PAPER

Generalized Feed Forward Shift Registers and Their Application to Secure Scan Design

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

SUMMARY In this paper, we introduce generalized feed-forward shift registers (GF^2SR) 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 that inversion-inserted linear feed-forward shift registers (I^2LF^2SR) are useful circuits for the secure and testable scan design. GF^2SR is an extension of I^2LF^2SR and the class is much wider than that of I^2LF^2SR. Since the cardinality of the class of GF^2SR is much larger than that of I^2LF^2SR, the security level of scan design with GF^2SR is much higher than that of I^2LF^2SR. We consider how to control/observe GF^2SR to guarantee easy scan-in/out operations, i.e., state-justification and state-identification problems are considered. Both scan-in and scan-out operations can be overlapped in the same way as the conventional scan testing, and hence the test sequence for the proposed scan design is of the same length as the conventional scan design. A program called WAGSR (Web Application for Generalized feed-forward Shift Registers) is presented to solve those problems.

key words: *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 [1], [2], they can also make 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, and thus a problem arises in testing these types of circuits. 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]–[14]. All the approaches except [11] 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. The approach of inserting inverters in scan chains [11] has a disadvantage in that the positions of inserted inverters can be determined by simply scanning out after resetting (to zero) all the flip-flops in the scan chain. Therefore, internal state can be identified and the security is breached.

The disadvantages of the previous works [4]–[10], [12]–[14] are high area overhead, timing overhead and performance degradation, and the disadvantage of the work [11] is the weakness from the reset-based attack. To resolve all those disadvantages, we have 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 [16]–[19] and *"SR-quasi-equivalents"* [20]. The proposed approach only replaces 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^2LF^2SR, for short) are useful circuits for the secure and testable scan design [20].

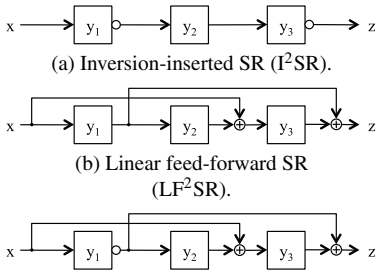
The objective application of secure and testable scan design is mainly to use it for cryptographic circuits though it can be used for IP protection and other purposes. In our proposed secure scan architecture, the scanned-out data from a scan register is not the same as the content of the scan register. Therefore, the attacker cannot obtain the content of the scan register, and hence existing scan-based attacks [6], [15] that depend on calculation from scanned data will fail, unless the attacker can identifies the configuration of the extended scan register.

In this paper, we introduce a new class of extended shift registers called *generalized feed-forward shift registers* (GF^2SR, for short) by relaxing the condition of the SR-equivalents and SR-quasi-equivalents. GF^2SR is an extension of I^2LF^2SR and the class is much wider than that of I^2LF^2SR. The security level of the secure scan architecture based on the extended shift registers like I^2LF^2SR and GF^2SR is determined by the probability that an attacker can correctly guess the configuration of the extended shift register used in the circuit, and hence the attack probability approximates to the reciprocal of the cardinality of the class of the extended shift registers. Since the cardinality of the class

Manuscript received October 10, 2012.
Manuscript revised December 26, 2012.

[†]The author is with the Graduate School of Engineering and Resource Science, Akita University, Akita-shi, 010–8502 Japan.
[††]The author is with the Faculty of Informatics, Osaka Gakuin University, Suita-shi, 564–8511 Japan.
a) E-mail: fujiwara@ie.akita-u.ac.jp

DOI: 10.1587/transinf.E96.D.1125



(a) Inversion-inserted SR (I^2SR).

(b) Linear feed-forward SR (LF^2SR).

(c) Inversion-inserted linear feed-forward SR (I^2LF^2SR).

Fig. 1 Three types of extended shift registers.

of GF^2SR is much larger than that of I^2LF^2SR, the security level of scan design with GF^2SR is much higher than that of I^2LF^2SR. We consider how to control/observe GF^2SR to guarantee easy scan-in/out operations, i.e., state-justification and state-identification problems are considered. Both scan-in and scan-out operations can be overlapped in the same way as the conventional scan testing and hence 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 extended shift register after ATPG. A program called WAGSR (Web Application for Generalized feed-forward Shift Registers) is presented to solve those problems.

2. Extended Shift Registers

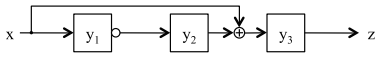
In our previous works [16]–[20], we introduced extended shift registers to organize secure and testable scan design. Figure 1 shows those circuits realized by a linear feed-forward shift register and/or by inserting inverters; inversion-inserted SR (I^2SR), linear feed-forward SR (LF^2SR) and inversion-inserted linear feed-forward SR (I^2LF^2SR).

