WAGSR: Web Application for Generalized Feed Forward Shift Registers

Katsuya Fujiwara$^1$ and Hideo Fujiwara$^2$

$^1$Faculty of Engineering and Resource Science
Akita University
Akita, 010-8502, JAPAN
fujiwara@ie.akita-u.ac.jp

$^2$Faculty of Informatics
Osaka Gakuin University
Osaka, 564-8511, JAPAN
fujiwara@ogu.ac.jp

Abstract—In this paper, we introduce generalized feed-forward shift registers (GF$^2$SR, for short) to apply them to secure and testable scan design. Previously, we introduced SR-equivalents and SR-quasi-equivalents which can be used in secure and testable scan design, and showed inversion-inserted linear feed-forward shift registers (I$^2$LF$^2$SR, for short) are useful circuits for the secure and testable scan design. GF$^2$SR is an extension of I$^2$LF$^2$SR and the class is much wider than that of I$^2$LF$^2$SR. Since the cardinality of the class of GF$^2$SR is much larger than that of I$^2$LF$^2$SR, the security level of scan design with GF$^2$SR is much higher than that of I$^2$LF$^2$SR. We consider how to control/observe GF$^2$SR to guarantee easy scan-in/out operations, i.e., state-justification and state-identification problems are considered. A program called WAGSR (Web Application for Generalized feed-forward Shift Registers) is presented to solve those problems.

Keywords – design-for-testability; scan design; shift register equivalents; shift register quasi-equivalents; generalized feed-forward shift registers; security; scan-based side-channel attack.

1. INTRODUCTION

The design of secure chips demands protection of secret information, which may cause conflicts with the requirements for making the chip easily testable. While testing techniques such as scan design entail increased testability (controllability and observability) of the chip, they can also allow access to important data in a secure chip a lot easier. This makes it difficult for scan chains to be used especially in special cryptographic circuits where secret key streams are stored in internal registers, thus a problem in testing these types of circuits is imminent. However, quality of these circuits is highly in demand currently due to increase in the need of secure systems [3]. Fundamentally, the problem lies on the inherent contradiction between testability and security for digital circuits. Hence, there’s a need for an efficient solution such that both testability and security are satisfied.

To solve this challenging problem, different approaches have been proposed [4]-[13]. All the approaches except [13] 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. To resolve those disadvantages, we have reported another 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 [14]-[17] and “SR-quasi-equivalents” [18]. The proposed approach is only to replace part of the original scan chains to SR-equivalents or SR-quasi-equivalents, which satisfy both testability and security of digital 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 showed inversion-inserted linear feed-forward shift registers (I$^2$LF$^2$SR, for short) are useful circuits for the secure and testable scan design.

In this paper, we introduce a new class of extended shift registers called generalized feed-forward shift registers (GF$^2$SR, for short) by relaxing the condition of the SR-equivalents and SR-quasi-equivalents. GF$^2$SR is an extension of I$^2$LF$^2$SR and the class is much wider than that of I$^2$LF$^2$SR. Since the cardinality of the class of GF$^2$SR is much larger than that of I$^2$LF$^2$SR, the security level of scan design with GF$^2$SR is much higher than that of I$^2$LF$^2$SR. We consider how to control/observe GF$^2$SR to guarantee easy scan-in/out operations, i.e., state-justification and state-identification problems are considered. A program called WAGSR (Web Application for Generalized feed-forward Shift Registers) is presented to solve those problems.

2. EXTENDED SHIFT REGISTERS

In our previous works [14]-[18], we introduced extended shift registers to organize secure and testable scan design. Figure 1 shows those circuits realized by a linear feed-forward shift register and/or by inserting inverters; inversion-inserted SR (I$^2$SR), linear feed-forward SR (LF$^2$SR) and inversion-inserted linear feed-forward SR (I$^2$LF$^2$SR).

![Figure 1. Three types of extended shift registers](image-url)

**Figure 1.** Three types of extended shift registers
Consider a 3-stage $\tilde{1}LF^2SR$, $R_1$, given in Figure 2(a). By using symbolic simulation, we can obtain a output sequence ($z(t)$, $z(t+1)$, $z(t+2)$, $z(t+3)$) and the output $z(t+3)=x(t)\oplus 1\oplus x(t+2)$ as shown in Figure 2(b). So, we can see the input value applied to x at any time t appears at output z after 3 clock cycles with exclusive-OR of some inputs and/or constant 1. By using 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 Figure 2(c). Similarly, as illustrated in Figure 2(d), we can derive equations to determine uniquely the initial state ($y_1(t)$, $y_2(t)$, $y_3(t)$) from the input/output sequence.

More generally, for any circuit C of $\tilde{1}SR$, $LF^2SR$, and $\tilde{1}LF^2SR$ with k flip-flops, the input value applied to input x at any time t appears at output z after k clock cycles with exclusive-OR of some inputs and/or constant 1, i.e.,

$$z(t+k) = x(t) \oplus c_0 \oplus c_1 x(t+1) \oplus c_2 x(t+2) \oplus \ldots \oplus c_{k} x(t+k)$$

where $c_0$, $c_1$, $c_2$, ..., $c_k$ are 0 or 1. The ordered set of coefficients ($c_0$, $c_1$, $c_2$, ..., $c_k$) is called the characteristic coefficient of the circuit C.

