IEICE TRANS. INF. & SYST., VOL.E98–D, NO.10 OCTOBER 2015

LETTER

Strongly Secure Scan Design Using Generalized Feed Forward Shift Registers

Hideo FUJIWARA[†a)], ***Fellow and*** **Katsuya FUJIWARA**[††], ***Member***

SUMMARY In our previous work [12], [13], we introduced generalized feed-forward shift registers (GF^{2}SR, for short) to apply them to secure and testable scan design, where we considered the security problem from the viewpoint of the complexity of identifying the structure of GF^{2}SRs. Although the proposed scan design is secure in the sense that the structure of a GF^{2}SR cannot be identified only from the primary input/output relation, it may not be secure if part of the contents of the circuit leak out. In this paper, we introduce a more secure concept called *strong security* such that no internal state of strongly secure circuits leaks out, and present how to design such strongly secure GF^{2}SRs.

key words: *design-for-testability, scan design, generalized feed-forward shift registers, security, scan-based side-channel attack*

1. Introduction

It is important to find an efficient design-for-testability (DFT) methodology that satisfies both security and testability, although there exists an inherent contradiction between security and testability for digital circuits. Scan design is a powerful DFT technique that provides high controllability and observability over a chip and yields high fault coverage [1]. However, this also accommodates reverse engineering, which damages security. For secure chip designers, there is a demand to protect secret data from side-channel attacks and other hacking schemes [2]. Different approaches [3]–[9] have been proposed to solve this problem. All the approaches except [7] add extra hardware outside of the scan chain. Disadvantages of this are high area overhead, timing overhead or performance degradation, increased complexity of testing, and limited security for the registers part, among others.

In a previous paper [10], we reported a secure and testable scan design approach by using extended shift registers called *SR-equivalents* that are functionally equivalent but not structurally equivalent to shift registers. We then extended the class of SR-equivalents to a wider class of *SR-quasi-equivalents* [11]. We further introduced *generalized feed-forward shift registers* (GF^{2}SR, for short) to apply them to secure and testable scan design [12], [13]. Our proposed approach is only to replace part of the original scan chains with modified scan chains, which satisfy both testability and security of circuits. This method requires very little area overhead and no performance overhead. Moreover, no additional keys and controller circuits outside of the scan chain are needed, thus making the scheme low-cost and efficient.

We considered the security problem from the viewpoint of the complexity of identifying the structure of GF^{2}SRs [12], [13]. There is another viewpoint for the security, i.e., the possibility of leakage of the contents of GF^{2}SRs. In this paper, by looking at the security problem from this viewpoint, we introduce a more secure concept called *strong security* such that no internal state of strongly secure circuits leaks out, and present a method for designing such strongly secure GF^{2}SRs.

2. Testability of Generalized Feed-Forward Shift Registers

In our previous papers [12], [13], we introduced a class of extended shift registers called *generalized feed-forward shift registers (GF^{2}SR)*. Figure 1 illustrates a general structure of GF^{2}SRs. In this figure, $f_0, f_1, \ldots, f_k$ are arbitrary logic functions. Figure 2 (a) shows an example of a 3-stage GF^{2}SR, R_1. Generally, for any GF^{2}SR with k flip-flops, the input value applied to the input x at any time t appears at the output z after k clock cycles with exclusive-OR of some logic function f of $x(t+1), x(t+2), \ldots, x(t+k)$, i.e., the output z at time $t+k$ behaves in accordance with the following equation.

$$z(t+k) = x(t) \oplus f(x(t+1), x(t+2), \ldots, x(t+k)).$$

As an example, consider a 3-stage GF^{2}SR, R_1, given in Fig. 2 (a). By using symbolic simulation, we can obtain an output sequence $(z(t), z(t+1), z(t+2), z(t+3))$ and the output $z(t+3) = x(t) \oplus x(t+2)x(t+1)$ as shown in Fig. 2 (b). From the result of symbolic simulation, we can derive equations to obtain an input sequence $(x(t), x(t+1), x(t+2))$ that transfers R_1 from any state to the desired final state $(y_1(t+3), y_2(t+3), y_3(t+3))$ as illustrated in Fig. 2 (b). Similarly, we can derive equations to determine uniquely the initial state $(y_1(t), y_2(t), y_3(t))$ from the input/output sequence as illustrated in Fig. 2 (b).