Consider a 3-stage I^2LF^2SR, 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 $z(t+3) = x(t) \oplus 1 \oplus x(t+2)$ as shown in Fig. 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 Fig. 2 (c). Similarly, as illustrated in Fig. 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 I^2SR, LF^2SR, and I^2LF^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.,

$$z(t+k) = x(t) \oplus c_0 \oplus c_1 x(t+1) \oplus c_2 x(t+2) \oplus \cdots \oplus c_k x(t+k)$$

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



(a) I^2LF^2SR, R_1.

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

(b) Symbolic simulation.

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

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

(c) Equations for state-justification.

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

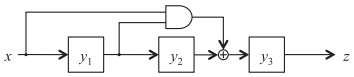
$$y_1(t) = z(t+2) \oplus x(t+1) \oplus 1$$
$$y_2(t) = z(t+1) \oplus x(t)$$
$$y_3(t) = z(t)$$

(d) Equations for state-identification.

Fig. 2 Example of I^2LF^2SR, R_1.

Further, generally as for any circuit C of I^2SR, LF^2SR, and I^2LF^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, we extend the class of I^2LF^2SR by relaxing linear



(a) GF^{2}SR, R_2.

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

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

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

(b) Symbolic simulation.

Fig. 4 Example of GF^{2}SR, R_2.

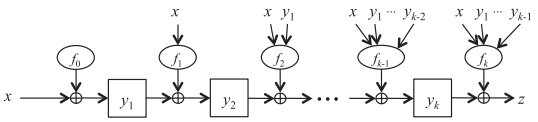


Fig. 3 Generalized feed-forward shift register (GF^{2}SR).

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 $x(t+1), x(t+2), \cdots, x(t+k)$, as follows.

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

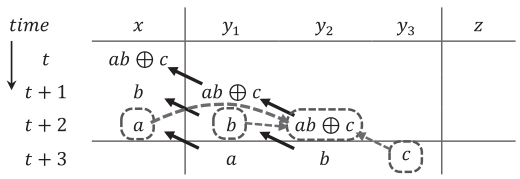
A circuit of the structure shown in Fig. 3 is called a *generalized feed-forward shift register (GF^{2}SR)*. In this figure, $f_0, f_1, \cdots, f_k$ are arbitrary logic functions of input x and state variables y_i of preceding stages. f_0 is a constant function, f_1 is a function of x, f_2 is a function of x and y_1, and f_i is a function of $x, y_1, y_2, \cdots, y_{i-1}$. 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 $z(t+3) = x(t) \oplus x(t+2)x(t+1)$ as shown in Fig. 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 Fig. 4 (b). Similarly, as illustrated in Fig. 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.

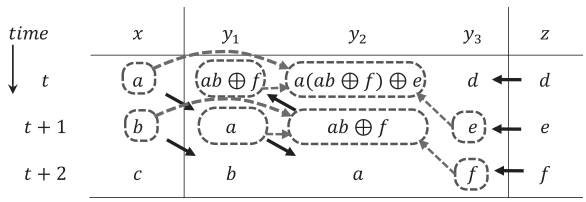
3. How to Control/Observe GF^{2}SR

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



(a) How to derive transfer sequence for final state.



(b) How to identify the initial state from input/output sequence.

Fig. 5 How to control/observe GF^{2}SR, R_2.

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 Fig. 2 (c) and Fig. 4 (b), respectively. Similarly, as illustrated in Fig. 2 (d) and Fig. 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.*

In Fig. 2 and Fig. 4, we showed how to derive transfer sequence and how to identify the initial state from input/output sequence by means of symbolic simulation. However, it is hard to derive those equations and to solve the solutions if the size of registers becomes large. As an alternative method, we can derive transfer sequence and identify the initial state by means of logic simulation instead of symbolic simulation. Figure 5 illustrates the method applied to GF^{2}SR, R_2. In Fig. 5 (a), given a final state $(y_1(t+3) = a, y_2(t+3) = b, y_3(t+3) = c)$, all other val-

ues can be uniquely derived only by implication operation from (a, b, c). For example, $y_1(t+3) = a$ implies $x(t+2) = a$ and $y_2(t+3) = b$ implies $y_1(t+2) = b$. This type of direct implication is indicated by solid arrow. After that, those implied values $x(t+2) = a$ and $y_1(t+2) = b$ with $y_3(t+3) = c$ imply $y_2(t+2) = ab \oplus c$. This implication is indirect implication or implied from more than two values, and is indicated by dotted arrows. In Fig. 5 (b), given input sequence (a, b, c) and output sequence (d, e, f), then all other values can be uniquely derived only by implication operation. For example, $y_1(t+1) = a$ is implied from $x(t) = a$. $y_2(t+1) = ab \oplus f$ is implied from $x(t+1) = b$, $y_1(t+1) = a$, and $y_3(t+2) = f$. Further, $y_1(t) = ab \oplus f$ is implied from $y_2(t+2) = ab \oplus f$. This method based on logic simulation using only implication operation is very fast and effective for very large scale of real scan chains. We have made a program to solve those problems, which is presented in the following section.

From the above observation, for the class of I^2LF^2SR and GF^2SR, 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 extended shift register after ATPG.

