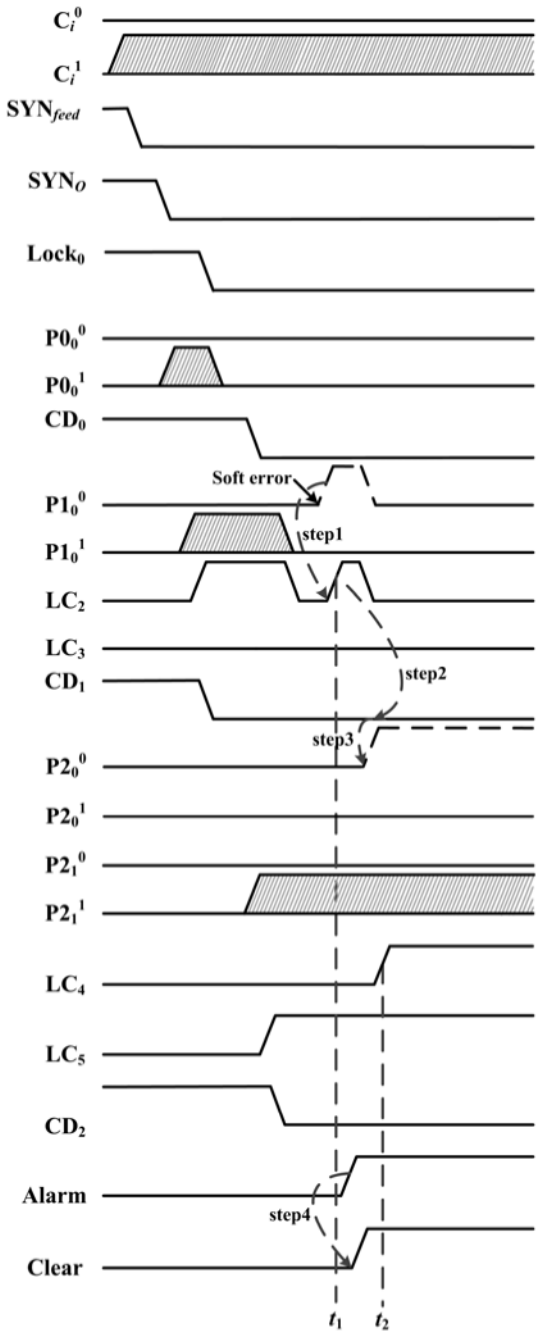


Fig. 11. Soft error detection.

(precise fault model, precise fault location), the argument does not hold. Furthermore, in the following two cases, the wrong DATA would be delivered to the next stage: 1) a fault reaches the inserted register before the DATA wavefronts, and the interval is large enough; and 2) a fault reaches the inserted register after the DATA wavefronts, and the interval is small enough.

Compared with BISEC, the SSFRT structure can cope with the situation where a fault takes to propagate prior to the correct DATA under the influence of the delay block, as shown in Section IV-B.1. When the correct DATA takes to propagate prior to the wrong DATA, the fault is allowed to flow into the SFRT register during the hold time of the signal $SYN_{illegal}$ as

shown in (7). However, it does not mean that the fault would be delivered to the next stage. As shown in Section IV-B3, if the interval between the occurring of fault and the correct value are large enough, the wavefront steering mechanism of NCL pipelines will prevent the fault from coming into the next stage. If the interval between the fault and the correct value are not large enough, the illegal dual-rail code will be exhibited. This will result in the local reset action, and then the NULL wavefront will overwrite the computational block. Only when a soft error is induced at an proper location where the correct DATA is resided, the fault would be delivered to the next stage, as shown in Section IV-C.