Further, generally as for any circuit C of $\tilde{1}SR$, $LF^2SR$, and $\tilde{1}LF^2SR$ with k flip-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, where k is the number of flip-flops.

Here, let us extend the class of $\tilde{1}LF^2SR$ by relaxing linear functions in the above equation to arbitrary logic functions, i.e., the input value applied to x at any time t appears at z after k clock cycles with exclusive-OR of some logic function f of $x(t+1)$, $x(t+2)$, ..., $x(t+k)$, as follows.

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

Figure 3. Generalized feed-forward shift register (GF$^2$SR)

A circuit of the structure shown in Figure 3 is called a generalized feed-forward shift register (GF$^2$SR). In this figure, $f_0$, $f_1$, ..., $f_k$ are arbitrary logic functions of input x and state variables $y_i$ of preceding states. $f_k$ is a constant function, $f_1$ is a function of $x$, $f_2$ is a function of x and $y_1$, and $f_i$ is a function of $x$, $y_1$, $y_2$, ..., $y_{k-1}$. It can be shown that, for any GF$^2$SR with k flip-flops, the output z at time t+k behaves in accordance with the above equation.

By using symbolic simulation, we can obtain an output sequence ($z(t)$, $z(t+1)$, $z(t+2)$, $z(t+3)$) and the output $z(t+3)=x(t) \oplus x(t+2) \oplus x(t+1)$ as shown in Figure 4(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 $R_2$ from any state to the desired final state ($y_1(t+3)$, $y_2(t+3)$, $y_3(t+3)$) as illustrated in Figure 4(b). Similarly, as illustrated in Figure 4(b), we can derive equations to determine uniquely...
the initial state \((y_1(t), y_2(t), y_3(t))\) from the input/output sequence.

3. How to Control/Observed GF\(^2\)SR

For an extended shift register, the following two problems are important in order to utilize the extended shift register as a scan shift register in testing. One problem is to generate an input sequence to transfer the circuit into a given desired state. This is called state-justification problem. The other problem is to determine the initial state by observing the output sequence from the state. This is called state-identification problem.

We have shown in the previous section that, for I\(^2\)LF\(^2\)SR, R\(_1\), and GF\(^2\)SR, R\(_2\), we can derive equations to obtain an input sequence that transfers \(R_1\) and \(R_2\) from any state to the desired final state as illustrated in Figure 2(c) and Figure 4(b), respectively. Similarly, as illustrated in Figure 2(d) and Figure 4(b), we can derive equations to determine uniquely the initial state from the input/output sequence.

This holds for any circuit \(C\) in the class of I\(^2\)LF\(^2\)SR and GF\(^2\)SR, i.e., (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 \(C\), where \(k\) is the number of flip-flops.

Consider three different structured 3-stage GF\(^2\)SRs, R\(_2\), R\(_3\) and R\(_4\), shown in Figures 4, 5 and 6. From the results of symbolic simulation, we can see their outputs \(z(t+3)\) are the same, i.e.,

- \(z(t+3) = x(t+3) \oplus x(t+2) \cdot x(t+1)\)
- \(z(t+3) = x(t) \oplus x(t+1) \cdot x(t+2)\)
- \(z(t+3) = x(t+3) \oplus x(t+2) \cdot x(t+1)\)

(b) Symbolic simulation

Figure 4. Example of GF\(^2\)SR, \(R_2\)

![Figure 4](image)

<table>
<thead>
<tr>
<th>(x)</th>
<th>(y_1)</th>
<th>(y_2)</th>
<th>(y_3)</th>
<th>(z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x(t))</td>
<td>(y_1(t))</td>
<td>(y_2(t))</td>
<td>(y_3(t))</td>
<td>(z(t) = y_3(t))</td>
</tr>
<tr>
<td>(x(t))</td>
<td>(y_1(t))</td>
<td>(y_2(t))</td>
<td>(y_3(t))</td>
<td>(z(t) = x(t) \oplus y_1(t) \oplus y_2(t))</td>
</tr>
<tr>
<td>(x(t))</td>
<td>(y_1(t))</td>
<td>(y_2(t))</td>
<td>(y_3(t))</td>
<td>(z(t) = y_3(t))</td>
</tr>
</tbody>
</table>

(b) Symbolic simulation

Figure 5. Example of another GF\(^2\)SR, \(R_3\)

![Figure 5](image)

<table>
<thead>
<tr>
<th>(x)</th>
<th>(y_1)</th>
<th>(y_2)</th>
<th>(y_3)</th>
<th>(z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x(t))</td>
<td>(y_1(t))</td>
<td>(y_2(t))</td>
<td>(y_3(t))</td>
<td>(z(t) = y_3(t))</td>
</tr>
<tr>
<td>(x(t))</td>
<td>(y_1(t))</td>
<td>(y_2(t))</td>
<td>(y_3(t))</td>
<td>(z(t) = x(t) \oplus y_1(t) \oplus y_2(t))</td>
</tr>
<tr>
<td>(x(t))</td>
<td>(y_1(t))</td>
<td>(y_2(t))</td>
<td>(y_3(t))</td>
<td>(z(t) = y_3(t))</td>
</tr>
</tbody>
</table>

(b) Symbolic simulation

Figure 6. Example of another GF\(^2\)SR, \(R_4\)

![Figure 6](image)
4. CARDINALITY OF EACH CLASS OF EXTENDED SRs

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 does not know the structure of modified scan chains.