4. Program WAGSR

WAGSR (Web Application for Generalized feed forward Shift Registers) is a web application program to compute/solve various problems on GF^2SR by symbolic and logic simulation as follows.

1. Design of GF^2SR by means of logic expression
2. Illustration of GF^2SR
3. Computation for GF^2SR 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. Figure A· 1 shows a window for designing GF^2SR. After entering the necessary information for the design such as the number of flip-flops and logic expressions in JavaScript form for flip-flops, the circuit diagram is generated. Figure A· 2 shows the structural information of designed GF^2SR. Figure A· 3 shows the outcome of symbolic simulation. Figure A· 4 and Fig. A· 5 illustrates the outcomes of logic simulation. From Fig. A· 4, we obtain an input sequence to transfer the circuit to all 1's state independently of the initial state. In Fig. A· 5, we can identify the initial state from the input/output sequence.

For several GF^2SR circuits of 16 bits, 32 bits, 64 bits, and 64 +16 bits size, we measured the computation time both for generating logic expressions by symbolic simulation (1st stage) and for generating a transfer sequence from a given final state by logic simulation (2nd stage), using the web browser Safari6 on 1.6 GHz Intel Core 2 Duo machine with 4 GB memory. The average computation time at the 1st stage is 0.2 seconds, 2.6 seconds, and 512.3 seconds for GF^2SR circuits of 16 bits, 32 bits, and 64 bits size, respectively. The average computation time of the 2nd stage is 0.2 seconds, 1.3 seconds, and 336.0 seconds for GF^2SR circuits of 16 bits, 32 bits, and 64 bits size, respectively. However, for GF^2SR circuits of 64+16 bits size, WAGSR cannot complete the computation due to lack of memory. Although WAGSR is a web application program using JavaScript, it can deal with GF^2SR circuits of 64 bits size with less than several minutes even on a small machine.

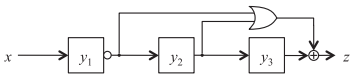
5. Cardinality of Each Class of Extended SRs

Our secure scan design through extended shift registers like GF^2SR provides both security and testability. With same effectiveness and efficiency of conventional scan design and with very minimal overhead, any digital circuit can be both easily testable and secure from attack.

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 may know the presence of test pins (scan in/out, scan, reset) of scan chains, but does not know any information inside of the circuit under consideration as well as the structure of the extended scan chains.* Based on this assumption, we consider the security to prevent scan-based attacks.

Consider three different structured 3-stage GF^2SRs, R_2, R_3 and R_4, shown in Fig. 4, Fig. 6 and Fig. 7. From the results of symbolic simulation, we can see their outputs $z(t+3)$ are the same, i.e., $z(t+3) = x(t) \oplus x(t+2)x(t+1)$. Therefore, their input/output behaviors after time t+3 are the same. Their input/output behaviors from time t to $t+2$ before $t+3$, become the same depending on their initial states. For example, R_2 with initial state $(y_1, y_2, y_3) = (0, 0, 0)$, R_3 with initial state $(0, 1, 1)$, and R_4 with initial state $(0, 0, 0)$ behave equivalently, i.e., their output sequences are the same for any input sequence. In this case, one cannot distinguish them. If one can initialize the circuit to a desired state, one may identify it from among three circuits. However, in our secure scan design, we protect the reset-based attack by adding one extra flip-flop to prohibit scan-after-reset operation [16], [19]. So, the attacker cannot initialize the circuit to a desired state, and hence cannot identify the structure of the circuit only from input/output behaviors.

Next, let us consider the security level by clarifying the cardinality of the class of GF^2SR's. The security level of the secure scan architecture based on GF^2SR is determined by the probability that an attacker can correctly guess the



(a) GF^{2}SR, R_3.

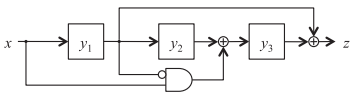
x	y_1	y_2	y_3	z
$x(t)$	$y_1(t)$	$y_2(t)$	$y_3(t)$	$z(t) = y_3(t) \oplus (\overline{y_1(t)} + y_2(t))$
$x(t+1)$	$x(t)$	$\overline{y_1(t)}$	$y_2(t)$	$z(t+1) = y_2(t) \oplus (\overline{x(t)} + y_1(t))$
$x(t+2)$	$x(t+1)$	$\overline{x(t)}$	$\overline{y_1(t)}$	$z(t+2) = \overline{y_1(t)} \oplus (\overline{x(t+1)}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

$x(t) = \overline{y_3(t+3)}$
$x(t+1) = \overline{y_2(t+3)}$
$x(t+2) = y_1(t+3)$

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

(b) Symbolic simulation.

Fig. 6 Example of GF^{2}SR, R_3.



(a) GF^{2}SR, R_4.

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

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

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

(b) Symbolic simulation.

Fig. 7 Example of GF^{2}SR, R_4.