The SSFRT structure brings a great fault-tolerant improvement, but it suffers from a significant size overhead due to the additional logics from the SFRT register. The SSFRT structure leads to a hardware overhead which depends on the number of input/output rails of the computational block. With the increasing number of the input/output rails, more input synchronization units or SFRT registers are needed. It is assumed that the threshold gates in the SSFRT structure are built with the semistatic complimentary metal–oxide–semiconductor (CMOS) circuits, and the number of the levels and cycles in the ring structure are two and three, respectively. If the computational block is a full adder as shown in Fig. 5, 54 transistors are needed for the semistatic implementations. Then, approximately 1448 transistors are needed for the fault detection and correction circuitry, and the relative overhead is 2681.4%. If the computational block is a 4×4 unsigned multiplier consuming 2004 transistors [25], 1388 transistors are approximately needed for the fault detection and correction circuitry, and the overhead is approximately 69.3%. Similarly, if the computational block is a 4×4 open core s-box consuming 6672 transistors [26], 1388 transistors are approximately needed for the fault detection and correction circuitry, and the overhead is approximately 20.8%. Obviously, the SSFRT structure is suitable to be used in these situations, where the size of the computational block is very large and the number of the input/output rails of the computational block is less. Because of the input synchronization block and SFRT register, the delay overhead is variable and depends on the fault scenario. In the worst case scenario as shown in Fig. 15, the delay overhead is 2.661 ns due to the pipeline simulation. One of our aims for the future research is to quantify more precisely on the performance and area overhead.

V. Simulation Results

To estimate the fault resistance of the SSFRT structure, in our experiment every circuit is designed in TSMC 0.18 μm CMOS technology and simulated by Candence SPECTRE with the supply voltage 1.8 V. All transistors (nMOS or pMOS) have a channel width of 2 μm and a length of 180 nm. For simplicity, it is assumed that the input dual-rail DATA on the rails (C_i^0, C_i^1), (X^0, X^1), (Y^0, and Y^1) are set to (1, 0), (0, 1), (1, 0). It is the same with the other scenarios.

The steady set faults in the SFRT register will result in the illegal dual-rail code, which will be detected by the illegal code monitor M_3 in the PRS block, and then, the overwriting will be started immediately. So, only the transient set faults should be considered in our simulation.

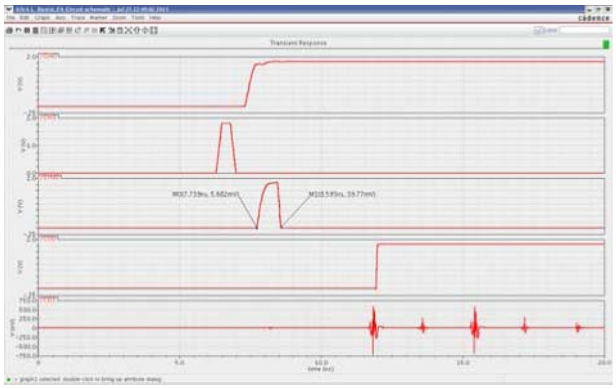


Fig. 12. Behavior of proposed framework when the fault propagates prior to the correct DATA.

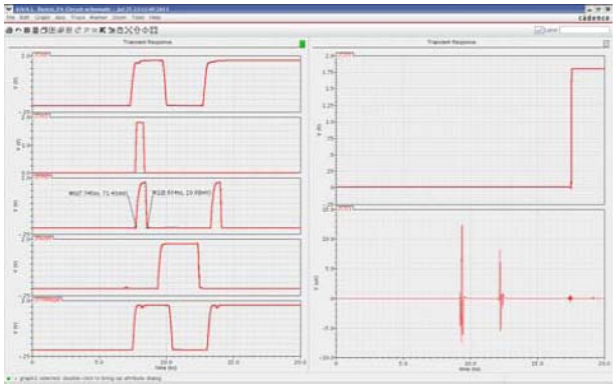


Fig. 13. Behavior of proposed framework when the fault is generated during the period that the signal SYN_O is high.

As shown in Fig. 12, when the fault on the rail C_m^1 of the computational block takes to propagate prior to the correct DATA on the rail C_m^0, the wrong DATA will not be delivered to the SFRT register, because the signal SYN_O will not be set to high until the correct DATA arrives at the rail C_m^0.

As shown in Fig. 13, when a transient fault on the wire C_m^1 occurs during the period that the signal SYN_O is high, the illegal dual-rail code (1, 1) will be exhibited on the rails (C_m^0, C_m^1), because of the binding between the signal SYN_O and the correct DATA. The signal $\mathrm{SYN}_{\mathrm{illegal}}$ will be set to low by the illegal code monitor M_3 in the PRS block, to prevent the wrong DATA from delivering to the SFRT register. Moreover, the local reset signal Reset_i will be set to high, to allow the NULL wavefronts to overwrite the computational block. Once the wrong DATA is eliminated, the input DATA will flow into the computational block again for the recomputation. Because the reset action is only performed in the local stage, and has no influence on the current state of the NCL pipeline, the attackers will not observe the fault sensitivity information of the NCL pipeline (e.g., deadlock reset). So, the local reset process is secure against the FSA attack [18].

