Random Switching Logic: A Countermeasure against DPA based on Transition Probability

Daisuke Suzuki[1], Minoru Saeki[1], and Tetsuya Ichikawa[2]

[1] Mitsubishi Electric Corporation, Information Technology R&D Center,
{dice, rebecca}@iss.isl.melco.co.jp
[2] Mitsubishi Electric Engineering Company Limited, Kamakura Office,
ichikawa@kam.mee.co.jp

Abstract. In this paper, we propose a new model for directly evaluating DPA leakage from logic information in CMOS circuits. This model is based on the transition probability for each gate, and is naturally applicable to various actual devices for simulating power analysis. We also report on our study of the effects of the previously known countermeasures on both our model and FPGA, and show the possibility of leaking information, which is caused by strict precondition for implementing a secure circuit. Furthermore, we present an efficient countermeasure, *Random Switching Logic*(RSL), for relaxing the precondition, and show that RSL makes a cryptographic circuit secure through evaluation on both our model and FPGA.

1 Introduction

SPA(Simple Power Analysis) and DPA(Differential Power Analysis), proposed by P.Kocher, have become a threat to the security of cryptographic implementation such as SmartCard [1][2][3]. Since these proposals, cryptographic researchers have begun to consider not only mathematical attacks but also side channel attacks. This work has resulted in many proposed countermeasures, especially against DPA. These countermeasures can roughly be divided into the following two groups:

- Algorithmic level
- Circuit level

Clavier, Coron and Messerges[4][5][6] deal with countermeasures for public key encryption algorithms. Using masked data with random numbers, Akkar and Coron, use countermeasures for block ciphers [7] [8]. We view all of the examples mentioned above as algorithmic. On the other hand, SABL (Sense Amplifier Based Logic)[9][10] based on DCVSL (Differential Cascode Voltage Switch Logic), SDDL (Simple Dynamic Differential Logic) based on CMOS circuit using the methodology of SABL, and WDDL(Wave Dynamic Differential Logic)[11] belong to circuit level.

Generally, ASICs, such as micro-processors and cryptographic co-processors, are implemented based on CMOS technology. We believe that countermeasures at the circuit level, such as WDDL and Masked-AND[12], are the most fundamental techniques because these techniques are related to power consumption and are applicable to various cryptographic algorithms.

What is important is how to show the effectiveness of a countermeasure. In this paper, we begin by considering a methodology for security evaluation of CMOS circuits. Some attempts already have been made to systematically analyze DPA leakage[14][15][16] Constructing a power consumption model is one effective method for analysis of the effects of countermeasures. For instance, the model based on analog characteristics of CMOS circuits[14], the model based on the Hamming weight[15], and the simplification model of Ref.[14] based on transition of data registers[16] were proposed in 1999, 2000 and 2002, respectively. Each model is complex or insufficient in regard to the reason why the leakage occurs, because the aim of the model is to simulate power consumption itself or to find bias of data, not bias of power consumption. We now present a new model that finds the origin of the leakage (see also [17]). This model is based on signal transition probability for each gate, and is not only more accurate than the digital model Ref.[15], but is more easily applied than the analog model Refs.[14],[16]. We also will point out that evaluation results of some primitive logics using our model are very similar to actual power analysis on FPGA.

We next discuss the relation between security strength and feasibility of implementation for some previously known countermeasures. As a result, we show that logic designs of those circuits are difficult to keep secure without very strict constraints for logic synthesis(see also [18]).

Finally, we propose the new countermeasure, *Random Switching Logic*(RSL), which is more efficient and more secure than current countermeasures (see also [19]). And we show that RSL makes a cryptographic circuit secure through evaluation of both our model and FPGA.

2 Previous Work

There are two approaches to the construction of countermeasures at the circuit level. The first approach uses complementary behavior and makes power consumption independent of data. The second uses data masking in combinational circuits and makes intermediate data unpredictable. In this section, we review a typical example based on each approach.

2.1 Wave Dynamic Differential Logic[11]

Tiri et al. proposed SABL[9] based on dynamic and differential logic. SABL is efficient due to the fact that power consumption is constant and independent of the signal. However, since standard CMOS libraries for implementation on ASIC including FPGA do not have SABL gates, SABL is not suitable for a current logic design system. Tiri et.al. then proposed SDDL using CMOS standard cell libraries based on the SABL. In addition, they presented WDDL, which optimized the function of precharge on the SDDL[11]. Fig.1(a) describes the basic components of WDDL.

As a first step WDDL executes precharge at the beginning of combinational logic. It also contains three logic gates, *i.e.*, AND, OR and NOT. In Ref.[11] they also proposed a method for the implementation of WDDL on FPGA.

2.2 Masked-AND Operation[12]

Figure 1(b) describes Masked-AND operation proposed by Trichina.

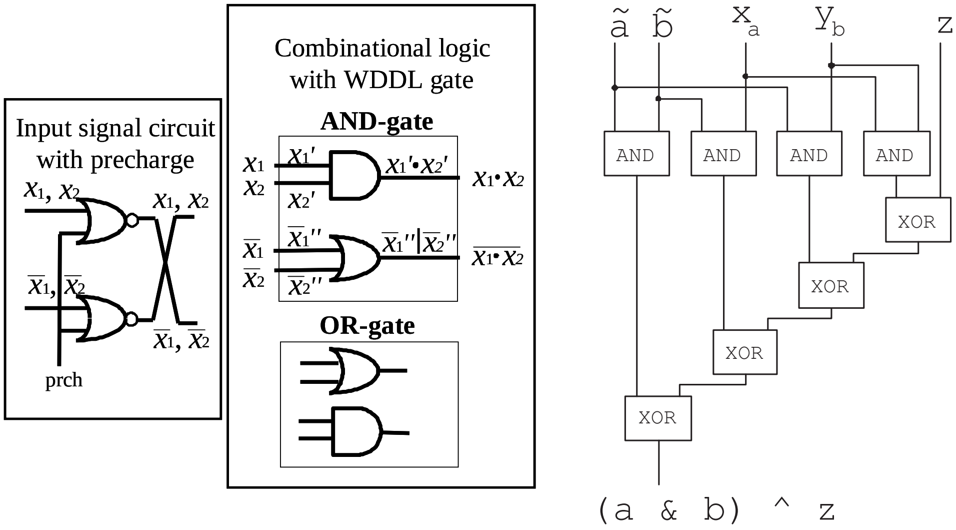
Let denote as follows;

- a and b are actual data.
- x_a, y_b and z are random data.(Each random data is an linearly independent.)
- "$\oplus$" and " $\cdot$ " mean eXclusive OR and AND, respectively.
- $\tilde{a} = a \oplus x_a$
- $\tilde{b} = b \oplus y_b$

Masked-AND is a method of calculating "$(a \cdot b) \oplus z$" using the above 5 input data, $\tilde{a}, \tilde{b}, x_a, y_b$ and z. Hence, the computations, as shown in Fig.1(b), can be carried out without compromising the bits of actual data. This is why Masked-AND is a countermeasure against DPA, according to Ref.[12].

