PAPER Special Section on Multiple-Valued Logic and VLSI Computing

Multiple-Valued Constant-Power Adder and Its Application to Cryptographic Processor*

Naofumi HOMMA[†a)], ***Member***, **Yuichi BABA**[†], **Atsushi MIYAMOTO**[†], ***Nonmembers***, ***and*** **Takafumi AOKI**[†], ***Member***

SUMMARY This paper proposes a constant-power adder based on multiple-valued logic and its application to cryptographic processors being resistant to side-channel attacks. The proposed adder is implemented in Multiple-Valued Current-Mode Logic (MV-CML). The important feature of MV-CML is that the power consumption can be constant regardless of input values, which makes it possible to prevent power-analysis attacks using dependencies between power consumption and intermediate values or operations of the executed cryptographic algorithms. In this paper, we focus on a multiple-valued Binary Carry-Save adder based on the Positive-Digit (PD) number system and its application to RSA processors. The power characteristic of the proposed design is evaluated with HSPICE simulation using 90 *nm* process technology. The result shows that the proposed design can achieve constant power consumption with lower performance overhead in comparison with the conventional binary design.

key words: *cryptographic processors, side-channel attacks, arithmetic circuits, multiple-valued logic, RSA cryptosystem*

1. Introduction

Cryptographic modules are now mounted on many embedded systems, such as smart-cards and mobile phones, and used to ensure the protection of privacy and confidential information in communications. The encryption/decryption process usually requires a large amount of arithmetic operations with very large operands. For example, public-key cryptosystems such as RSA perform modular exponentiation operations using more than 1,000-bit operands, which can be done by a set of modular multiplication operations. Therefore, the hardware implementation of cryptographic algorithm (i.e., cryptographic processor) is essential for achieving high-performance cryptographic modules in embedded systems with limited computational capability.

On the other hand, cryptanalysis based on side-channel information is a major concern for hardware designers. When a cryptographic module performs encryption or decryption, secret parameters correlated with the intermediate data being processed can be leaked as side-channel information via power dissipation, electromagnetic radiation, or during operation. In particular, power-analysis attacks using power dissipation from cryptographic modules have gained much attention due to their powerful properties. Two of the best known attacks are Simple Power Analysis (SPA) and Differential Power Analysis (DPA) proposed by Kocher [2], [3].

Typical countermeasure schemes against such power-analysis attacks are known as "masking" and "hiding." For both schemes, several techniques have been proposed in algorithm and circuit levels [4]. The concept of masking is to randomize intermediate data independently of secret keys, whereas that of hiding is to avoid or at least to reduce dependencies between power consumption and intermediate data or operation. The countermeasures in algorithm level [5]–[7] can be implemented easily, but can also be vulnerable to newly-developed algorithmic attacks [8], [9]. On the other hand, the countermeasures in circuit level can be general-purpose solutions at the expense of their difficulties in design.

Many circuit-level countermeasures have been proposed up until now [4], [10]–[12]. Random Switching Logic (RSL) [10] and Sense Amplifier Based Logic (SABL) [11] are well-known countermeasures for masking and hiding, respectively. RSL uses random data to mask the transition probabilities of inputs and outputs. SABL is a kind of differential logic to make the power constant with complementary operation scheme. Wave Dynamic Differential Logic (WDDL) [12] is an extended version of SABL and balances circuit activity successfully with complementary logic gates and precharge phase. Such conventional countermeasures, however, still require large overhead in power and delay as compared with straight-forward implementations without any countermeasure, and thus their applications are mainly limited to smaller ciphers, that is, symmetric key ciphers such as Data Encryption Standard (DES) and Advanced Encryption Standard (AES).

Addressing the above issue, we introduce Multiple-Valued Current-Mode Logic (MV-CML) as a key technology to design an efficient countermeasure in circuit level. MV-CML can perform linear summation operation by wiring point without any active devices. In addition, the power consumption of MV-CML can be independent of input values. Using such MV-CML circuitry, we design constant-power high-performance adder that can be a main component to build public-key cryptographic processors being resistant to power-analysis attacks. In this paper, we first propose a multiple-valued adder based on the binary Positive-Digit (PD) number [13] and its application to the

Manuscript received November 10, 2009.
Manuscript revised March 14, 2010.

[†]The authors are with the Department of Computer and Mathematical Sciences, Graduate School of Information Sciences, Tohoku University, Sendai-shi, 980–8579 Japan.

*A preliminary version of this paper was partially presented at ISMVL 2009 [1].

a) E-mail: homma@aoki.ecei.tohoku.ac.jp

DOI: 10.1587/transinf.E93.D.2117

Copyright © 2010 The Institute of Electronics, Information and Communication Engineers

data-path of RSA processors. We then evaluate the power characteristics of the propoased design by HSPICE simulation with 90 *nm* process technology in comparison with the conventional binary designs based on a standard CMOS cell library.

2. Binary Carry-Save Adder on Multiple-Valued Source-Coupled Logic

This section presents the design of Binary Carry-Save adder (BCSA) based on Multiple-Valued Source-Coupled Logic (MV-SCL) which is a variation of Multiple-Valued Current-Mode Logic (MV-CML). BCSA performs 2-input 1-output addition using the binary Positive-Digit (PD) number representation. First, we describe the binary PD number representation and the addition algorithm of two PD numbers. We then present the BCSA design based on MV-SCL.