Generally, as for any circuit C of GF^{2}SR with k flip-

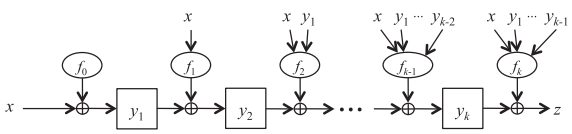
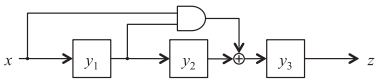


Fig. 1 Generalized feed-forward shift register (GF^{2}SR)

Manuscript received April 26, 2015.
Manuscript revised June 5, 2015.
Manuscript publicized June 24, 2015.
[†]The author is with Osaka Gakuin University, Suita-shi, 564–8511 Japan.
[††]The author is with Akita University, Akita-shi, 010–8502 Japan.
a) E-mail: fujiwara@ogu.ac.jp
DOI: 10.1587/transinf.2015EDL8100



(a) GF^{2}SR, R_1

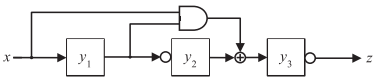
x	y_1	y_2	y_3	z
$x(t)$	$y_1(t)$	$y_2(t)$	$y_3(t)$	$z(t) = y_3(t)$
$x(t+1)$	$x(t)$	$y_1(t)$	$y_2(t) \oplus x(t) \cdot y_1(t)$	$z(t+1) = y_2(t) \oplus x(t) \cdot y_1(t)$
$x(t+2)$	$x(t+1)$	$x(t)$	$y_1(t) \oplus x(t+1) \cdot x(t)$	$z(t+2) = y_1(t) \oplus x(t+1) \cdot x(t)$
$x(t+3)$	$x(t+2) = y_1(t+3)$	$x(t+1) = y_2(t+3)$	$x(t) \oplus x(t+2) \cdot x(t+1) = y_3(t+3)$	$z(t+3) = x(t) \oplus x(t+2) \cdot x(t+1)$

$x(t) = y_3(t+3) \oplus y_2(t+3) \cdot y_1(t+3)$
$x(t+1) = y_2(t+3)$
$x(t+2) = y_1(t+3)$

$y_1(t) = z(t+2) \oplus x(t+1) \cdot x(t)$
$y_2(t) = z(t+1) \oplus x(t) \cdot y_1(t)$
$= z(t+1) \oplus x(t) \cdot z(t+2) \oplus x(t+1) \cdot x(t)$
$y_3(t) = z(t)$

(b) Symbolic simulation

Fig. 2 Example of GF^{2}SR, R_1



(a) GF^{2}SR, R_2

x	y_1	y_2	y_3	z
$x(t)$	$y_1(t)$	$y_2(t)$	$y_3(t)$	$z(t) = y_3(t) \oplus 1$
$x(t+1)$	$x(t)$	$y_1(t) \oplus 1$	$y_2(t) \oplus x(t) \cdot y_1(t)$	$z(t+1) = y_2(t) \oplus x(t) \cdot y_1(t) \oplus 1$
$x(t+2)$	$x(t+1)$	$x(t) \oplus 1$	$y_1(t) \oplus x(t+1) \cdot x(t) \oplus 1$	$z(t+2) = y_1(t) \oplus x(t+1) \cdot x(t)$
$x(t+3)$	$x(t+2) = y_1(t+3)$	$x(t+1) \oplus 1 = y_2(t+3)$	$x(t) \oplus x(t+2) \cdot x(t+1) \oplus 1 = y_3(t+3)$	$z(t+3) = x(t) \oplus x(t+2) \cdot x(t+1)$

(b) Symbolic simulation

Fig. 3 Strongly secure GF^{2}SR, R_2

flops, *(1) for any internal state of C a transfer sequence (of length k) to the state (final state) can be generated only from the connection information of C, independently of the initial state; (2) any present state (initial state) of C can be identified from the input-output sequence (of length k) and the connection information of C.*

Therefore, for the class of GF^{2}SRs, we can easily generate scan-in and scan-out sequences such that both scan-in and scan-out operations can be overlapped and hence testing can be done in the same way as the conventional scan testing. The test sequence is of the same length as the conventional scan design. There is no need to change traditional ATPG algorithm though a logic implication process is needed only for the GF^{2}SR after ATPG. To reduce the area overhead as much as possible, not all scan chains are replaced with modified scan chains. Only parts of scan chains necessary to be secure are replaced with modified GF^{2}SR scan chains. The delay overhead due to additional logic and XOR gates influences only scan operation, and hence there is no delay overhead for normal operation.