3 Leakage Model on CMOS Circuit against DPA

A current evaluation model against DPA is constructed by simulating the power consumption of the circuit. In general, there are two approaches. One method constructs a detailed model of a characteristic of the analog device (see Refs.[14],[16]). In this case, the power consumption can be estimated with high accuracy. However, the estimation of the power consumption greatly depends on the device, and thus tends to become complex. The other method makes a rough estimate of power consumption assuming a certain digital model; for example, based on the Hamming weights[20]. In this approach it is possible to construct



(a) WDDL (b) Masked-AND

Fig. 1. Examples of current countermeasures

simple models and evaluate the power consumption without the device dependency. However, there is the possibility that the result might not accurately reflect the behavior of the actual device.

In this section, we propose a more detailed model that improves on the flipping model introduced in Ref.[21] for CMOS curcuits. Hereafter we call this model the *leakage model*. The basic idea of the model is to evaluate only the leakage information for DPA. Power consumption is not considered in this model.

3.1 Leakage Model Based on Transition Probability

Power consumption in CMOS circuits is summarized by the following equation[22].

$$P_{\mathrm{total}} = p_{\mathrm{t}} \cdot C_{\mathrm{L}} \cdot V_{\mathrm{dd}}^2 + p_{\mathrm{t}} \cdot I_{\mathrm{sc}} \cdot V_{\mathrm{dd}} \cdot f_{\mathrm{clk}} + I_{\mathrm{leakage}} \cdot V_{\mathrm{dd}}, \tag{1}$$

where C_{L} is loading capacitance, f_{clk} is the clock frequency, V_{dd} is the supply voltage, p_{t} is the transition probability of the signal, I_{sc} is the direct-path short circuit current, and I_{leakage} is the leakage current.

The first term is due to the charge/discharge of the loading capacitance. The second term depends on I_{sc}, which arises when both the NMOS and PMOS transistors are simultaneously active. The third term represents power consumption caused by the leakage current, which is mainly determined by characteristics of the CMOS process.

DPA is an attack in which the attacker estimates the intermediate value in the encryption/decryption process, classifies the patterns of power consumption based on this estimate, and obtains the secret information from the measured differences. Here, only p_{t} has the possibility of depending on the intermediate value in Eq.(1). Other parameters are fixed when the circuit is constructed. We take that the power difference in DPA measurements occurs because of biasing the transition probability of the signal according to the intermediate value. In the following we discuss the bias of the transition probability in detail.

In general, the transitions of the signals also depend on the delay in the transistors and the wiring in the CMOS device as well as the logic functions of the circuits. Thus we consider the leakage model in either of the following cases:

- *Static Model* : An ideal circuit with no delay in transistor and wiring.
- *Dynamic Model* : A real circuit where transient hazard is generated by the influence of the delay.

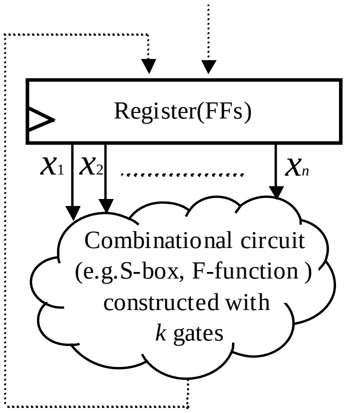


Fig. 2. General combinational circuit

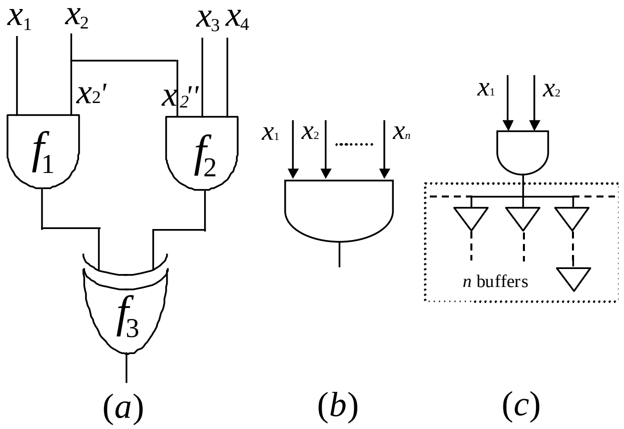


Fig. 3. Sample circuits (a) AND-XOR, (b) n-AND, (c) 2-AND with n-buffer

To clarify the discussion, we analyze the generalized circuit as shown in Fig.2. This circuit is constructed with k gates and n inputs x_1,x_2,$\cdots$,x_n and feedback paths from combinational circuit to registers. The transition of the output signal at the ith gate is

$$\Delta f_{(i)} = f_{(i)}(x_1 \oplus \Delta x_1, \cdots, x_n \oplus \Delta x_n) \oplus f_{(i)}(x_1, \cdots, x_n), \tag{2}$$

where Δx is a transition of the input signal, f_i is a Boolean function at the output of the ith gate. In what follows, we define the leakage model by considering bias of the probability of $\Delta f_{(i)} = 1$ in cases of either $\alpha = 0$ or $\alpha = 1$, with α being the value of the signal used by the attacker for grouping. We will call this signal a *selection bit*.

3.2 Static Leakage Model

We assume that x_1,x_2,$\cdots$,x_n in Fig.2 are independent variables[3]. In static models, the expectation of the transition frequency in one clock cycle is given by the following equation.

$$N_{\alpha}^{\mathrm{stc}} = \sum_{i=1}^{k} p_{\alpha,(i)}^{\mathrm{stc}}, \tag{3}$$

where $p_{\alpha,(i)}^{\mathrm{stc}}$ is the transition probability at the output of the ith gate corresponding to the value of selection bit α.