2.1 Addition Algorithm for Binary PD Numbers

The binary PD number representation is a redundant representation with a positive digit set $\{0, 1, 2\}$ [14]. Assume that two integers X, and Y can be represented as follows:

$$X = [x_{n-1} \cdots x_i \cdots x_1 x_0]_{PD2},$$
$$Y = [y_{n-1} \cdots y_i \cdots y_1 y_0]_{PD2},$$

where $x_i, y_i \in \{0, 1, 2\}$ $(0 \le i \le n-1)$. The 1-digit 2-operand addition is represented by a counter tree diagram (CTD) [15] as shown in Fig. 1 (a). The inputs x_i, y_i and the output z_i have the digit set $[0:2] = \{0, 1, 2\}$, and the carries c_{i-1} and c_i have the digit set $[a:b] = \{a, a+1, \cdots, b\}$, where a and b are unknown integers satisfying $a < b$. These integers are determined by the addition algorithm. The outgoing carry c_i has the weight of 2 for the next digit.

The types of redundant addition algorithms are dependent on the characteristics (i.e., radix and redundancy) of the number representation. According to [15], there are two binary PD addition algorithms in CTD representation (Dual-carry algorithm and Conditional single-carry algorithm) since the radix is 2 and the redundancy is 1. Each algorithm has the two variants of 3-stage architectures: Type-I and Type-II. Thus, we have four types of two-operand binary PD addition algorithms in total.

Among them, we employ the dual-carry Type-I algorithm as shown in Fig. 1 (b). It is observed in [16] that this type achieved the highest performance for the redundant binary adder (RBA) in MV-CML as compared with other types. The main reason is that the Type-I algorithm can be implemented with a minimum set of comparators/current sources without any level shifter. The same implementation technique can also be available for BCSA with the same radix and redundancy as RBA. More precisely, the selected type can be implemented with only 10 current sources while other types require 13-19 current sources.

The addition algorithm in Fig. 1 (b) is performed by the following three steps in each digit:

$$\text{Step 1} : x_i + y_i = c_i^{(1)} + u_i,$$
$$\text{Step 2} : u_i + c_{i-1}^{(1)} = c_i^{(2)} + s_i,$$
$$\text{Step 3} : s_i + c_{i-1}^{(2)} = z_i,$$

where the left-hand side indicates input, the right-hand side indicates output, the carry c_i in Fig. 1 (a) is given by $c_i^{(1)}$ and $c_i^{(2)}$, and u_i and s_i are the intermediate sums. These variables take the following ranges:

$$x_i, y_i, z_i, u_i \in \{0, 1, 2\},$$
$$c_i^{(1)}, c_i^{(2)} \in \{0, 2\},$$
$$c_{i-1}^{(1)}, c_{i-1}^{(2)}, s_i \in \{0, 1\}.$$

Step 1 adds the two input digits x_i and y_i and decomposes the sum into the first intermediate sum u_i and the first carry $c_i^{(1)}$. The first carry $c_i^{(1)}$ is activated when the logical value of liner sum $(x_i + y_i)$ is equal to or higher than 3. Figure 2 depicts the values of s_i and $c_i^{(2)}$ depending on $c_{i-1}^{(1)}$ in Step 1. Step 2 then adds the intermediate sum u_i and the lower-digit carry $c_{i-1}^{(1)}$ and decomposes the sum into the second intermediate sum s_i and the second carry $c_i^{(2)}$. Finally, Step 3 creates the final sum z_i from the intermediate sum and the lower-digit carry $c_{i-1}^{(2)}$. As a result, the output range is the same as the input one. Since the final sum z_i is computed with the second lower-digit carry $c_{i-1}^{(2)}$, the carry propagation is limited to only one digit.

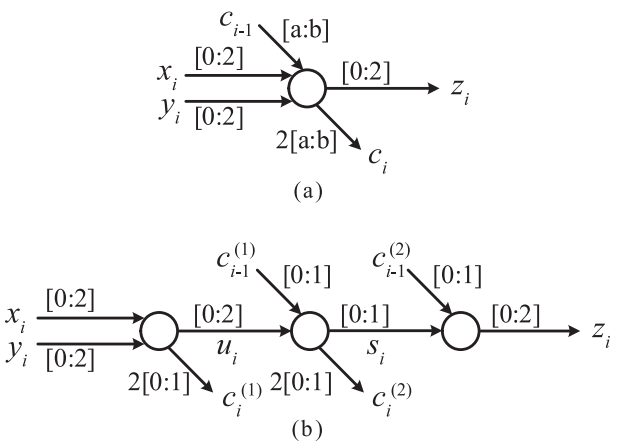

Fig. 1 Counter tree diagrams for binary PD addition algorithm: (a) 1-stage form and (b) 3-stage form (Dual-carry Type-I algorithm).

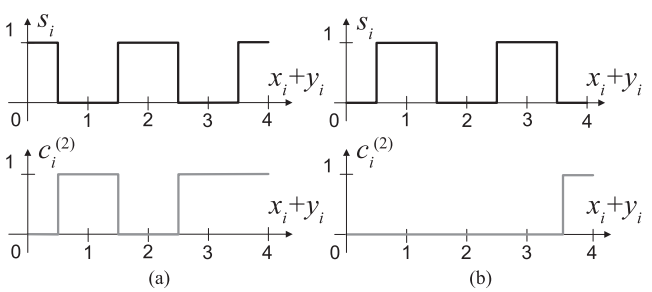

Fig. 2 Transfer characteristics for Step 2: (a) $c_{i-1}^{(1)} = 1$ and (b) $c_{i-1}^{(1)} = 0$.