Table 1 Cardinality of each class.

	# of circuits in the class
I^{2}SR	$2^{k+1} - 1$
LF^{2}SR	$2^{k(k+1)/2} - 1$
I^{2}LF^{2}SR	$(2^{k(k+1)/2} - 1)(2^{k+1} - 1)$
GF^{2}SR	$2^{(2^{k+1}-1)}$

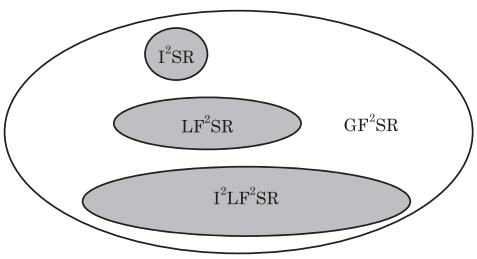


Fig. 8 Cover relation among classes.

structure of the GF^{2}SR used in the scan design, and hence the attack probability approximates to the reciprocal of the cardinality of the class of GF^{2}SR.

In [17], [20] we showed the cardinality of each class of linear structured circuits (I^{2}SR, LF^{2}SR, and I^{2}LF^{2}SR) which is summarized in Table 1. Obviously, the class of GF^{2}SR covers I^{2}SR, LF^{2}SR, and I^{2}LF^{2}SR. So, we have the covering relation as shown in Fig. 8.

Let us calculate the number of circuits in the class of GF^{2}SR. Let $f_0, f_1, \cdots, f_k$ be the functions shown in Fig. 3. The number of functions for each $f_0, f_1, \cdots$, and f_k are $2^{2^0} = 2, 2^{2^1} = 4, \cdots$, and 2^{2^k}, respectively. Hence the total number of k-stage GF^{2}SR 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 1. From this table, we can see the cardinality of GF^{2}SR is much larger than that of I^{2}LF^{2}SR, and hence very secure. For any GF^{2}SR, 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.

6. Application to Scan Design

A scan-designed circuit under consideration consists of a single or multiple scan chains and the remaining combinational logic circuit (*kernel*). A scan chain can be regarded as a circuit consisting of a shift register with multiplexers that select the normal data from the combinational logic circuit and the shifting data from the preceding flip-flop. Here, we replace the shift register with a GF^{2}SR.

However, to reduce the area overhead as much as possible, not all scan chains are replaced with extended scan chains. Only parts of scan chains necessary to be secure, e.g. secret registers, are replaced with GF^{2}SRs, and the size of the extended scan chains is large enough to make it secure. The delay overhead due to additional logic and Exclusive-

OR gates influences only scan operation, and hence there is no delay overhead for normal operation.

As mentioned in Sect. 3, testing can be done in the same way as the conventional scan testing. The length of test sequence is the same as the conventional scan design. There is no need to change traditional ATPG algorithm. There is no degradation in testability compared to the conventional scan design.

The scan design with embedded compactors seems to be secure, however, it is not secure if there exists a path such that the contents of a secret register leak out through part of the scan chain and the kernel (combinational circuit part) to primary outputs without passing through compactors. In this case, if we replace the secret register itself by an appropriate GF^2SR, it becomes secure.

7. Conclusion

In our previous work, we reported a secure and testable scan design approach by using extended shift registers called *SR-equivalents* [16]–[19] and *SR-quasi-equivalents* [20], where the class of I^2LF^2SR is one of the most useful class. In this paper, we introduced a further extended class of *generalized feed-forward shift registers* (GF^2SR). GF^2SR is an extension of I^2LF^2SR and the class is much wider than that of I^2LF^2SR. Since the cardinality of the class of GF^2SR is much larger than that of I^2LF^2SR, the security level of scan design with GF^2SR is much higher than that of I^2LF^2SR. We considered state-justification and state-identification problems for GF^2SR, i.e., how to control/observe GF^2SR to guarantee easy scan-in/out operations. Both scan-in and scan-out operations can be overlapped in the same way as the conventional scan testing, and hence 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 extended shift register after ATPG. A program called WAGSR (Web Application for Generalized feed-forward Shift Registers) that solves those problems was introduced.

References

[1] H. Fujiwara, Y. Nagao, T. Sasao, and K. Kinoshita, "Easily testable sequential machines with extra inputs," IEEE Trans. Comput., vol.24, no.8, pp.821–826, Aug. 1973,

[2] H. Fujiwara, Logic Testing and Design for Testability, The MIT Press 1985.

[3] K. Hafner, H. Ritter, T. Schwair, S. Wallstab, M. Deppermann, J. Gessner, S. Koesters, W. Moeller, and G. Sandweg, "Design and test of an integrated cryptochip," IEEE Des. Test Comput., pp.6–17, Dec. 1999.

[4] D. Hely, M.-L. Flottes, F. Bancel, B. Rouzeyre, and N. Berard, "Scan design and secure chip," 10th IEEE International On-Line Testing Symposium, pp.219–224, 2004.