Definition 1. (Static Leakage) *Static Leakage $N_{\mathrm{diff}}^{\mathrm{stc}}$ in the combinational circuit is*

$$N_{\mathrm{diff}}^{\mathrm{stc}} = N_{\alpha=1}^{\mathrm{stc}} - N_{\alpha=0}^{\mathrm{stc}} \tag{4}$$

$$= \sum_{i=1}^{k} (p_{\alpha=1,(i)}^{\mathrm{stc}} - p_{\alpha=0,(i)}^{\mathrm{stc}}), \tag{5}$$

where $p_{x,(i)}^{\mathrm{stc}}$ is the transition probability of $\Delta f_{(i)} = 1$ under the condition that $\Delta x_1, \cdots, \Delta x_n$ are n independent variables.

If $N_{\mathrm{diff}}^{\mathrm{stc}} \neq 0$, there is a possibility that the correlation peak is observed in DPA measurements from Eq.(1). In general, a normal circuit using a CMOS standard cell library has $N_{\mathrm{diff}}^{\mathrm{stc}} \neq 0$. We show some examples below.

[3] These are not strictly independent, but any variation from independence is negligible when bias of the transition probability for each gate is discussed in a cryptographic circuit.

Example 1: (AND-XOR) We consider the static leakage of Fig.3(a) with random inputs. If selection bit is x_1, we get

$$\begin{aligned} \Delta f_{(1)} &= x_1 \cdot \Delta x_2 \oplus x_2 \cdot \Delta x_1 \oplus \Delta x_1 \cdot \Delta x_2, \\ \Delta f_{(2)} &= x_2 \cdot x_3 \cdot \Delta x_4 \oplus x_3 \cdot x_4 \cdot \Delta x_2 \oplus x_4 \cdot x_2 \cdot \Delta x_3 \oplus x_2 \cdot \Delta x_3 \cdot \Delta x_4 \\ &\quad \oplus x_3 \cdot \Delta x_4 \cdot \Delta x_2 \oplus x_4 \cdot \Delta x_2 \cdot \Delta x_3 \oplus \Delta x_2 \cdot \Delta x_3 \cdot \Delta x_4, \\ \Delta f_{(3)} &= \Delta f_{(1)} \oplus \Delta f_{(2)}. \end{aligned} \tag{6}$$

Namely,

$$\begin{aligned} \Delta f_{x_1=1,(1)} &= \Delta x_2 \oplus x_2 \cdot \Delta x_1 \oplus \Delta x_1 \cdot \Delta x_2, \\ \Delta f_{x_1=0,(1)} &= x_2 \cdot \Delta x_1 \oplus \Delta x_1 \cdot \Delta x_2. \end{aligned}$$

$x_i = 1$ and $\Delta x_i = 1$ occur with probability 1/2. Here, input states $(x_2, \Delta x_1, \Delta x_2)$ assumed to be $\Delta f_{x_1=1,(1)} = 1$ are (0,0,1),(1,0,1),(1,1,0) and (1,1,1). Hence, we have

$$p^{\text{stc}}_{x_1=1,(1)} = 1/2,\ p^{\text{stc}}_{x_1=0,(1)} = 1/4.$$

Similarly,

$$p^{\text{stc}}_{x_1=1,(2)} = 1/8,\ p^{\text{stc}}_{x_1=0,(2)} = 1/8,$$

$$p^{\text{stc}}_{x_1=1,(3)} = 7/16,\ p^{\text{stc}}_{x_1=0,(3)} = 5/16.$$

Thus, the static leakage of Fig.3(a) is

$$N^{\text{static}}_{\text{diff}} = 3/8.$$

AND-XOR being a basic element for S-boxes means that a normal implementation of a block cipher necessarily has static leakage.

Example 2: (n-AND) Under the condition similar to Example 1, static leakage of n-input AND shown in Fig.3(b) is

$$N^{\text{static}}_{\text{diff}} = (2^{n-1} - 1)/2^{2n-2},$$

where a selection bit $\alpha \in \{x_1, \cdots, x_n\}$.

Example 3: (Buffer Tree) The static leakage of two-input AND gates connected to n buffers (Fig.3(c)) is

$$N^{\text{static}}_{\text{diff}} = \frac{1}{4} \cdot n,$$

where a selection bit is x_1 or x_2 . Stated simply, static leakage at the gate with large fan-out is amplified.

From Definition 1, the static leakage has the following property.

Property 1: (Consecutive Static Leakage) *An equal amount of static leakage occurs both in the cycle when the selection bit appears and in the next cycle.*

From Eq.(2) it is obvious that the transitions related to the selection bit occur in the cycle when the selection bit appears and also in the next cycle. In cryptographic circuits, $\Delta x_1, \cdots, \Delta x_n$ are, in general, independent random variables. Thus, two static leakages of equal amounts occur for two consecutive cycles because two biased state transitions occur (random state → state depending on α, state depending on α → random state). This means that two similar DPA peaks are observed for two consecutive clock cycles in the DPA measurements if the target device is ideal.

3.3 Dynamic Leakage Model

In an actual cryptographic circuit, the delay time depends on the route of the signal. Thus each and tends to be non-uniform. Such non-uniformity is especially remarkable in the circuits designed with automatic synthesis/layout.

As in Section 3.2, we consider the transition probability in Fig.2. We assume that the transitions $\varDelta x_1, \cdots, \varDelta x_n$ of the registers reach each gate at a different time. Here, the transition of Boolean function $\varDelta f_{(i)}$ occurs only when transitions of the registers reach the ith gate. From these, we can evaluate the transition probability at a certain timing by supposing that only the transition corresponding to the timing is a variable and the others are 0. We define the *Dynamic Leakage* using this probability.