As shown in Fig. 14, if a transient set fault is induced on the rail Rin_0^1 (in Fig. 7) at time 4.181 ns, before the correct

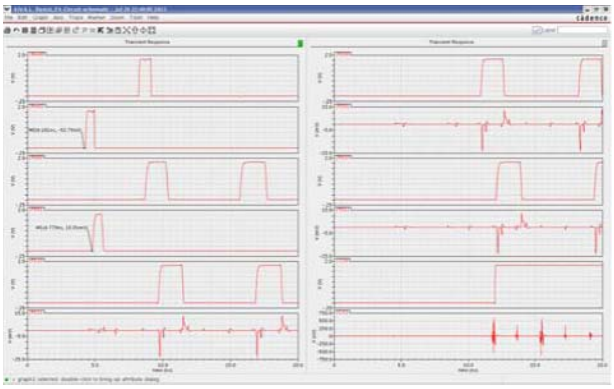


Fig. 14. Behavior of proposed framework when the fault is induced in SFRT register, before the correct DATA is delivered to SFRT.

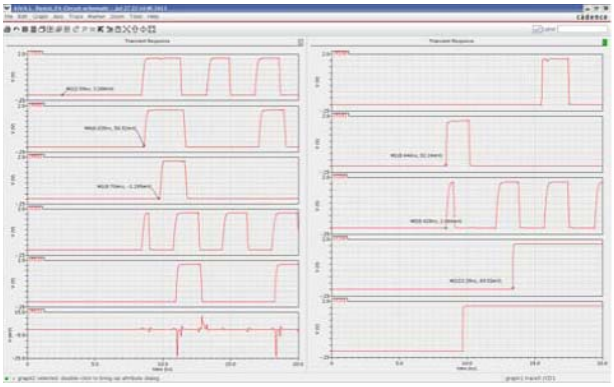


Fig. 15. Behavior of proposed framework when the fault is induced in SFRT register, after the correct DATA have been delivered to SFRT.

DATA (1, 0) is delivered to the SFRT register, the wrong DATA (0, 1) will occur in the register prior to the correct DATA. At this time, because the results of the computational block have not been delivered to the SFRT register, the signal Clear of the register will be high. Consequently, the wrong DATA cannot take to propagate in the SFRT register. In the same way, all the transient set faults, that occur prior to the input of the correct DATA, cannot take to propagate in the SFRT register.

As shown in Fig. 15, if a transient set fault is induced on the rail $P1_0^1$ at time 8.446 ns, after the correct DATA of the computational block have been delivered to SSFR. Then, the wrong DATA will flow out from the output of the SFRT register, and the illegal dual-rail code (1, 1) will be exhibited at time 13.39 ns, and then the illegal dual-rail code can be easily detected by the illegal codes monitor in the next stage.

It is important to note that the wrong DATA will be suppressed by the handshaking mechanism between stages in most cases, because of the wavefront's monotonically transition between the disjoint domains in NCL circuits. As shown in Fig. 15, the wrong DATA on the rail $P1_0^1$ can take to propagate to the rail $P0_0^1$, as the transient set faults occur during the period that the signal CD_0 is high. Nevertheless, the fault propagation will not continue because the transient set fault occurs at time 9.704 ns, when the signal CD_2 is low. Furthermore, even if the wrong DATA takes to propagate in the two-level and three-cycle ring structure, it will result in the deadlock or the occurrence of two DATA wavefronts in two stages in the same level, and the signal alarm will be set to high, to allow the NULL wavefront to overwrite the SFRT register. So, the faults injection can still be detected by the built-in fault monitor M_2 in the worst case scenario.

VI. Conclusion

In this paper, we mainly focus on the architecture-level improvements on the resistance against fault attacks. We have investigated the propagation characteristics of faults in NCL pipelines, and proposed the SSFRT structure based on the robustness of NCL pipelines and dual-rail encoded systems.

The following fault scenarios can be mitigated by the SFFRT structure.