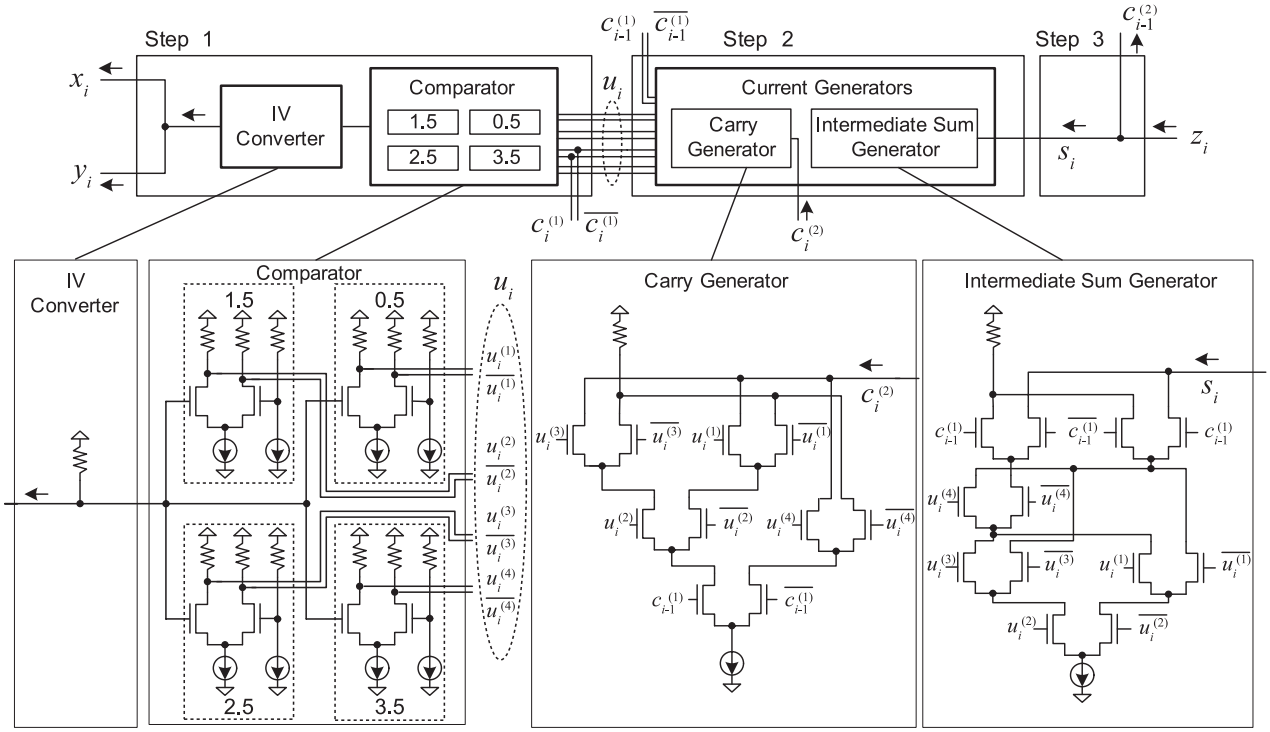

Fig. 3 Schematic of BCSA based on MV-SCL.

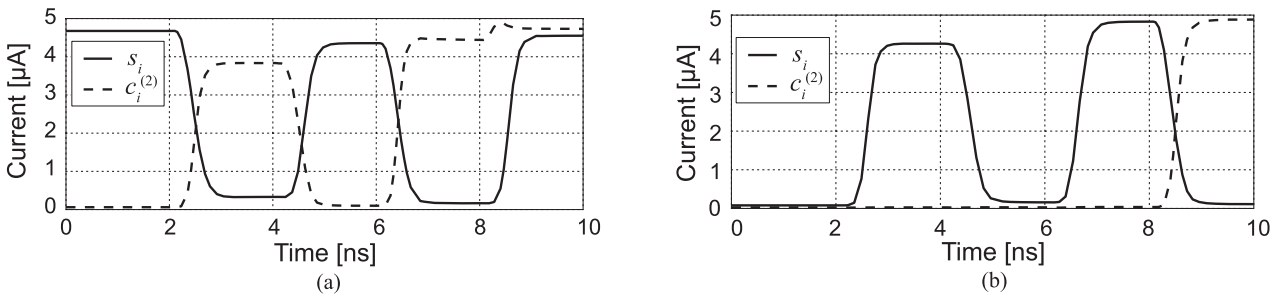

Fig. 4 HSPICE simulation for proposed BCSA: (a) $c_{i-1}^{(1)} = 1$ and (b) $c_{i-1}^{(1)} = 0$.

2.2 Binary Carry-Save Adder in Multiple-Valued Source-Coupled Logic

The proposed BCSA is designed by the combination of the above limited-carry-propagation algorithm and the MV-SCL circuit technology. The significant feature of MV-SCL considered here is that the power consumption can be constant regardless of the input values. In addition, the MV-SCL handles differential signals to perform faster operation as compared with conventional MV-CML circuits.

Figure 3 shows the schematic of our BCSA design. Step 1 consists of a wiring point, IV converter, and comparator. Multi-level current signals given by a wiring point are converted into voltage signals in the IV converter. The comparator then generates four pairs of differential signals from the voltage signals. The four signals from $u_i^{(1)}$ to $u_i^{(4)}$ show whether the voltage level of input is larger than 0.5, 1.5, 2.5, and 3.5 values, respectively. The carry signal $c_i^{(1)}$ which determines the transfer characteristics of the following intermediate sum and carry signals is equivalent to $u_i^{(3)}$. Each pair of signals is regarded as active when the value of positive signal (e.g., $u_i^{(1)}$) is higher than that of negative signal (e.g., $\overline{u_i}^{(1)}$). Note here that the comparator is symmetrically designed to balance the power consumption.

Step 2 then adds the intermediate sum u_i and the lower-digit carry $c_{i-1}^{(1)}$. The two current generators use the differential signals as gate inputs in source-coupled logic, and generate the current signals of carry $c_i^{(2)}$ and intermediate sum s_i, respectively. Each generator is designed so as to activate any one of the current paths independently of the input values, which allows us to observe constant power consumption.

Step 3 is implemented only by a wiring point, where the lower-digit carry $c_{i-1}^{(2)}$ and the intermediate sum s_i are added.