Defination 2. (Dynamic Leakage) *Let $\varDelta t$ be a time interval that an attacker can observe. Dynamic Leakage $N^{\mathrm{dyc}}_{\mathrm{diff}}$ in $\varDelta t$ on the combinational circuit is*

$$N^{\mathrm{dyc}}_{\mathrm{diff}} = N^{\mathrm{dyc}}_{\alpha=1} - N^{\mathrm{dyc}}_{\alpha=0} \tag{7}$$

$$= \sum_{i=1}^{k} \sum_{e \in E(i)} (p^{\mathrm{dyc}}_{\alpha=1,(i)}(e) - p^{\mathrm{dyc}}_{\alpha=0,(i)}(e)), \tag{8}$$

where $E(i)$ is the set of events with possibility that transition occurs in the state after α appeared at the ith gate in $\varDelta t$, $p^{\mathrm{dyc}}_{\alpha,(i)}(e)$ is the probability of $\varDelta f_{(i)} = 1$ under the condition that the transition of the input signal corresponding to e is a variable and the others are 0.

Here, we consider the relation between the transitions of the registers $\varDelta x_1, \cdots, \varDelta x_n$ and the event $e \in E(i)$ that depends on the selection bit α. If the circuit has not been redundantly constructed and $\varDelta t \geq 2$ cycles, $E(i)$ contains at least n events corresponding to transitions of the registers in the state that α appeared. This doesn't depend on the order of the signal transitions. Note that these events are distributed between two cycles according to the delay time, which was fixed when the circuit was constructed, for each signal to propagate. Additionally, there is the possibility that two or more transitions occur by the same transitions of the registers if each propagation route of those transitions is different. In this case, the transitions corresponding to each route are treated as independent variables in Eq.(2). In the following, we evaluate the dynamic leakage of Fig.3.

Example 4: (AND-XOR) We consider the circuit, shown in Fig.3(a), on the dynamic model. If $\varDelta t \geq 2$ cycles, we get

$$E(1) = \{e(\varDelta x_1), e(\varDelta x_2^{'})\}, \quad E(2) = \{e(\varDelta x_2^{''}), e(\varDelta x_3), e(\varDelta x_4)\},$$
$$E(3) = \{e(\varDelta x_1), e(\varDelta x_2^{'}), e(\varDelta x_2^{''}), e(\varDelta x_3), e(\varDelta x_4)\}.$$

From Eq.(6), $\varDelta f_3$ at each event is

$$\varDelta f_{(3)}(e(\varDelta x_1)) = x_2 \cdot \varDelta x_1, \quad \varDelta f_{(3)}(e(\varDelta x_2^{'})) = x_1 \cdot \varDelta x_2^{'}, \quad \varDelta f_{(3)}(e(\varDelta x_2^{''})) = x_3 \cdot x_4 \cdot \varDelta x_2^{''},$$
$$\varDelta f_{(3)}(e(\varDelta x_3)) = x_4 \cdot x_2 \cdot \varDelta x_3, \quad \varDelta f_{(3)}(e(\varDelta x_4)) = x_2 \cdot x_4 \cdot \varDelta x_3.$$

If x_1 is a selection bit, we have

$$p^{\mathrm{dyc}}_{x_1=1,(3)}(e(\varDelta x_2^{'})) = 1/2, \quad p^{\mathrm{dyc}}_{x_1=0,(3)}(e(\varDelta x_2^{'})) = 0.$$

In $\varDelta f_{(1)}$, we have similarly

$$p^{\mathrm{dyc}}_{x_1=1,(1)}(e(\varDelta x_2^{'})) = 1/2, \quad p^{\mathrm{dyc}}_{x_1=0,(1)}(e(\varDelta x_2^{'})) = 0.$$

The dynamic leakage of Fig.3(a) is

$$N^{\mathrm{dyc}}_{\mathrm{diff}} = 1.$$

Note that the difference between x_1 and $x_2^{'}$ at delay time determines the timing whereby dynamic leakage occurs in the circuit. $N^{\mathrm{dyc}}_{\mathrm{diff}}$ occurs at the cycle when the predicted x_1 appears if $x_2^{'}$ is slower than x_1, and it occurs at the next cycle if the opposite is true.

Example 5: (n-AND) Under the condition similar to Example 4, dynamic leakage, shown in Fig.3(b), is

$$N_{\mathrm{diff}}^{\mathrm{dyc}} = (n-1)/2^{n-1},$$

where $x \in \{ \ x_1 \ , \cdots , \ x_n \ \}$.

Finally, we describe a common property to static and dynamic leakage.

Property 2. (Complementary Leakage from AND- and OR-gate) *The static/dynamic leakages of an equal amount but opposite polarity occur from AND- and OR-gate(resp., NAND- and NOR-gate) respectively under the same input and delay time condition.*

This means that there is the possibility that the leakage of the whole circuit is counterbalanced. Actually, a countermeasure using this property has been proposed[23].

4 Security Evaluation for Complementary Logics

From Propery 2, a complementary logic has possibility of counterbalancing the leakage. WDDL is a method that refines this considering. We consider the circuit shown Fig.1(a). At the end of the precharge phase (prch = 0), all output signals of the WDDL gates are at 0. Therefore, the transitions for each gate in evaluation phase (prch = 1) are as follows.

$$\Delta f_{(1)} = x_1' \cdot x_2' = \Delta x_1' \cdot \Delta x_2', \quad \Delta f_{(2)} = \bar{x}_1'' \mid \bar{x}_2'' = \Delta x_1'' \cdot \Delta x_2'' \oplus \Delta x_1'' \oplus \Delta x_2'',$$

where

$$\Delta x_1' = \Delta x_1'', \ \Delta x_2' = \Delta x_2''.$$

Therefore, we obviously get $N_{\mathrm{diff}}^{\mathrm{stc}} = 0$ in the static model.

Next, $E(i)$ in dynamic model is as follows.

$$E(1) = \{e(\Delta x_1') \text{ or } e(\Delta x_2')\}, \quad E(2) = \{e(\Delta x_1''), e(\Delta x_2'')\}.$$

In the AND-gate, The transition occurs only by the later input transition. On the other hand, the transition can occur with both input transitions in OR-gate. From these, the transition probability corresponding to each event is

$$p_{x_1=1,(1)}^{\mathrm{dyc}}(e(\Delta x_2')) = 1/2, \quad p_{x_1=0,(1)}^{\mathrm{dyc}}(e(\Delta x_2')) = 0,$$

$$p_{x_1=1,(2)}^{\mathrm{dyc}}(e(\Delta x_1'')) = 0, \quad p_{x_1=0,(2)}^{\mathrm{dyc}}(e(\Delta x_1'')) = 1,$$

$$p_{x_1=1,(2)}^{\mathrm{dyc}}(e(\Delta x_2'')) = 1/2, \quad p_{x_1=0,(2)}^{\mathrm{dyc}}(e(\Delta x_2'')) = 0,$$

where the event concerning x_1 occurs faster than that of x_2. At this time, if Δt is long enough, we get $N_{\mathrm{diff}}^{\mathrm{dyc}} = 0$. However, if $e(\Delta x_2'')$ doesn't occur for the period of Δt, we have $N_{\mathrm{diff}}^{\mathrm{dyc}} = -1/2$. Fig.4 is shown that dynamic leakage occurs in complementary gates when some timings of the events are different.