1) The fault takes to propagate prior to the correct DATA on the output rails of the computational block.
2) The correct DATA takes to propagate prior to the fault on the output rails of the computational block.
3) The fault and the correct DATA take to propagate simultaneously on the output rails of the computational block.
4) The fault is induced in the SFRT register before the correct DATA has been delivered to the SFRT register.
5) The SEU is induced in the SFRT register after the correct DATA has been delivered to the SFRT register, but the SEU and the correct DATA do not reside in the same subregister at the same time.

Another contribution of this paper lies in a register design that offers an efficient resistance against the SEU. The design employs a two-level and three-cycle ring structure, and a more efficient resistance against the fault injection can be achieved. The SEU can result in a situation where the correct DATA is tampered (token insertion or deletion) only if the correct DATA simultaneously resides in the same subregister. Because there are six subregisters in the two-level and three-cycle ring structure, and the DATA wavefronts will randomly choose the path through the PRS block, the probability that a SEU and correct DATA reside in the same subregister will be reduced to one-sixth, compared with the solution in which there is only a single subregister.

We are now working on the implementation of an application-specific integration circuit for a further evaluation of the proposed framework.

References

[1] D. Boneh, R. A. DeMillo, and R. J. Lipton, "On the importance of checking cryptographic protocols for faults," in *Advances in Cryptology*. Berlin, Germany: Springer-Verlag, 1997, pp. 37–51.
[2] C. H. Kim, "Differential fault analysis against AES-192 and AES-256 with minimal faults," in *Proc. Workshop FDTC*, Santa Barbara, CA, USA, Aug. 2010, pp. 3–9.
[3] E. Trichina and R. Korkikyan, "Multi fault laser attacks on protected CRT-RSA," in *Proc. Workshop FDTC*, Santa Barbara, CA, USA, Aug. 2010, pp. 75–88.
[4] L. Hemme and L. Hoffmann, "Differential fault analysis on the SHA1 compression function," in *Proc. Workshop FDTC*, Nara, Japan, Sep. 2011, pp. 54–64.
[5] H. Sakamoto, Y. Li, K. Ohta, and K. Sakiyama, "Fault sensitivity analysis against elliptic curve cryptosystems," in *Proc. Workshop FDTC*, Nara, Japan, Sep. 2011, pp. 11–20.