Figure 4 shows the transfer characteristics of the BCSA

obtained from an HSPICE simulation where the unit current is $5\,\mu A$, and the input signal $(x_i + y_i)$ is incremented from 0 to $20\,\mu A$ at every $2\,ns$. Figures 4 (a) and (b) indicate the conditions $c_{i-1}^{(1)} = 1$, and $c_{i-1}^{(1)} = 0$, respectively. The simulation is performed with ST Microelectronics $90\,nm$ process technology, where the supply voltage is 1.2 V, the resistors at comparators and IV converter/Curret generators are $20\,k\Omega$ and $80\,k\Omega$, respectively. We can observe here that the output signals s_i and $c_i^{(2)}$ are satisfied with the transfer characteristics shown in Fig. 2.

3. Applying BCSA to RSA Processor

This section presents the application of the proposed BCSA to RSA processor. The RSA cryptosystem is the de facto standard scheme in public-key cryptosystem, and is widely used for practical applications. On the other hand, a large amount of multiplication and squaring operations is required during the RSA operation, and the power waveform is mainly characterized by the arithmetic operations in the datapath. In the following, we describe the RSA algorithm based on Binary Carry-Save addition and its datapath architecture.

3.1 RSA Cryptosystem

RSA cryptosystem employs modular exponentiation [17] for encryption and decryption as follows:

$$C = P^E \bmod N, \tag{1}$$

$$P = C^D \bmod N, \tag{2}$$

where P is the plaintext, C is the ciphertext, E and N are the public-keys, and D is the secret-key. The key length k is at least 1,024 bits for security reasons.

The modular exponentiation is performed by repeating square and multiply operations according to the bit pattern of the keys (E and D). ALGORITHM 1 shows a left-to-right binary method which is commonly used for the modular exponentiations in smart-cards and embedded systems due to the low resource consumption. This algorithm scans the bits of the exponent from MSB to LSB, and always performs a squaring at Line 3 independently of the scanned bit value, but the multiply operation at Line 5 is executed only if the scanned bit is '1'. Therefore, the simple power analysis of RSA [2] is to observe differences between multiplication and squaring operations performed during modular exponentiation.

ALGORITHM 1
MODULAR EXPONENTIATION (MSB FIRST)

Input:	$X, N,$ $E = [e_{k-1} \ldots e_1 e_0]_2.$	
Output:	$Z = X^E \bmod N.$	
1 :	$Z := 1;$	
2 :	**for** $i = k - 1$ **downto** 0	
3 :	$Z := Z \times Z \bmod N;$	– squaring
4 :	**if** $(e_i = 1)$ **then**	
5 :	$Z := Z \times X \bmod N;$	– multiplication
6 :	**end if**	
7 :	**end for**	

One popular method to speed up the exponentiation is to use Montgomery's modular multiplication algorithm [18]. This algorithm replaces the modular division-by-N with a k-bit right shift operation since the division is one of the most time-consuming parts in modular multiplication. Given two large integers X and Y, the output of Montgomery multiplication algorithm is given as follows:

$$Z = XY2^{-k} \bmod N, \tag{3}$$

where $0 \le X, Y < N < 2^k$.

ALGORITHM 2 shows the original Montgomery multiplication algorithm. Eq. (3) can be calculated by one multiplication and a k-bit right shift operation if the lowest k bits of XY are equal to 0. For this purpose, a multiple of N is added to XY in this algorithm. The final result is not changed by the addition since Eq. (3) is in modulo N arithmetic. Also, the coefficient t is generated in advance using a pre-computed number W.

The very long word-length over 1,000 bits is sometimes unsuitable for the hardware implementation, and then the operands are divided into smaller m blocks ($k = r \cdot m$). The separation weight 2^r is called radix. In order to use r-bit operations, the k-bit operand X can be represented by r-bit words x_i $(0 \le i \le m - 1)$ as follows:

$$X = x_{m-1} \cdot 2^{r(m-1)} + \ldots + x_1 \cdot 2^r + x_0. \tag{4}$$

For simplicity, we use the following notation:

$$X = [x_{m-1} \ldots x_1 x_0]_{2^r}. \tag{5}$$

ALGORITHM 3 shows the radix-2 Montgomery multiplication algorithm where the multiply X is divided into m blocks and the multiplicand Y is given as k bits. As a result, ALGORITHM 3 is based on an r-bit three-term addition operation $z_j + tn_j + y_i x_j$.

3.2 Radix-2 RSA Processor Based on BCSA

The addition operation at Line 5 in ALGORITHM 3 is the most critical operation for circuit delay and area. In particular, the long carry-propagation chain of the addition causes longer delay for higher bits. In order to improve the circuit performance, we apply binary carry-save addition operation

ALGORITHM 2
MONTGOMERY MULTIPLICATION

Input:	$X, Y, N,$ $W = -N^{-1} \bmod 2^k.$	
Output:	$Z = XY2^{-k} \bmod N.$	
1 :	$t := XY \cdot W \bmod 2^k;$	
2 :	$Z := (XY + tN)/2^k;$	
3 :	**if** $(Z > N)$ **then** $Z := Z - N;$	

ALGORITHM 3
RADIX-2 MONTGOMERY MULTIPLICATION

Input:	$X = [x_{m-1} \dots x_1 x_0]_{2^r}$, $N = [n_{m-1} \dots n_1 n_0]_{2^r}$, $Y = [y_{k-1} \dots y_1 y_0]_2$.
Output:	$Z = XY2^{-k} \bmod N = [z_{m-1} \dots z_1 z_0]_{2^r}$.

```
1 : Z := 0;
2 : for i = 0 to k − 1
3 :     t := (z_0 + y_i x_0) mod 2;
4 :     for j = 0 to m − 1
5 :         z_j := (z_j + t n_j + y_i x_j)/2;
6 :     end for
7 : end for
```

ALGORITHM 4
RADIX-2 MONTGOMERY MULTIPLICATION BASED ON BCSA (*MontMult*)