[5] D. Hely, F. Bancel, M.L. Flottes, and B. Rouzeyre, "Securing scan control in crypto chips," J. Electronic Testing - Theory and Applications, vol.23, no.5, pp.457–464, Oct. 2007.

[6] B. Yang, K. Wu, and R. Karri, "Scan based side channel attack on dedicated hardware implementations of data encryptionstandard," International Test Conference 2004, pp.339–344, 2004.

[7] B. Yang, K. Wu, and R. Karri, "Secure scan: A design-for-test architecture for crypto chips," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.25, no.10, pp.2287–2293, Oct. 2006.

[8] J. Lee, M. Tehranipoor, and J. Plusquellic, "A low-cost solution for protecting IPs against scan-based side-channel attacks," 24th IEEE VLSI Test Symposium, pp.94–99, 2006.

[9] 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, Oct.-Dec. 2007.

[10] S. Paul, R.S. Chakraborty, and S. Bhunia, "VIm-Scan: A low overhead scan design approach for protection of secret key inscan-based secure chips," 25th IEEE VLSI Test Symposium, pp.455–460, 2007.

[11] G. Sengar, D. Mukhopadhyay, and D.R. Chowdhury, "Secured flipped scan-chain model for crypto-architecture," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.26, no.11, pp.2080–2084, Nov. 2007.

[12] M. Inoue, T. Yoneda, M. Hasegawa, and H. Fujiwara, "Partial scan approach for secret information protection," 14th IEEE European Test Symposium, pp.143–148, May 2009.

[13] U. Chandran and D. Zhao, "SS-KTC: A high-testability low-overhead scan architecture with multi-level security integration," 27th IEEE VLSI Test Symposium, pp.321–326, May 2009.

[14] M.A. Razzaq, V. Singh, and A. Singh, "SSTKR: Secure and testable scan design through test key randomization," 20th IEEE Asian Test Symposium, pp.60–65, Nov. 2011.

[15] R. Nara, N. Togawa, M. Yanagisawa, and T. Ohtsuki, "A Scan-Based Attack Based on Discriminators for AES Cryptosystems," IEICE Trans. Fundamentals, vol.E92-A, no.12, pp.3229–3237, Dec. 2009.

[16] H. Fujiwara and M.E.J. Obien, "Secure and testable scan design using extended de Brujin graph," 15th Asia and South Pacific Design Automation Conference, pp.413–418, Jan. 2010.

[17] K. Fujiwara, H. Fujiwara, M.E.J. Obien, and H. Tamamoto, "SREEP: Shift register equivalents enumeration and synthesis program for secure scan design," 13th IEEE International Symposium on Design and Diagnosis of Electronic Circuits and Systems, pp.193–196, April 2010.

[18] K. Fujiwara, H. Fujiwara, and H. Tamamoto, "SREEP-2: SR-equivalent Generator for Secure and Testable Scan Design," 11th IEEE Workshop on RTL and High Level Testing, pp.7–12, Dec. 2010.

[19] K. Fujiwara, H. Fujiwara, and H. Tamamoto, "Differential behavior equivalent classes of shift register equivalents for secure and testable scan design," IEICE Trans. Inf. & Syst., vol.E94-D, no.7, pp.1430–1439, July 2011.

[20] K. Fujiwara, H. Fujiwara, and H. Tamamoto, "SR-Quasi-Equivalents: Yet Another Approach to Secure and Testable Scan Design," 12th IEEE Workshop on RTL and High Level Testing, pp.77–82, Dec. 2011.