[6] D. Gu, J. Li, S. Li, Z. Ma, Z. Guo, and J. Liu, "Differential fault analysis on lightweight blockciphers with statistical cryptanalysis techniques," in *Proc. Workshop FDTC*, Leuven, Belgium, Sep. 2012, pp. 27–33.
[7] W. Fischer and C. A. Reuter, "Differential fault analysis on Grøstl," in *Proc. Workshop FDTC*, Leuven, Belgium, Sep. 2012, pp. 44–56.
[8] S. Banik, S. Maitra, and S. Sarkar, "A differential fault attack on the grain family of stream ciphers," in *Proc. 14th Int. Workshop CHES*, Leuven, Belgium, Sep. 2012, pp. 122–139.
[9] S. Guilley, L. Sauvage, J.-L. Danger, and N. Seimane, "Fault injection resilience," in *Proc. Workshop FDTC*, Santa Barbara, CA, USA, Aug. 2010, pp. 51–65.
[10] K. Yumbul, S. S. Erdem, and E. Savas, "On protecting cryptographic applications against fault attacks using residue codes," in *Proc. Workshop FDTC*, Nara, Japan, Sep. 2011, pp. 69–79.
[11] K. Ma, H. Liang, and K. Wu, "Homomorphic property-based concurrent error detection of RSA: A countermeasure to fault attack," *IEEE Trans. Comput.*, vol. 61, no. 4, pp. 1040–1049, Jul. 2012.
[12] S. Briais, J.-M. Cioranesco, J.-L. Danger, S. Guilley, D. Naccache, and T. Porteboeuf, "Random active shield," in *Proc. Workshop FDTC*, Leuven, Belgium, Sep. 2012, pp. 103–114.
[13] Y. Li, K. Sakiyama, S. Gomisawa, T. Fukunaga, J. Takahashi, and K. Ohta, "Fault sensitivity analysis," in *Proc. 12th Int. Workshop CHES*, Santa Barbara, CA, USA, Aug. 2010, pp. 320–334.
[14] Y. Li, K. Ohta, and K. Sakiyama, "New fault-based side-channel attack using fault sensitivity," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 88–97, Feb. 2012.
[15] Y. Monnet, M. Renaudin, R. Leveugle, S. Dumount, and F. Bouesse, "An asynchronous DES crypto-processor secured against fault attacks," TIMA Lab., Grenoble, France, Tech. Rep. TIMA-RR-06/02–04-FR, 2005.
[16] K. J. Kulikowski, M. G. Karpovsky, Z. Wang, A. Kulikowski, and A. Taubin, "Concurrent fault detection for secure QDI asynchronous circuits," in *Proc. 2nd Workshop DSN*, Anchorage, AK, USA, Jun. 2008, pp. 1–6.
[17] S. Peng and R. Manohar, "Efficient failure detection in pipelined asynchronous circuits," in *Proc. 20th IEEE Int. Symp. Defect Fault Tolerance VLSI Syst.*, Oct. 2005, pp. 484–493.
[18] Y. Li, K. Sakiyama, S. Gomisawa, T. Fukunaga, J. Takahashi, and K. Ohta, "Fault sensitivity analysis," in *Proc. 12th Int. Workshop CHES*, Santa Barbara, CA, USA, Aug. 2010, pp. 320–334.
[19] W. Kuang, E. Xiao, C. M. Ibarra, and P. Zhao, "Design asynchronous circuits for soft error tolerance," presented at the IEEE Int. Conf. Integr. Circuit Design Technol., Austin, TX, USA, Jun. 2007, pp. 1–5.
[20] W. Kuang, P. Zhao, J. S. Yuan, and R. F. DeMara, "Design of asynchronous circuits for high soft error tolerance in deep submicrometer CMOS circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 3, pp. 410–422, Mar. 2010.
[21] K. M. Fant, *Logically Determined Design*. Hoboken, NJ, USA: Wiley, 2005, pp. 19–27.
[22] C. L. Seitz, "System timing," in *Introduction to VLSI Systems*. Boston, MA, USA: Addison-Wesley, 1980, pp. 218–262.
[23] S. C. Smith and J. Di, *Designing Asynchronous Circuits Using NULL Convention Logic*. San Rafael, CA USA: Morgan & Claypool, 2009, p. 8.
[24] J. Waddle and D. Wagner, "Fault attacks on dual-rail encoded systems," Dept. EECS, Univ. California, Berkeley, CA, USA, Tech. Rep. UCB/CSD-4-1347, Aug. 2004.
[25] S. K. Bandapati, S. C. Smith, and M. Choi, "Design and characterization of NULL convention self-timed multipliers," *IEEE Des. Test. Comput.*, vol. 30, no. 6, pp. 26–36, Nov./Dec. 2003.
[26] D. Sokolov, J. Murphy, A. Bystrov, and A. Yakovlev, "Improving the security of dual-rail circuits," in *Proc. 6th Int. Workshop CHES*, Aug. 2004, pp. 282–297.

Qingyu Ou received the B.S. degree in computer software and the M.S. degree in electrical engineering from PLA Information Engineering University, Zhengzhou, China, in 2001 and 2004, respectively.

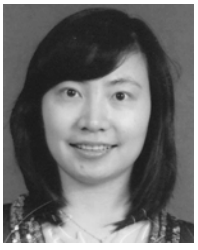
He has been an Associate Professor with the Naval University of Engineering, Wuhan, China, since 2012. His current research interests include asynchronous circuits and steel-reinforced aluminum cable resistant IC design.

Fang Luo received the B.S. degree in mathematics and the M.S. degree in electrical engineering from PLA Information Engineering University, Zhengzhou, China, in 2004 and 2008, respectively.

She has been a Lecturer with the Naval University of Engineering, Wuhan, China, since 2009. Her current research interests include cryptanalysis and steel-reinforced aluminum cable resistant IC design.

Shilei Li received the Ph.D. degree in control science and engineering from the National University of Defense Technology, Changsha, China, in 2010.

He is currently a Lecturer with the Naval University of Engineering, Wuhan, China. His current research interests include information security and computer simulation.



Lu Chen received the Ph.D. degree in information security from Wuhan University, Wuhan, China, in 2010.

Her current research interests include information security.