Input:	$X = [x_{m-1} \dots x_1 x_0]_{2^r}$, $N = [n_{m-1} \dots n_1 n_0]_{2^r}$, $Y = [y_{k-1} \dots y_1 y_0]_2$.
Output:	$Z = XY2^{-k} \bmod N$.
Variable:	$W = [w_{m-1} \dots w_1 w_0]_{(PD2)^r}$, $V = [v_{m-1} \dots v_1 v_0]_{(PD2)^r}$.

```
1 : W = 0;
2 : for i = 0 to k − 1
3 :     t := (w_0 + y_i x_0) mod 2;
4 :     for j = 0 to m − 1
5 :         v_j = t n_j + y_i x_j;
6 :         w_j := (w_j + v_j)/2;
7 :     end for
8 : end for
9 : Z :=PD2toBinary(W);
```

with limited carry-propagation to the Montgomery multiplication algorithm.

ALGORITHM 4 shows the Montgomery multiplication algorithm where the addition operations of binary PD numbers are used at Line 6. Each addition outputs an intermediate sum and a carry with limited carry-propagation. No carry-propagation addition (CPA) is performed during the iteration cycles. The final output Z is obtained by CPA operations PD2toBinary only at the end of the algorithm (Line 9) which is equal to a conversion of binary PD numbers to binary numbers, but the adverse effect is limited.

ALGORITHM 5 shows the modular exponentiation algorithm (i.e., the left-to-right binary method) with the Montgomery multiplication (*MontMult*). Based on ALGORITHM 5, we designed the radix-2 RSA processor architecture with BCSA. Figure 5 shows a block diagram of the proposed RSA processor, which consists of five components: Datapath, Sequencer, Memories, Data Counter, and Key Shift. The exponent E is set into the k-bit shift register Key Shift. The k-bit data X and modulus N are divided into m blocks, and are sequentially stored into Memory. Datapath performs the above exponentiation operation by repeating multiplication and squaring operations according to the exponent bits. Each multiplication/squaring operation is done by a set of addition operations. The CPA is performed

ALGORITHM 5
MODULAR EXPONENTIATION WITH *MontMult*

Input:	X, N, $E = [e_{k-1}, \dots, e_1, e_0]_2$.
Output:	$Z = X^E \bmod N$.

```
1 : Y := XR mod N;
2 : Z := R mod N;
3 : for i = k − 1 downto 0
4 :     Z := MontMult (Z, Z, N);        – squaring
5 :     if (e_i = 1) then
6 :         Z := MontMult (Z, Y, N);    – multiplication
7 :     end if
8 : end for
9 : Z := MontMult (Z, 1, N);
```

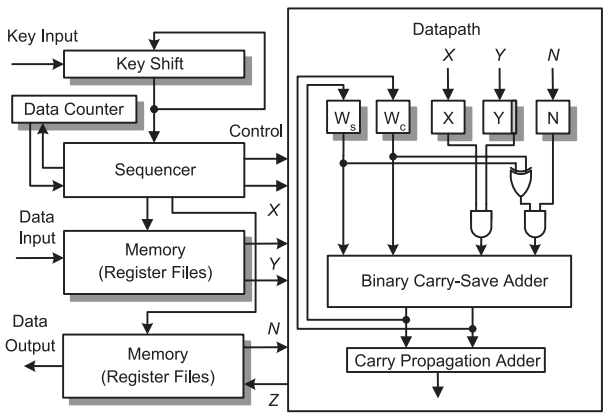

Fig. 5 RSA processor architecture.

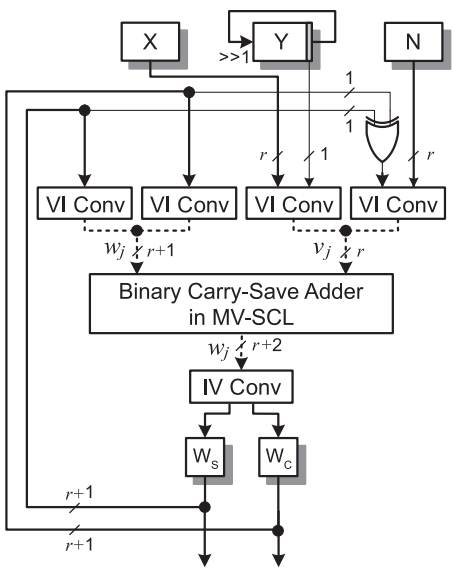

Fig. 6 Proposed datapath architecture.

at the end of iteration cycles for one multiplication/squaring. The iteration count and/or the data size are counted by Data Counter. The secret key bit is fed to the Sequencer bit by bit from the MSB side through the Key Shift.

Figure 6 shows the datapath architecture of the pro-

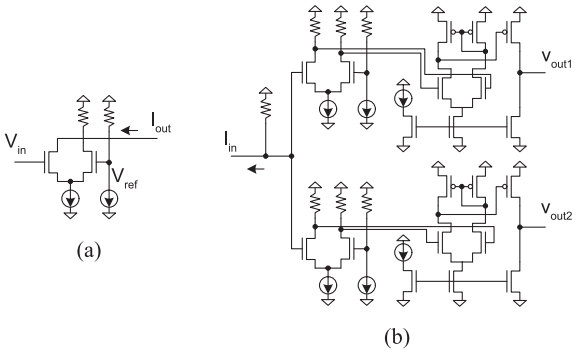

Fig. 7 Schematics of converters: (a) VI converter and (b) IV converter.

posed RSA processor, where the solid and dotted lines indicate the data flow of voltage and current signals, respectively. The squares VI Conv and IV Conv indicate VI and IV converters, respectively. Figure 7 shows the schematics of VI and IV converters. Besides, there are two variations of the VI converters with two inputs. The voltage signals for the intermediate sum and the carry are fed back into the BCSA in carry-save form.