The similar observation applies to other countermeasures using the complementary logic. From the consideration the condition to make complementary gates(logic) secure against DPA is

- input signals reach each complementary gate(logic) at the same time.

In general, this condition is hard to implement circuits. In particular, it is not guaranteed in the LSI designed by the automatic synthesis/layout.

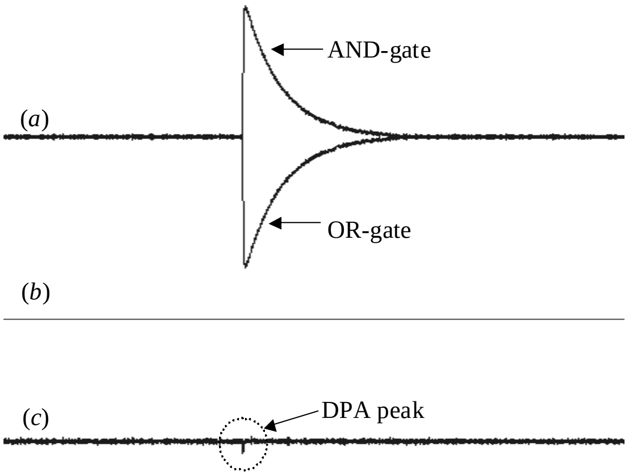


Fig. 4. Hypothesis of DPA trace for complementary gate (a) Actual DPA trace of AND(OR) operation on FPGA(see Section 6), (b) Hypothesis of DPA trace led from (a) under the secure conditions. (c) Hypothesis of DPA trace led from (a) under the different input delay time conditions.

5 Random Switching Logic

5.1 The Basic Idea of RSL

The countermeasures that equalize the signal transition frequency by complementary operations are dependent on wire length or fan-out. This often makes the design very difficult. To solve this problem, we propose a new countermeasure against DPA called *Random Switching Logic*(RSL). RSL does not require complementary operations. We start by considering the condition for single-rail CMOS circuits to be secure against DPA according to the leakage model shown in Section 3.

Definition 3. (Secure Single-rail CMOS Circuits) *Let X be a set of all predictable intermediates related to secret information. Let E_x be a set of all events for which signal transitions may occur in the circuits related to $\forall x \in X$. We define such circuits, that satisfy the following condition, as secure single-rail CMOS circuits from the viewpoint of transition probabilities.*

$$p_{1,(i)}(e_x) = p_{0,(i)}(e_x) \text{ for } 1 \leq \text{i} \leq \text{k}.$$

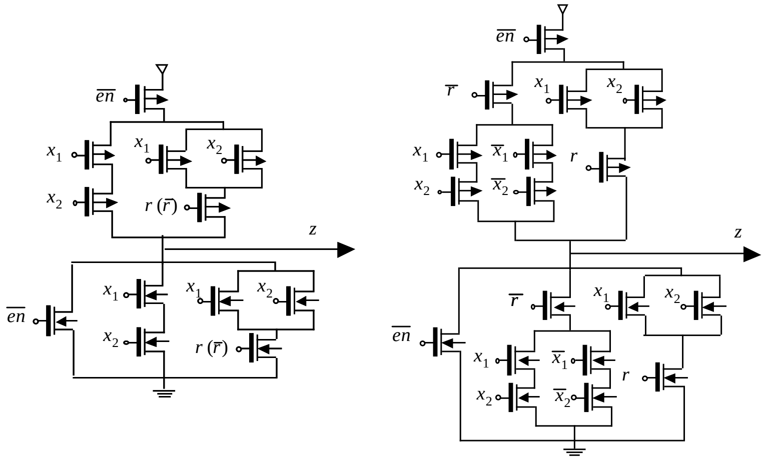
If this condition is satisfied, transition probability in the single-rail CMOS circuit is equalized. According to Eq.(5) and Eq.(7), we have $N_{\text{diff}} = 0$, if this condition is satisfied. And the single-rail CMOS circuits related to x are secure against DPA.

Single-rail circuits do not have timing problems as shown in Section 4. However, CMOS primitive gates(NAND,NOR, etc.) normally do not satisfy the above-mentioned condition. Thus, we propose *Random Switching Logic*(RSL), which processes original signals and an additional random signal simultaneously (see Fig.5). RSL has the following two properties.

I: RSL executes masked-operations for all input/output signals using the same 1bit random value.
II: RSL executes operations while enable signal(en) is 1, otherwise drives 0.

Fig.5(a) shows a 2-input NAND(NOR) RSL gate, and Fig.5(b) shows a 2-input XOR RSL gate.

Considering a stable state, signal transition probabilities return at a random signal's rate of change according to **I**. If input signals(x_1, x_2, r) arrive at the same time while the enable signal(en) is 1, the transition probability of the output signal is 1/2, which is independent of predictable signals(a or b), where $a = x_1 \oplus r$, $b = x_2 \oplus r$. However, transitions independent of a random signal may occur in masked-operations



(a) 2-input NAND(NOR) RSL gate (b) 2-input XOR RSL gate

Fig. 5. Examples of RSL-primitive gates

if transient hazards exist. To avoid such transitions, we adopted **II**. By raising the enable signals(en) to 1 after all input signals(x_1, x_2, r) arrive, transient hazards are suppressed. Therefore, the above-mentioned condition is satisfied and RSL based CMOS circuits can be secure against DPA.

Other than Fig.5, various complex gates for various functions which satisfy both properties can be constructed. Furthermore, RSL gates for any odd-number-input XOR/XNOR function does not need a random signal input if other input signals are already masked (in this case, only **II** is required).

As mentioned above, conditions for RSL to be secure against DPA are as follows.

- The transition of the random signal(r) is not biased.
- The enable signal(en) rises after all other input signals are fixed.