[21] SREEP: http://sreep.fujiwaralab.net

[22] WAGSR: http://wagsr.fujiwaralab.net

Appendix

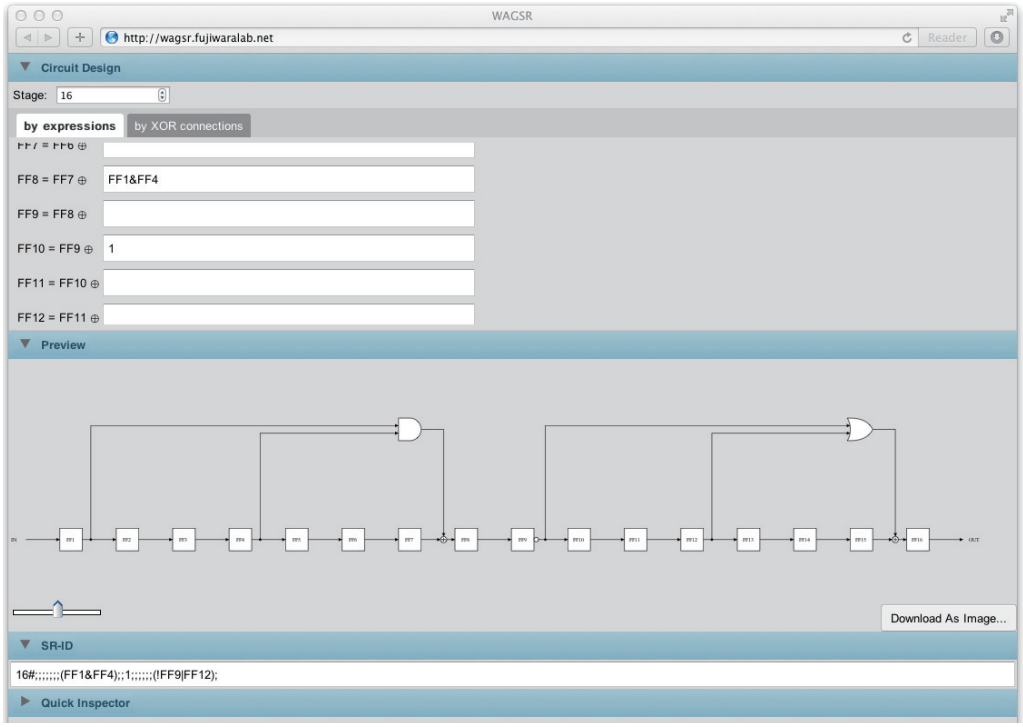


Fig. A· 1 Design of GF^{2}SR by means of logic expression.

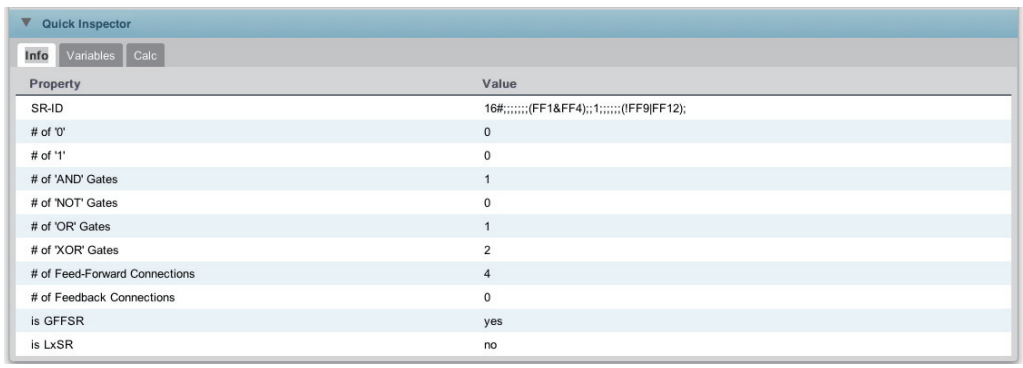


Fig. A· 2 Structural information of GF^{2}SR.

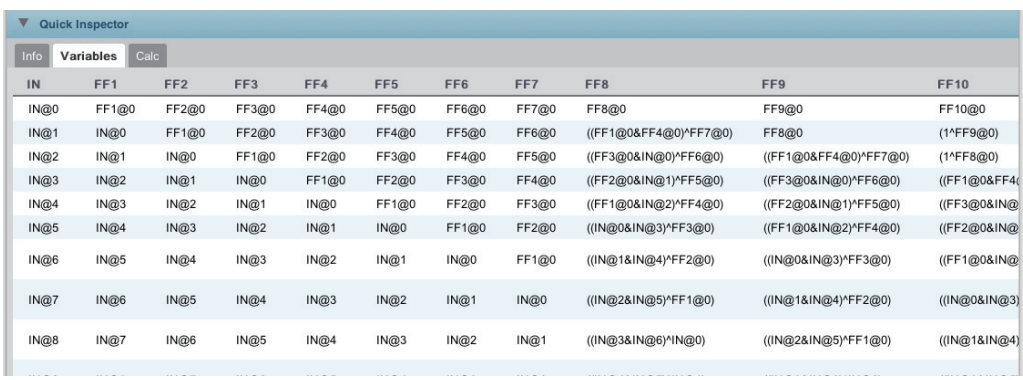
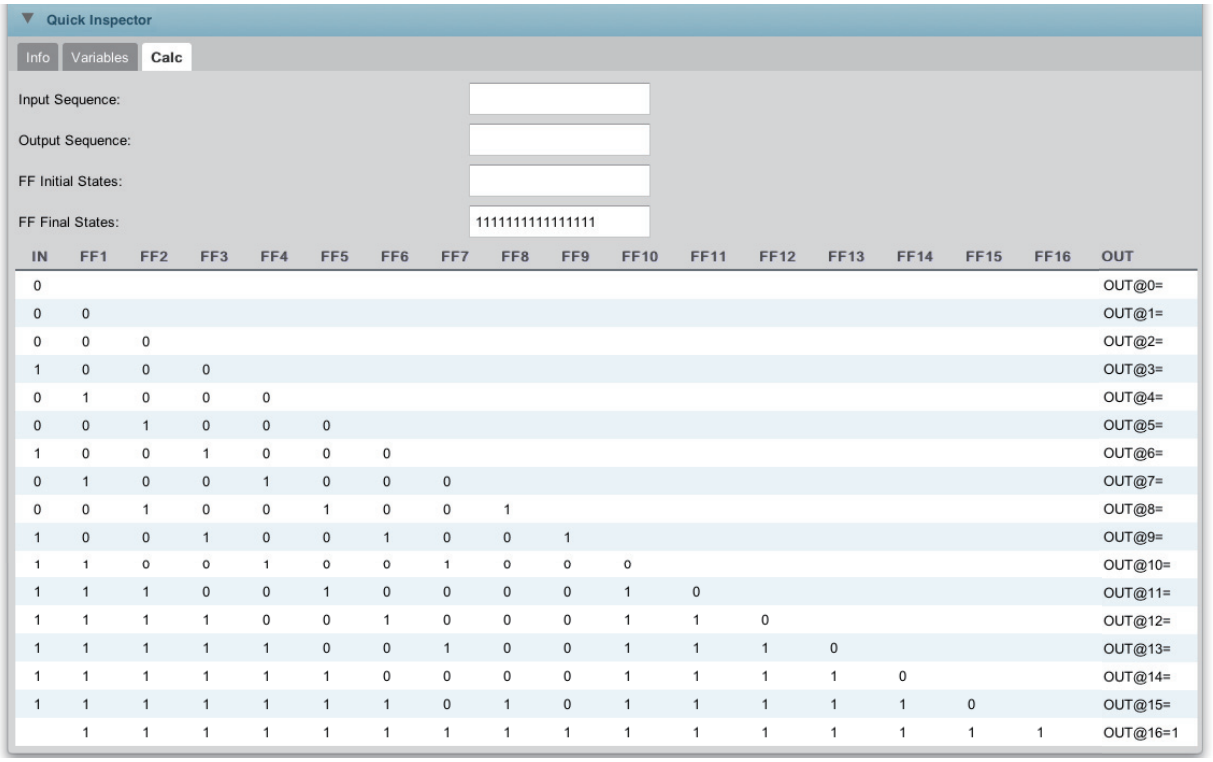