4. Performance Evaluation

We designed a 32-bit datapath of the radix-2 RSA processor shown in Fig. 6, where the BCSAs and VI/IV converters are implemented in MV-SCL. We developed the MV-SCL cells using the ST Microelectronics 90 *nm* process technology. Figure 8 shows a chip layout of the designed datapath and the BCSA cell. For comparison, we also designed the corresponding binary-logic voltage-mode datapaths with/without a WDDL countermeasure [12]. The binary logic BCSA was implemented with a standard cell library using the same process technology. The unprotected and WDDL datapaths do not employ VI and IV converters since the intermediate sum and carry (in $\{0, 1, 2\}$) are given by binary vectors and are directly fed back to the BCSA. We obtained the physical layout data of the three data-paths from a place-and-route tool Astro and evaluated them by HSPICE simulations with BSIM4 hvt model, where the supply voltage is 1.2 *V*.

Table 1 compares the performance of the three processors, where Proposed indicates the proposed MV-SCL datapath, and Unprotected and WDDL indicate the binary CMOS datapaths without and with the WDDL countermeasure, respectively. The use of MV-SCL technology makes it possible to reduce the circuit delay (operating time) to 53% in comparison with the unprotected design, even though the circuit area and the power consumption (@312 MHz) are increased up to 1.64 and 2.18 times, respectively. The major reason is that the MV-CML design can perform liner summation operations (i.e., Steps 1 and 3) by wiring point without any active devices. Such result is consistent with the previous works [19]. The Power-Delay (PD) product of the proposed design is eventually 1.16 times larger than that of the unprotected design. However, the increase rate is much smaller than that of the WDDL countermeasure. The proposed design is 4.01 times faster in operation time, 3.23 times smaller in area, 1.78 times smaller in power, and 3.57 times smaller in PD product as compared with the WDDL design.

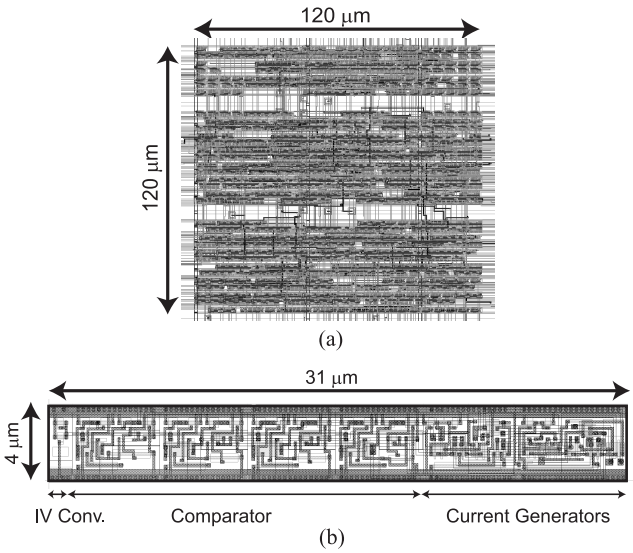

Fig. 8 Chip layout of datapath shown in Fig. 6: (a) overview and (b) BCSA cell.

Table 1 Performance comparison.

	Unprotected	WDDL	Proposed
Delay [*ns*]	3.11	3.31	1.65
MM Time* [*ms*]	50.95	108.46	27.03
Area [μm^2]	8736.9	46300.1	14332.7
Power [*mW*]	9.53	36.96	20.76
PD Product	29.64	122.34	34.25

∗ Computation time for one Montgomery multiplication

In order to evaluate the dependencies between power consumption and input values, we employed two specific inputs: random and one-hot-bit inputs. The use of random input allows us to observe the average power characteristics for the processors. On the other hand, the use of one-hot-bit input can provide the lowest power consumption for the binary logic processors since the gate switching activities become lower than any other inputs. This type of chosen-message technique is known as one of the most powerful techniques to enhance the effectiveness of power analysis attacks [20]. Applying such input to the left-to-right binary method (i.e., ALGORITHM 5), for example, we can reduce the power consumption only for the multiplication operation, and thus distinguish the difference between multiplication and squaring operations easily. In this experiment, we employ $Y = 80000000_{(16)}$ for the one-hot-bit input.

Figures 9-11 show the power characteristics of the three datapaths obtained by HSPICE simulations. Figures 9 and 10 are those of the binary logic processors without and with a WDDL countermeasure, respectively. Figure 11 is that of the proposed MV-SCL processor. Each power waveform contains eighteen binary carry-save addition operations. The result in Fig. 9 clearly shows that the unprotected

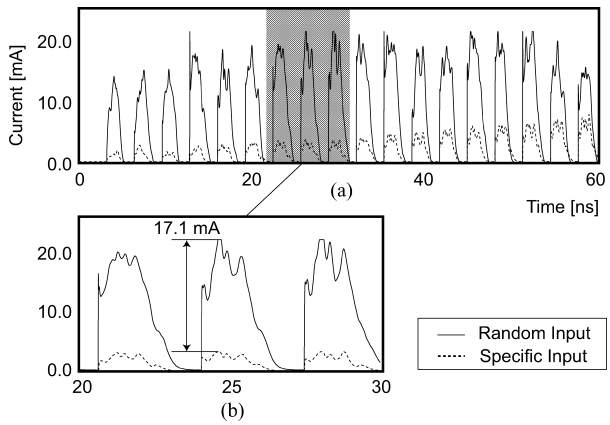

Fig. 9 Power waveforms of unprotected design: (a) overview and (b) magnified view.