5.2 RSL Implementation on FPGA

In this section we consider implementing RSL on FPGA. Figure 6 is an example of implementation of an RSL equivalent circuit using LUT(Look Up Table) in FPGA. Hereafter, we call such RSL implementation RSLUT. As shown in Fig.6, RSLUT needs at least four input signals for a logical function. On the other hand, many FPGAs available on the market are composed of SRAM-based LUTs that have four input signals and one output signal. Therefore, as long as circuit designers know this, RSLUT can be implemented on almost all FPGAs. As for the power consumption of FPGA, the switching- matrixes that occupy the majority of the area are predominant. Therefore, if the transition probabilities of LUTs' I/O signals do not depend on data, the circuits are secure against DPA because the transition probabilities of CMOS inverters in the switching-matrixes between LUTs are equalized.

6 Experimental Results and Considerations

6.1 Security Evaluation using FPGA

We can easily evaluate the validity of the leakage model or countermeasures by using FPGA. In this section, we show experimental results of AES circuits implemented on FPGA. The evaluation environment is the general one shown in Table 1. An XCV1000-6-BG560C FPGA of Xilinx Inc.[24] is mounted on the target board.

Using the following four techniques we implemented and evaluated AES S-boxes. We adopted the method shown in Ref.[25] for the basic architecture of the S-box.

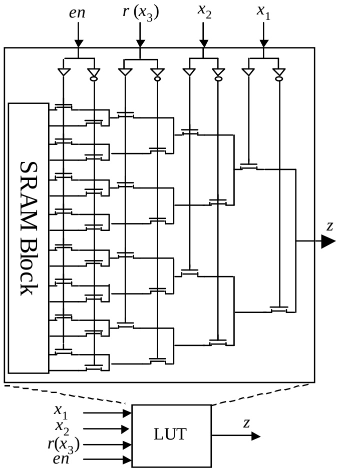


Fig. 6. Basic construction of RSLUT

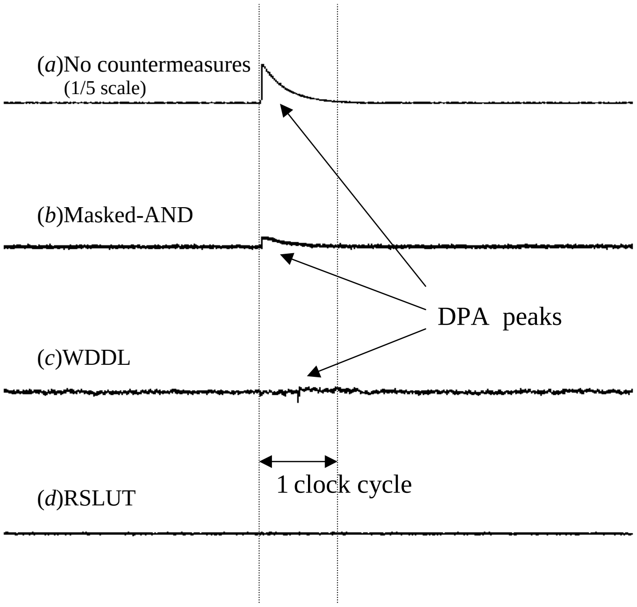


Fig. 7. Comparison of differential power traces (60000 samples)

Table 1. Evaluation environment

Design environment	
Language	Verilog-HDL
Simulator	Verilog-XL
Logical synthesis	Synplify version 7.3.4
Place and Route	ISE version 6.1.03i
Measurement environment	
Target FPGA	XCV1000-6-BG560C
Oscilloscope	Tektronix TDS 7104

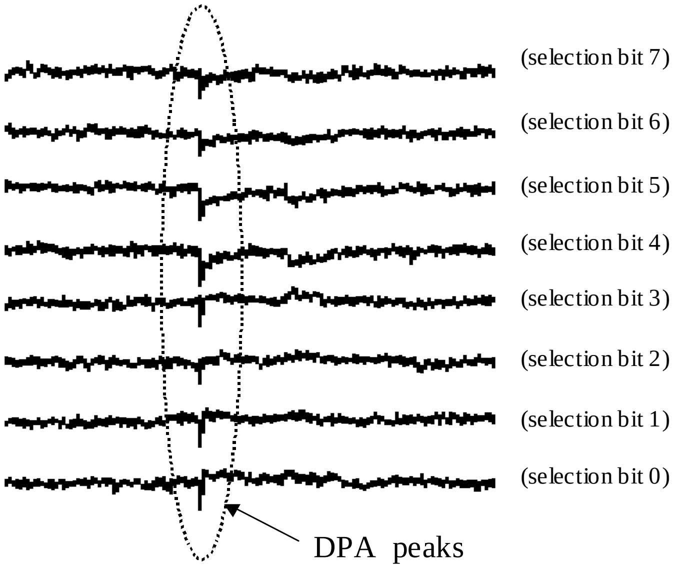


Fig. 8. Differential power traces of WDDL (60000 samples)

(a) No countermeasures
(b) Masked-AND[12]
(c) WDDL[11]
(d) RSLUT

Automatic place-and-route tools were used for all layout design. To meet the requirements in logical synthesis of each technique (b),(c) and (d), logical synthesis was controlled by using the tool's option.

Figure 7 shows differential power traces for each implementation. (The enlargement of each trace is shown in Fig.9 in Appendix A.) An input signal to multiplication in Galois field in a S-box[25] was used as a selection bit. As shown in Fig.7, obvious DPA peaks appear in (a) and (b). On the other hand, there seems to be no DPA peak in (c) and (d). But the trace of (c) looks like one we showed in Fig.4(c) in Section 4. This means that, in the case of WDDL, DPA peaks caused by timing differences may appear if automatic place-and-route tools are used. Figure 8 gives another piece of evidence that the small peak in (c) is DPA leakage, because similar peaks appear at the same time in the traces when looking at every bit of the same intermediate. In the case of RSLUT, there is no DPA peak as shown in Fig.7(d).

We also implemented and evaluated the entire AES circuit using RSLUT. As a result, RSLUT is quite effective as a countermeasure against DPA for FPGA. (See Appendix B.)