Fig. A· 3 Symbolic simulation.



Quick Inspector

Info | Variables | Calc

Input Sequence:

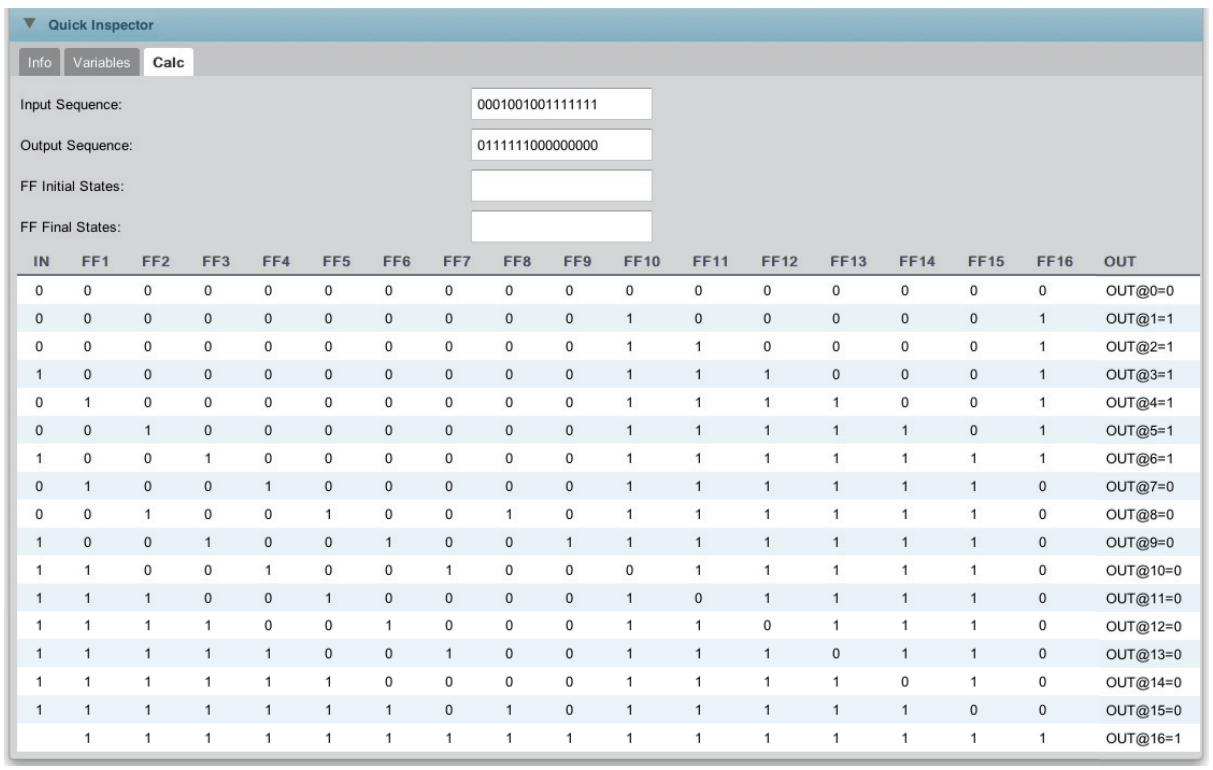
Output Sequence:

FF Initial States:

FF Final States: 1111111111111111

IN	FF1	FF2	FF3	FF4	FF5	FF6	FF7	FF8	FF9	FF10	FF11	FF12	FF13	FF14	FF15	FF16	OUT
0																	OUT@0=
0	0																OUT@1=
0	0	0															OUT@2=
1	0	0	0														OUT@3=
0	1	0	0	0													OUT@4=
0	0	1	0	0	0												OUT@5=
1	0	0	1	0	0	0											OUT@6=
0	1	0	0	1	0	0	0										OUT@7=
0	0	1	0	0	1	0	0	1									OUT@8=
1	0	0	1	0	0	1	0	0	1								OUT@9=
1	1	0	0	1	0	0	1	0	0	0							OUT@10=
1	1	1	0	0	1	0	0	0	0	1	0						OUT@11=
1	1	1	1	0	0	1	0	0	0	1	1	0					OUT@12=
1	1	1	1	1	0	0	1	0	0	1	1	1	0				OUT@13=
1	1	1	1	1	1	0	0	0	0	1	1	1	1	0			OUT@14=
1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	0		OUT@15=
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	OUT@16=1

Fig. A·4 Logic simulation.



Quick Inspector

Info | Variables | Calc

Input Sequence: 0001001001111111

Output Sequence: 0111111000000000

FF Initial States:

FF Final States:

IN	FF1	FF2	FF3	FF4	FF5	FF6	FF7	FF8	FF9	FF10	FF11	FF12	FF13	FF14	FF15	FF16	OUT
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OUT@0=0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	OUT@1=1
0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	OUT@2=1
1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	OUT@3=1
0	1	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	OUT@4=1
0	0	1	0	0	0	0	0	0	0	1	1	1	1	1	0	1	OUT@5=1
1	0	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1	OUT@6=1
0	1	0	0	1	0	0	0	0	0	1	1	1	1	1	1	0	OUT@7=0
0	0	1	0	0	1	0	0	1	0	1	1	1	1	1	1	0	OUT@8=0
1	0	0	1	0	0	1	0	0	1	1	1	1	1	1	1	0	OUT@9=0
1	1	0	0	1	0	0	1	0	0	0	1	1	1	1	1	0	OUT@10=0
1	1	1	0	0	1	0	0	0	0	1	0	1	1	1	1	0	OUT@11=0
1	1	1	1	0	0	1	0	0	0	1	1	0	1	1	1	0	OUT@12=0
1	1	1	1	1	0	0	1	0	0	1	1	1	0	1	1	0	OUT@13=0
1	1	1	1	1	1	0	0	0	0	1	1	1	1	0	1	0	OUT@14=0
1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	0	0	OUT@15=0
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	OUT@16=1

Fig. A·5 Logic simulation.

Katsuya Fujiwara received the B.E., the M.E., and the Ph.D. degrees in Engineering from Meiji University, Tokyo, Japan, in 1997, 1999, and 2002, respectively. He joined Akita University, Akita, Japan in 2002. Presently he is a Assistant Professor with the Department of Computer Science and Engineering, Akita University. His research interests are software engineering and network software. He is a member of the IPSJ, the JSSST and the IEEE Computer Society.

Hideo Fujiwara received the B.E., M.E., and Ph.D. degrees in electronic engineering from Osaka University, Osaka, Japan, in 1969, 1971, and 1974, respectively. He was with Osaka University from 1974 to 1985, Meiji University from 1985 to 1993, Nara Institute of Science and Technology (NAIST) from 1993 to 2011, and joined Osaka Gakuin University in 2011. Presently he is Professor Emeritus of NAIST and a Professor at the Faculty of Informatics, Osaka Gakuin University, Osaka, Japan. His research interests are logic design, digital systems design and test, VLSI CAD and fault tolerant computing, including high-level/logic synthesis for testability, test synthesis, design for testability, built-in self-test, test pattern generation, parallel processing, and computational complexity. He has published over 400 papers in refereed journals and conferences, and nine books including the book from the MIT Press (1985) entitled "Logic Testing and Design for Testability." He received the IECE Young Engineer Award in 1977, IEEE Computer Society Certificate of Appreciation Awards in 1991, 2000 and 2001, Okawa Prize for Publication in 1994, IEEE Computer Society Meritorious Service Awards in 1996 and 2005, IEEE Computer Society Continuing Service Award in 2005, and IEEE Computer Society Outstanding Contribution Award in 2001 and 2009. Dr. Fujiwara is a life fellow of the IEEE, a Golden Core member of the IEEE Computer Society, a fellow of the IPSJ (the Information Processing Society of Japan).