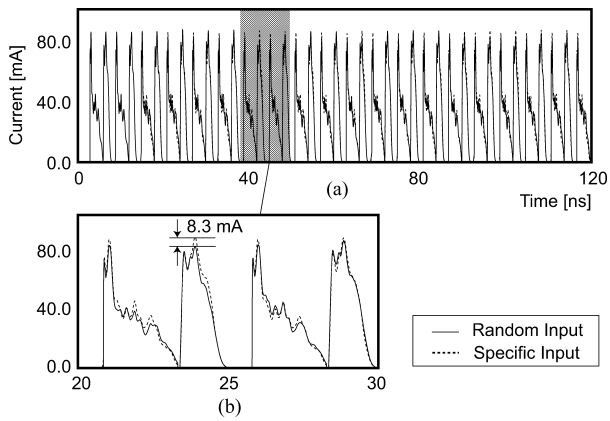

Fig. 10 Power waveforms of WDDL design: (a) overview and (b) magnified view.

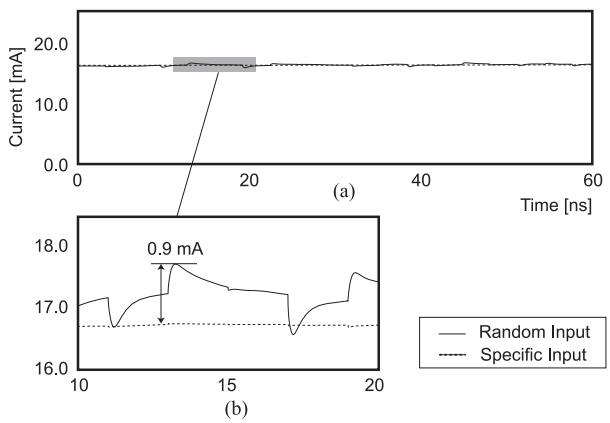

Fig. 11 Power waveforms of proposed design: (a) overview and (b) magnified view.

datapath has a large difference in power consumption between the random input and the specific input. On the other hand, the results in Figs. 10 and 11 show that protected datapaths achieve constant power consumption independently of the input values. We can observe here that the maximum differences in current at the same timing for Unprotected and Proposed are 17.1 and 0.9 mA, respectively.

Table 2 Statistic comparison of power waveforms.

	Input	Average	SD*
Unprotected	Random	18.83	4.42
	Specific	5.74	2.47
WDDL	Random	83.31	13.09
	Specific	85.73	13.38
Proposed	Random	17.65	0.24
	Specific	16.71	0.03

*SD: Standard Deviation

More precisely, Table 2 summarizes the statistic comparison of power waveforms for the three processors, where Random and Specific indicate the random input and the specific input, respectively. The statistic data indicate that the two waveforms by Proposed are statistically indistinguishable during each operation with 18 cycles. Assuming that attackers would not estimate any one of the 1,024 bits for the secret key, we apply the t-test function in the significant level p of 0.05 percent [4]. The number of points n to distinguish two sets X_1 and X_2 is given as

$$n = \frac{s_1^2 + s_2^2}{(\overline{x_1} - \overline{x_2})^2} t^2, \tag{6}$$

where $\overline{x}$ indicates the average value of X, s indicates the standard deviation, and t indicates the t-value. When $p = 0.0005$, $t = 2.447$. As a result, the n values of Unprotected, WDDL, and Proposed in Table 2 are calculated, respectively. In this sense, both WDDL and Proposed are sufficient to prevent power analysis on the proposed architecture. Thus, the MV-SCL circuits can be an effective countermeasure against power-analysis attacks.

5. Conclusions

This paper presented the design of a constant-power adder based on MV-SCL and its application to RSA processor. We evaluated the power characteristics with HSPICE simulations and showed that the proposed adder has constant power consumption independently of the input value, which is comparable to the binary design with a conventional WDDL countermeasure. In addition, we confirmed that the efficiency (i.e., power-delay product) of the proposed datapath was almost equal to the unprotected binary datapath and was about 4-times better than the WDDL design. Note here that the power consumption can be further reduced by an advanced MV-SCL technology such as Dynamic Source-Coupled Logic [21]. Thus, the MV-SCL circuit technology has a definite possibility of achieving high-performance and tamper-resistant cryptographic processors.

On the other hand, the simulation in this paper did not include the power transitions of registers surrounding the datapaths. The power differences caused by the registers would also be replaced with tamper-resistant resistors implemented in masking or hiding logic style. The total evaluation of power consumption in the datapath is being left for the future study. In addition, the statistical data in Table 2 showed that the proposed design still has a difference

in power consumption between random and specific inputs in comparison with WDDL design. The further equalization would be another important work in the future.