3. Security of Generalized Feed-Forward Shift Registers

When we consider a secure scan design, we need to assume what the attacker knows and how he can potentially make the attack. Here, we assume that *the attacker does not know the detailed information in the gate-level design, and that the attacker knows the presence of test pins (scan in/out, scan, and reset) and modified scan chains. However, he/she does not know the structure of extended scan chains.* Based on this assumption, we consider the security to prevent scan-based attacks.

In previous papers [11]–[13], we introduced a concept called *scan-secure* as follows. A circuit C with a single input x, a single output z, and k flip-flops is called *scan-secure* if the attacker cannot determine the structure of C. The security level of the secure scan architecture based on those GF^{2}SRs is determined by the probability that an attacker can identify the structure of the GF^{2}SR used in the circuit, and hence the attack probability approximates to the reciprocal of the cardinality of the class of GF^{2}SRs. In [12], [13] we showed the cardinality of the class of k-stage GF^{2}SRs is $2^{(2^{k+1}-1)}$. Hence, it is very hard and intractable to identify the structure of a given GF^{2}SR from the information on input/output relation only.

Although the structure of a GF^{2}SR is hard to be identified, it may not be secure if part of the contents of the GF^{2}SR leak out. For example, consider again the GF^{2}SR, R_1, and the result of symbolic simulation, shown in Fig. 2. When $y_1(t+3) = 0$, it holds that $(x(t), x(t+1), x(t+2)) = (y_3(t+3), y_2(t+3), y_1(t+3))$, i.e., any input sequence $(x(t), x(t+1), x(t+2))$ that transfers R_1 from any state to the desired final state $(y_1(t+3), y_2(t+3), y_3(t+3))$ becomes $(y_3(t+3), y_2(t+3), y_1(t+3))$ when $y_1(t+3) = 0$. This means R_1 behaves in the same way as a shift register during scan-in operation when $y_1(t+3) = 0$, and hence it is not secure when the attacker regards R_1 as a shift register and tries to initialize it to a final state with $y_1(t+3) = 0$. Similarly, when $x(t) = 0$, it holds that $(y_1(t), y_2(t), y_3(t)) = (z(t+2), z(t+1), z(t))$, i.e., the output sequence $(z(t), z(t+1), z(t+2))$ equals to $(y_3(t), y_2(t), y_1(t))$ when $x(t) = 0$. This means R_1 behaves in the same way as a shift register during scan-out operation when $x(t) = 0$, and hence it is not secure when the attacker regards R_1 as a shift register and tries to observe a present state of R_1 by applying an input sequence such that the first

input $x(t)$ happens to be 0. In this way, it may happen that the attacker succeeds in initializing the contents of R_1 and/or observing the contents of R_1, though he/she does not notice them.

To avoid such leakage, we consider more secure scan registers whose contents never leak out. We define new concepts in the following. Consider a circuit C with a single input, a single output, and k flip-flops. C is called to be *scan-in secure* if for any internal state of C a transfer sequence (of length k) to the state (final state) can be generated only from the connection information of C, independently of the initial state, such that the transfer sequence is always different from that of a k-stage shift register. C is called to be *scan-out secure* if any present state (initial state) of C can be identified only from the *input-output sequence* (of length k) and the connection information of C, such that the output sequence is always different from that of a k-stage shift register. C is called to be *strongly secure* if C is scan-in secure and scan-out secure.

Consider a 3-stage GF^{2}SR, R_1, given in Fig. 2 (a). As we mentioned above, R_1 behaves in the same way as a 3-stage shift register during scan-in operation when $y_1(t+3) = 0$, and hence it is not scan-in secure. Also, R_1 behaves in the same way as a 3-stage shift register during scan-out operation when $x(t) = 0$, and hence it is not scan-out secure. Next, consider another 3-stage GF^{2}SR, R_2, given in Fig. 3 (a). From the result of symbolic simulation shown in Fig. 3 (b), we can see $y_2(t + 3)$ never equals $x(t + 1)$, and hence R_2 is scan-in secure. Similarly, we can see $z(t)$ never equals $y_3(t)$, which implies R_2 is scan-out secure. Therefore, R_2 is strongly secure. In the following section, we consider how to design strongly secure GF^{2}SRs.