Based on the above assumption, we consider the security to prevent scan-based attacks.

The security level of the secure scan architecture based on GF²SR is determined by the probability that an attacker can identify the structure of the GF²SR used in the scan design, and hence the attack probability approximates to the reciprocal of the cardinality of the class of GF²SR.

In [15, 18] we showed the cardinality of each class of linear structured circuits (I²SR, LF²SR, and ILF²SR) which is summarized in Table I. Obviously, the class of GF²SR covers I²SR, LF²SR, and ILF²SR. So, we have the covering relation as shown in Figure 7.

Let us calculate the number of circuits in the class of GF²SR. Let \( f_0, f_1, \ldots, f_k \) be the functions shown in Figure 3. The number of functions for each \( f_0, f_1, \ldots, f_k \) are \( 2^{2^n} = 2, 2^{2^1} = 4, \ldots, \) and \( 2^{2^k} \), respectively. Hence the total number of \( k \)-stage GF²SR is \( 2 \times 4 \times \cdots \times 2^{2^k} = 2^{(2^{k+1})-1} \). The summary of the cardinality of each class is shown in Table I. From this table, we can see the cardinality of GF²SR is much larger than that of ILF²SR, and hence very secure. For any GF²SR, the state-justification and state-identification problems can be easily solved, and hence we can use any of them to organize the secure and testable scan circuits.

<table>
<thead>
<tr>
<th>CLASS</th>
<th># OF CIRCUITS IN THE CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>I²SR</td>
<td>( 2^{k+1} - 1 )</td>
</tr>
<tr>
<td>LF²SR</td>
<td>( 2^{k(k+1)/2} - 1 )</td>
</tr>
<tr>
<td>I²LF²SR</td>
<td>((2^{k(k+1)/2} - 1)(2^{k+1} - 1))</td>
</tr>
<tr>
<td>GF²SR</td>
<td>( 2^{(2^{k+1})-1} )</td>
</tr>
</tbody>
</table>

5. PROGRAM WAGSR

WAGSR (Web Application for Generalized feed forward Shift Registers) is a web application program to compute/solve various problems on GF²SR by symbolic and logic simulation as follows.

1. Design of GF²SR by means of logic expression
2. Illustration of GF²SR
3. Computation for GF²SR to solve state-justification and state-identification problems
   - Symbolic simulation
   - Logic simulation by partially specifying values 0, 1, and/or X to input/output sequence, initial state, and/or final state.

WAGSR adopts GUI (graphical user interface) for expressing outcome by circuit diagram and table. SR-ID code is introduced to represent the structure of each type of extended shift register uniquely. In Appendix, some examples of the outcome by WAGSR are presented. For example, from Figure 11, we obtain an input sequence to transfer the circuit to all 1's state independently of the initial state. In Figure 12, we can identify the initial state from the input/output sequence.

6. CONCLUSION

In our previous work, we reported a secure and testable scan design approach by using extended shift registers called SR-equivalents [14-17] and SR-quasi-equivalents [18], where the class of I²LF²SR is one of the most useful class. In this paper, we introduced a further extended class of generalized feed-forward shift registers (GF²SR). GF²SR is an extension of I²LF²SR and the class is much wider than that of ILF²SR. Since the cardinality of the class of GF²SR is much larger than that of I²LF²SR, the security level of scan design with GF²SR is much higher than that of I²LF²SR. We considered how to control/observe GF²SR to guarantee easy scan-in/out operations, i.e., state-justification and state-identification problems were considered. A program called WAGSR (Web Application for Generalized feed-forward Shift Registers) that solves those problems was introduced.

REFERENCES

APPENDIX

Figure 8. Design of GF^2SR by means of logic expression
Figure 9. Structural information of GF^2SR

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR-ID</td>
<td>16F16FF16FF41;16F16FF12;</td>
</tr>
<tr>
<td># of '0'</td>
<td>0</td>
</tr>
<tr>
<td># of '1'</td>
<td>0</td>
</tr>
<tr>
<td># of AND Gates</td>
<td>1</td>
</tr>
<tr>
<td># of OR Gates</td>
<td>0</td>
</tr>
<tr>
<td># of XOR Gates</td>
<td>1</td>
</tr>
<tr>
<td># of Feed-Forward Connections</td>
<td>4</td>
</tr>
<tr>
<td># of Feedback Connections</td>
<td>0</td>
</tr>
<tr>
<td>is GFSR</td>
<td>yes</td>
</tr>
<tr>
<td>is LxSR</td>
<td>no</td>
</tr>
</tbody>
</table>

Figure 10. Symbolic simulation
Figure 11. State-justification

Figure 12. State-identification