References

[1] Y. Baba, A. Miyamoto, N. Homma, and T. Aoki, "Multiple-valued constant-power adder for cryptographic processor," Proc. 39th IEEE Int. Symp. Multiple-Valued Logic, pp.239–244, May 2009.
[2] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," CRYPTO 1999, Lecture Notes in Computer Science, vol.1666, pp.388–397, Aug. 1999.
[3] P. Kocher, R. Lee, G. McGraw, and A. Raghunathan, "Security as a new dimension in embedded system design," Proc. 41st annual conference on Design automation, pp.753–760, ACM Press, June 2004.
[4] S. Mangard, E. Oswald, and T. Popp, Power Analysis Attacks: Revealing the Secrets of Smart Cards, Springer-Verlag, 2007.
[5] J.S. Coron, "Resistance against differential power analysis for elliptic curve cryptosystems," CHES 1999, Lecture Notes in Computer Science, vol.1717, pp.192–302, Aug. 1999.
[6] M. Joye and S.M. Yen, "The montgomery powering ladder," CHES 2002, Lecture Notes in Computer Science, vol.2523, pp.291–302, Aug. 2002.
[7] M. Joye, "Highly regular right-to-left algorithms for scalar multiplication," CHES 2007, Lecture Notes in Computer Science, vol.4727, pp.135–147, Sept. 2007.
[8] A.P. Fouque and F. Valette, "The doubling attack -why upwards is better than downwards," CHES 2003, Lecture Notes in Computer Science, vol.2779, pp.269–280, Sept. 2003.
[9] N. Homma, A. Miyamoto, T. Aoki, A. Satoh, and A. Shamir, "Collision-based power analysis of modular exponentiation using chosen-message pairs," CHES 2008, Lecture Notes in Computer Science, vol.5154, pp.15–29, Aug. 2008.
[10] D. Suzuki, M. Saeki, and T. Ichikawa, "Random switching logic: A new countermeasure against DPA and second-order DPA at the logic level," IEICE Trans. Fundamentals, vol.E90-A, no.1, pp.160–168, Jan. 2007.
[11] K. Tiri, M. Akmal, and I. Verbauwhede, "A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards," Proc. 28th European Solid-State Circuits Conference, pp.403–406, Sept. 2002.
[12] K. Tiri, D. Hwang, A. Hodjat, B.C. Lai, S. Yang, P. Schaumont, and I. Verbauwhede, "Prototype IC with WDDL and differential routing — DPA resistance assessment," Proc. Cryptographic Hardware and Embedded Systems 2005, pp.354–365, May 2005.
[13] S. Kawahito, M. Ishida, T. Nakamura, M. Kameyama, and T. Higuchi, "High-speed area-efficient multiplier design using multiple-valued current-mode circuits," IEEE Trans. Comput., vol.43, no.1, pp.34–42, Jan. 1994.
[14] N. Takagi, "Multiple-valued-digit number representations in arithmetic circuit algorithm," Proc. 32nd International Symposium on Multiple-Valued Logic, pp.224–235, May 2002.
[15] N. Homma, T. Aoki, and T. Higuchi, "Systematic interpretation of redundant arithmetic adders in binary and multiple-valued logic," IEICE Trans. Electron., vol.E89-C, no.11, pp.1645–1654, Nov. 2006.
[16] N. Homma, K. Degawa, T. Aoki, and T. Higuchi, "Algorithm-level optimization of multiple-valued arithmetic circuits using counter tree diagrams," Proc. 37th IEEE Int. Symp. Multiple-Valued Logic, no.31, pp.1–8, May 2007.
[17] J.A. Menezes, C.P. Oorschot, and A.S. Vanstone, Handbook of Applied Cryptography, CRC Press, 1997.
[18] P.L. Montgomery, "Modular multiplication without trial division," Mathematics of Computation, vol.44, no.170, pp.519–521, April 1985.
[19] T. Hanyu and M. Kameyama, "A 200 MHz pipelined multiplier using 1.5 V-supply multiple-valued MOS current-mode circuits with dual-rail source-coupled logic," IEEE J. Solid-State Circuits, vol.30, no.11, pp.1239–1245, Feb. 1995.
[20] A. Miyamoto, N. Homma, T. Aoki, and A. Satoh, "Chosen-message SPA attacks against FPGA-based RSA hardware implementation," Proc. 2008 International Conference on Field Programmable Logic and Applications, pp.35–40, Sept. 2008.
[21] A. Mochizuki, T. Hanyu, and M. Kameyama, "Design of a low-power multiple-valued integrated circuit based on dynamic source-coupled logic," Proc. 35th International Symposium on Multiple-Valued Logic, pp.481–517, May 2005.

Naofumi Homma received the B.E. degree in information engineering, and the M.S. and Ph.D. degrees in information sciences from Tohoku University, Sendai, Japan, in 1997, 1999 and 2001, respectively. He is currently an Associate Professor of the Graduate School of Information Sciences at Tohoku University. For 1999-2001, he was a Research Fellow of the Japan Society for the Promotion of Science. For 2002-2006, he also joined the Japan Science and Technology Agency (JST) as a researcher for the PRESTO project. He is a member of Cryptographic Implementation Committee in CRYPTREC (Cryptography Research and Evaluation Committees). His research interests include computer arithmetic, EDA methodology, high-performance/secure VLSI computing, and cryptographic hardware. Dr. Homma received the IP Award at the 2005 LSI IP Design Award, and the Best Papar Award at the Workshop on Synthesis And System Integration of Mixed Information Technologies in 2007.

Yuichi Baba received the B.E. degree in information engineering from Tohoku University, Sendai, Japan, in 2008. He is currently working toward the M.S. degree at Tohoku University. His research interest includes secure VLSI computing and embedded systems.

Atsushi Miyamoto received the B.E. degree in information engineering and the M.S. degree in information sciences from Tohoku University, Sendai, Japan, in 2005 and 2007, respectively. He is currently working towards the Ph.D. degree at Tohoku University. From 2009, he is also a Research Fellow of the Japan Society for the Promotion of Science. His research interests include cryptographic hardware, computer arithmetic and algorithms for high-performance VLSI computing.

Takafumi Aoki received the B.E., M.E., and D.E. degrees in electronic engineering from Tohoku University, Sendai, Japan, in 1988, 1990, and 1992, respectively. He is currently a Professor of the Graduate School of Information Sciences at Tohoku University. For 1997–1999, he also joined the PRESTO project, Japan Science and Technology Corporation (JST). His research interests include theoretical aspects of computation, digital signal processing, computer vision, image processing, biometric authentication, and security issues in computer systems. Dr. Aoki received the Outstanding Paper Award at the 1990, 2000, 2001 and 2006 IEEE International Symposiums on Multiple-Valued Logic, the Outstanding Transactions Paper Award from the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan in 1989 and 1997, the IEE Ambrose Fleming Premium Award in 1994, the IEICE Inose Award in 1997, the IEE Mountbatten Premium Award in 1999, the Best Paper Award at the 1999 IEEE International Symposium on Intelligent Signal Processing and Communication Systems, the IP Award at the 7th LSI IP Design Award in 2005, and the Best Paper Award at the 14th Workshop on Synthesis And System Integration of Mixed Information technologies in 2007.