6.2 Implementation cost comparison

In this section, we discuss the implementation costs associated with area, performance, and difficulty. Table 2 shows area and the maximum propagation delay of an AES S-box using the above-mentioned four techniques. As for area, RSLUT is the smallest among the countermeasures. Masked-And needs both unmasking and masking for every AND operation, and it also needs linear transformation for random values. WDDL must have complementary pairs for all LUTs. Furthermore, WDDL can use only AND, OR, and NOT, so XOR operations tend to increase area. Although RSLUT needs additional control signals for all LUTs that execute AND operation, the required random signal is 1bit for the entire circuit, so the control circuit for random signal is unnecessary. And, because up to three input XOR operations can be executed in an LUT, the area overhead is relatively small. As for the propagation delay, Table 2 shows that RSLUT is the fastest. Next, we consider the conditions that every countermeasure requires to resist DPA. As for Masked-And it was shown in Fig. 1(b) that DPA is very difficult if the signal transitions according to random signals occur within a limited period of time. This condition is quite difficult to satisfy when using the general design environment. And Masked-And needs tens of bits of random signals per S-Box for every clock cycle. As for WDDL, it is necessary to equalize load capacity of every LUT pair that complementary switches. Generally, as shown before, this is quite difficult when using automatic place-and-route tools. As for RSL/RSLUT, control signals must change after all input data signals are fixed. This can be achieved easily even by automatic place-and-route tools if enough delay time is given. Therefore, we conclude that RSL/RSLUT is excellent in respect to efficiency and feasibility compared with other countermeasures against DPA.

Table 2. Implementation result of AES-Sbox by each countermeasure

Method	**Area[LUTs]**	**Critical path[ns]**
No countermeasures	86	20.68
Masked-AND	332	35.39
WDDL	456	46.80
RSLUT	174	30.35

7 Conclusion

In this paper, we proposed the leakage model of the CMOS device based on signal transition probabilities. And, we examined the generation mechanism of DPA leakage by using this model. Further, we proposed a new countermeasure technique against DPA called *Random Switching Logic*(RSL) that equalizes transition probabilities and suppresses transient hazards. RSL is sufficiently effective against DPA, and the condition for RSL to be secure against DPA is easy to implement even using automatic place-and-route tools.

Acknowledgement. We would like to thank Toyohiro Tsurumaru for helpful comments.

References

1. P. Kocher, “Timing attacks on implementations of Diffe-Hellmann, RSA, DSS, and other systems”, Proc. Advances in Cryptology - Crypto’96, LNCS 1109, pp. 104-113, 1996.
2. P. Kocher, J. Jaffe and B. Jun,“Differential power analysis,” In Advances in Cryptology - CRYPTO’99, LNCS 1666, pp. 388-397, 1999.
3. P. Kocher, J. Jaffe and B. Jun,“Introduction to Differential Power Analysis and Related Attacks”, http://www.cryptography.com/dpa/technical/index.html
4. C. Clavier and M. Joye, “Universal Exponentiation Algorithm - A First Step Towards Provable SPA Resistance,” CHES 2001, LNCS 2162, pp. 300-308, 2001.

5. J.-S. Coron, “Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems,” CHES’99, LNCS 1717, pp. 292-302, 1999.
6. T. Messerges, E. Dabbish and R. Sloan, “Power Analysis Attacks of Modular Exponentiation in Smartcards,” CHES’99, LNCS 1717, pp. 144-157, Springer-Verlag, 1999.
7. M.Akkar and C.Giraud, “An implementation of DES and AES, secure against some attacks,” CHES 2001, LNCS 2162, pp. 309-318, 2001.
8. J.-S. Coron and L. Goubin, “On Boolean and arithmetic masking against differential power analysis,” CHES 2000, LNCS 1965, pp. 231-237, 2000.
9. K.Tiri, M.Akmal and I.Verbauwhede, “A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on SmartCards,” Proc. Of 28th European Solid-State Circuits Conference, pp.403-406,2002.
10. K.Tiri and I.Verbauwhede, “Securing Encryption Algorithms against DPA at the Logic Level: Next Generation Smart Card Technology,” CHES 2003, LNCS 2779, p.125-136, 2003.
11. K. Tiri and I. Verbauwhede, “A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation,”in Proc. of Design Automation and Test in Europe Conference (DATE 2004), pp. 246-251,2004.
12. E.Trichina, “Combinational Logic Design for AES SubByte Transformation on Masked Data,” Cryptology ePrint Archive, 2003/236, http://eprint.iacr.org/complete/ .
13. E. Trichina, D. De Seta and L. Germani, “Simplified Adaptive Multiplicative Masking for AES and its secure implementation,” CHES 2002, LNCS 2523, pp. 187-197, 2002.
14. S.Chari, C.S.Jutla, J.R.Rao and P.Rohatgi, “Towards sound approaches to counteract poweranalysis attacks,” In Advances in Cryptology - - CRYPTO’99, LNCS 1666, pp. 398-412, 1999.
15. C. Clavier, J.-S. Coron and N. Dabbous, “Differential Power Analysis in the Presence of Hardware Countermeasures,” CHES 2000, LNCS 1965, pp. 252-263, 2000.
16. R. Bevan and E. Knudsen, “Ways to Enhance Differential Power Analysis,” ICISC 2002, LNCS 2587, pp. 327-342, 2003.
17. M.Saeki, D. Suzuki and T.Ichikawa, “Construction of DPA Leakage Model and Evaluation by Logic Simulation,” Technical Report ISEC2004-57, IEICE, 2004(in Japanese).
18. T.Ichikawa, D. Suzuki and M.Saeki, “An Attack on Cryptographic Hardware Design with Masking Method,” Technical Report ISEC2004-58, IEICE, 2004(in Japanese).
19. D. Suzuki, M.Saeki and T.Ichikawa, “Countermeasure against DPA Considering Transition Probabilities,” Technical Report ISEC2004-59, IEICE, 2004(in Japanese).
20. T. Messerges, “Using Second-Order Power Analysis to Attack DPA Resistant Software,” CHES 2000, LNCS 1965, pp. 238-251, 2000.
21. M.Akkar, R. Bevan, P. Dischamp and D. Moyart,“Power Analysis, What Is Now Posible...,” ASIACRYPTO 2000, LNCS 1976, pp. 489-502, 2000.
22. A.P. Chandrakasan, S. Sheng and R.W.Brodersen, “Low Power Digital CMOS Design,” IEEE Journal of Solid State Circuits, Vol.27, N0.4. pp. 473-484,1992.
23. Philips Electronics NV, “DATA CARRIER WITH OBSCURED POWER CONSUMPTIONI,” Patent, WO00/026746.
24. Xilinx, Inc., Data sheet “Virtex^{TM} 2.5 V Field Programmable Gate Arrays”, http://www.xilinx.com/
25. A. Satoh, S. Morioka, K. Takano and S. Munetoh, “A compact Rijndael hardware architecture with S-Box optimization,” In Advances in Cryptology - ASIACRYPT 2001, LNCS 2248, pp. 239-254, 2001.
26. P. Chodowiec and K. Gaj, “Very Compact FPGA Implementation of the AES Algorithm”, CHES 2003, LNCS 2779, pp.319-333, 2003

A Enlargements of differential power traces

Figure 9 shows the enlargement of each differential power trace in Fig.7 shown in Section 6.