4. How to Design Strongly Secure GF^{2}SRs

Consider a GF^{2}SR C with input x, output z, and k flip-flops $y_1, y_2, \ldots, y_k$, such that the most left XOR gate is located between y_p and y_{p+1} as shown in Fig. 4 (a) and the most right XOR gate is located between y_{q-1} and y_q as shown in Fig. 4 (b). As illustrated in Fig. 4 (a), if there is at least one NOT gate between a primary input x and flip-flop y_p, the final state of $(y_1, y_2, \ldots, y_p)$ of C is always different from that of a shift register. Hence, we can see C is scan-in secure. Similarly, as illustrated in Fig. 4 (b), if there is at least one NOT gate between flip-flop y_q and a primary output z, the output sequence of C is always different from that of a shift register, and hence C is scan-out secure. If there is no flip-flop between the most right XOR gate and a primary output z, we need to add a dummy flip-flop between them so that we can insert a NOT gate on the flip-flop and make C scan-out secure.

From the above observation, we can see that any GF^{2}SR can be modified to be scan-in secure by inserting at least one NOT gate as illustrated in Fig. 4 (a). Also, we can see that any GF^{2}SR can be modified to be scan-out secure by inserting a dummy flip-flop if necessary and at least one NOT gate as illustrated in Fig. 4 (b). We present these methods in the following.

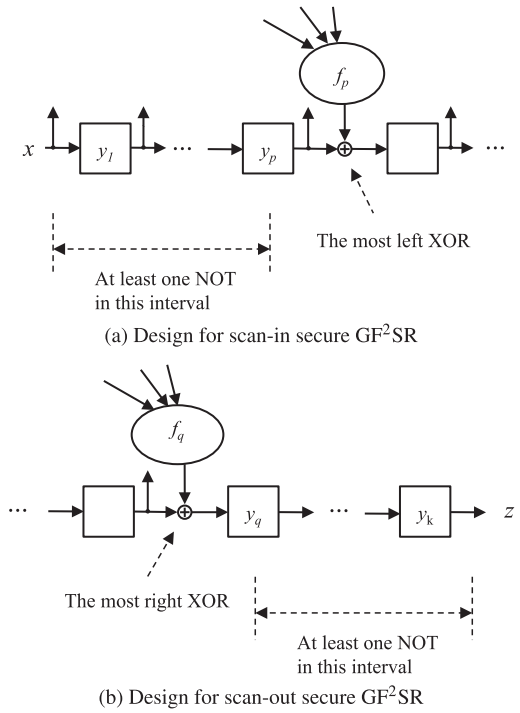


Fig. 4 Design for strongly secure GF^{2}SR

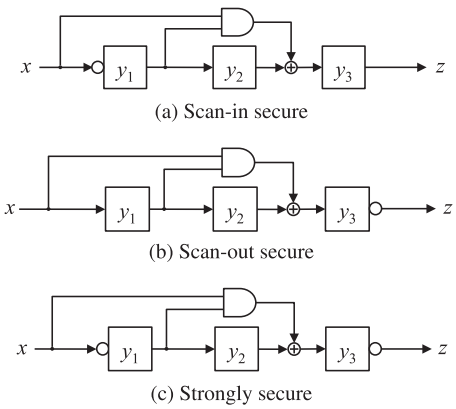


Fig. 5 Making strongly secure by inserting NOT

Method for making scan-in secure:

(1) If there is no NOT gate between a primary input x and flip-flop y_p (see Fig. 4 (a)), insert at least one NOT gate between them.

Method for making scan-out secure:

(1) If there is no flip-flop between the most right XOR gate and a primary output z, add a dummy flip-flop between them.
(2) If there is no NOT gate between flip-flop y_q and a primary output z (see Fig. 4 (b)), insert at least one NOT gate between them.

As an example, consider GF^{2}SR, R_1, shown in Fig. 2.

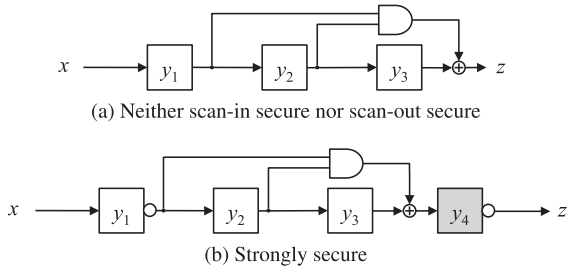


Fig. 6 Making strongly secure by inserting NOT and dummy FF

As mentioned in the previous section, R_1 is neither scan-in secure nor scan-out secure. We apply step (1) of the method for making scan-in secure. Here, flip-flop y_p is y_2 in R_1. Since there is no NOT gate between the primary input x and flip-flop y_2, we insert a NOT gate between them. Figure 5 (a) shows a result by inserting one NOT gate. It is obvious that the modified circuit is scan-in secure since the content of flip-flop y_1 is always different from that of the shift register.