B Experimental results of AES circuit

We implemented and evaluated the entire AES circuit using RSLUT. This AES circuit adopts the architecture described in Ref.[26]. S-boxes are the same as mentioned in Section 6. Figure 10 shows the DPA evaluation results. This is an example of differential power traces, the selection bits of which are the same as mentioned in the text. Maximum length code generated by shift-registers implemented in the FPGA is used for random signals. As shown in Fig.10, there is no DPA peak when looking at any bit. Thus, RSLUT works well as a countermeasure against DPA for the FPGA. The evaluation results of the AES circuit that uses Masked-AND instead of RSLUT are shown in Fig.11 for the purpose of comparison.

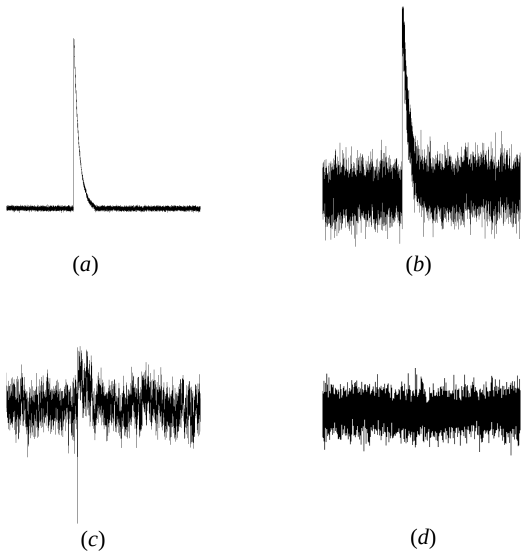


Fig. 9. Enlargements of traces in Fig.7

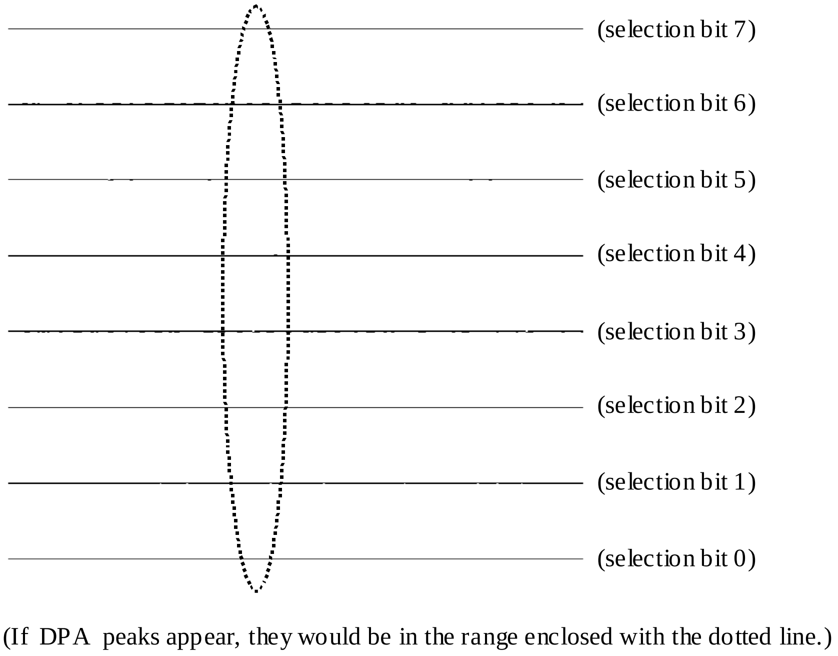


Fig. 10. Differential power traces of RSLUT (200000 samples)

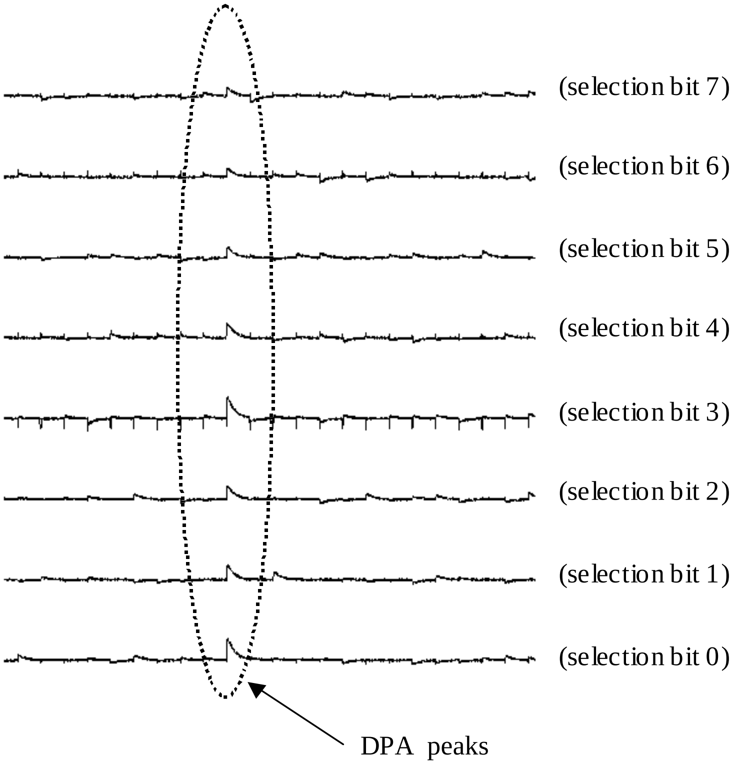


Fig. 11. Differential power traces of Masked-AND (200000 samples)