Next, we apply the method for making scan-out secure to R_1. Here, flip-flop y_q is y_3 in R_1, and hence we apply step (2) by inserting a NOT gate between y_3 and the primary output z. Figure 5 (b) shows a result. It is obvious that the modified circuit is scan-out secure since the first observed content of flip-flop y_3 is always different from that of the shift register.

Figure 5 (c) shows a result that both methods for making scan-in secure and scan-out secure are applied to R_1 so that it is scan-in secure and scan-out secure, and hence strongly secure.

As another example, consider a GF^{2}SR, shown in Fig. 6 (a), which is neither scan-in secure nor scan-out secure. Since there is no flip-flop between the most right XOR gate and a primary output, we add a dummy flip-flop, and then insert two NOT gates to make it scan-in secure and scan-out secure as shown in Fig. 6 (b).

5. Conclusion

In our previous work, we reported a secure and testable scan design approach by using generalized feed-forward shift registers [12], [13], where we considered the security problem from the viewpoint of the difficulty or complexity of identifying the structure of GF^{2}SRs. There is another viewpoint for the security, i.e., the possibility of leakage of the contents of GF^{2}SRs. In this paper, by looking at the security problem from this viewpoint, we have introduced a new concept of *strong security* such that no internal state of strongly secure circuits leaks out, and presented how to design such strongly secure GF^{2}SRs. We have shown a straightforward method for making a given GF^{2}SR strongly secure by adding inverters and at most one dummy flip-flop.

References

[1] H. Fujiwara, Logic Testing and Design for Testability, The MIT Press, 1985.
[2] K. Hafner, H.C. Ritter, T.M. Schwair, S. Wallstab, M. Deppermann, J. Gessner, S. Koesters, W. Moeller, and G. Sandweg, "Design and test of an integrated cryptochip," IEEE Design and Test of Computers, vol.8, no.4, pp.6–17, Dec. 1991.
[3] D. Hély, F. Bancel, M.-L. Flottes, and B. Rouzeyre, "Securing scan control in crypto chips," Journal of Electronic Testing - Theory and Applications, vol.23, no.5, pp.457–464, Oct. 2007.
[4] B. Yang, K. Wu, and R. Karri, "Secure scan: A design-for-test architecture for crypto chips," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol.25, no.10, pp.2287–2293, 2006.
[5] J. Lee, M. Tehranipoor, C. Patel, and J. Plusquellic, "Securing designs against scan-based side-channel attacks," IEEE Trans. Dependable and Secure Computing, vol.4, no.4, pp.325–336, 2007.
[6] S. Paul, R.S. Chakraborty, and S. Bhunia, "VIm-Scan: A low overhead scan design approach for protection of secret key in scan-based secure chips," Proc. 25th IEEE VLSI Test Symposium, pp.455–460, 2007.
[7] G. Sengar, D. Mukhopadhyay, and D.R. Chowdhury, "Secured flipped scan-chain model for crypto-architecture," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol.26, no.11, pp.2080–2084, Nov. 2007.
[8] U. Chandran and D. Zhao, "SS-KTC: A high-testability low-overhead scan architecture with multi-level security integration," Proc. 27th IEEE VLSI Test Symposium, pp.321–326, 2009.
[9] M.A. Razzaq, V. Singh, and A. Singh, "SSTKR: Secure and testable scan design through test key randomization," Proc. 20th IEEE Asian Test Symposium, pp.60–65, 2011.
[10] H. Fujiwara and M.E.J. Obien, "Secure and testable scan design using extended de Brujin graph," Proc. 15th Asia and South Pacific Design Automation Conference, pp.413–418, 2010.
[11] K. Fujiwara, H. Fujiwara, and H. Tamamoto, "Secure and testable scan design utilizing shift register quasi-equivalents," IPSJ Trans. System LSI Design Methodology, vol.6, pp.27–33, 2013.
[12] K. Fujiwara and H. Fujiwara, "WAGSR: Web application for generalized feed forward shift registers," Proc. 13th IEEE Workshop on RTL and High Level Testing, pp.1.2.1–1.2.7, 2012.
[13] K. Fujiwara and H. Fujiwara, "Generalized feed-forward shift registers and their application to secure scan design," IEICE Trans. Inf. & Syst., vol.E96-D, no.5, pp.1125–1133, 2013.