Reconfigurable Architecture of a High-Speed RSA Encryption Processor with Built-in Table for Residue Calculation of Redundant Binary Numbers

Nobuhiro Tomabechi
Hachinohe Institute of Technology, Japan
E-mail: tomabech@hi-tech.ac.jp

1. Introduction

With the recent progress in communication through computer networks, realization of high-speed encryption/decryption processors to ensure security of communication is expected. Numerous studies on high-speed RSA encryption processors have been reported [1]-[6], however a much higher speed is desired.

In the preceding study [7], the authors presented a high-speed RSA encryption processor based on redundant binary number arithmetic and table-look-up. Since both redundant binary number arithmetic and table-look-up are basically parallel operations, a high-speed RSA encryption processor can be realized. However, the encryption rate of the presented processor is not much higher than that of conventional processors when the plaintext data is continuously input.

The purpose of this study is to improve the encryption rate of the processor for continuous input data without decreasing the encryption rate for discontinuous input data. In this paper, we newly present a reconfigurable architecture in which the structure of the processor is changeable depending on the form of input data.

At first, the structure suitable for continuous input data is presented in which pipelining is effectively employed. Secondly, the structure suitable for discontinuous input data is shown in which the binary tree structure is employed in order to utilize the parallelism of redundant binary arithmetic and table-look-up. Finally, the reconfigurable structure is shown. By the introduction of the presented structure, a high-speed encryption for both continuous and discontinuous input can be realized.

It is demonstrated that, (1) the encryption rate for continuous input data of the proposed processor is approximately 3/2 times that of conventional processors, (2) the encryption rate for discontinuous input data is 33 times that of conventional processors.

2. High-Speed RSA Encryption Algorithm

Let us express the encryption key as (e, n), the key length as N ($N \geq 1024$), the plaintext as M, and the ciphertext as C. In the RSA encryption, C is given by $C=M^e \bmod n$.

The redundant binary number is denoted by RBN and is expressed as x^*, here.

When e is expressed as $e=e_{N-1}2^{N-1}+e_{N-2}2^{N-2}+\ldots+e_0$, C is calculated as follow;

[Algorithm 1] RSA encryption algorithm

(Step 1) $y^*_1=1$

(Step 2) Following steps are repeated while i=1,2,…,N.

(Step 2-1) $y^*_2=y^*_1 \times y^*_1$

(Step 2-2) $y^*_1=y^*_2 \bmod n$

(Step 2-3) If $e_{N-i}=0$ then return to Step 2-1,
otherwise Step 2-4 and Step 2-5 are performed.

(Step 2-4) $y^*_2=y^*_1 \times M$

(Step 2-5) $y^*_1=y^*_2 \bmod n$

(Step 3) $C=y^*_1$

[Design of the table for residue calculation]

1.The table is composed of H RAMs, which are expressed as RAM_i (i=0, 1, 2,..., H-1).

2. The address of the RAMs is expressed in the RBN form. The address of RAM_i (i=0, 1, 2,..., H-1) is L bits, where L=N/H.
3. If an address of RAM_i, (i=0, 1, 2,..., H-1) is expressed as $A(y^*_{L-1}, y^*_{L-2},..., y^*_0)$, then T_i below is stored in this address.

$T_i=(y^*_{L-1}2^{N+iL+L-1}+y^*_{L-2}2^{N+iL+L-2}+,...,+y^*_0 2^{N+iL})$ mod n

[Algorithm 2] Residue calculation algorithm

The input data is expressed as $x^*=(x^*_{2N-1}, x^*_{2N-2}, .., x^*_0)$, then x^* mod n is calculated as follows;

(Step 1) The following steps are repeated while j=1,2,3,...,R.

(Step 2) For RAM_i (i=0, 1, 2,..., H-1), $A(x^*_{N+iL+L-1}, x^*_{N+iL+L-2}, .., x^*_{N+iL})$ is given as the address data, and the output T_i is obtained.

(Step 3) $y^*=\sum_{i=0}^{C(j)-1} T_i +(x^*_{N-1}, x^*_{N-2},..., x^*_0)$

C(1)=H, $C(j)=\log_2(C(j-1)+1)/L$, (j=2,3,4...,R)

3. Structure for Continuous Input Data

Algorithm 1 repeats multiplication and residue calculation 2N times. The multiplier and the residue calculation circuit will be designed to perform pipelined operation. However, pipelined operation will not be applied to operations repeating 2N times.

3.1 Multiplier Circuit

A set of a 1-bit×N-bit RBN multiplier and a 2N-bit RBN adder with a 2N-bit latch will be referred as a unit cell for the multiplier. N unit cells are sequentially connected in the pipelined form.

3.2 Residue Calculation Circuit

1. We will take L =2 for RAMs in the residue table.
2. A set of a 2-bit RBN address RAM with N-bit output and a 2N-bit RBN adder with a 2N-bit latch will be referred as a unit cell for the residue calculation circuit.
3. (N/2+Q) unit cells are sequentially connected in the pipelined form, where Q denotes the sum of C(j) (j=2, 3, … , R).

Fig. 1 shows the circuit diagram of the structure for continuous input data, and Fig. 2 shows the time chart.

4. Structure for Discontinuous Input Data

4.1 Multiplier Circuit

1. N×N 1-bit RBN multipliers are arrayed.
2. N partial products are summed up using a binary tree structure of (N-1) 2N-bit RBN adders.
3. The output of the binary tree structure is held by a 2N-bit latch.

4.2 Residue Calculation Circuit

1. We will take L =2 for RAMs in the residue table.
2. The overall residue calculation circuit is composed of R partial residue calculation circuits corresponding to R repeating cycles.
3. In each partial residue calculation circuit, the outputs of RAMs in the table are summed up using a binary tree structure of 2N-bit RBN adders.
4. The unit clock interval is determined equal to the operation time of the 2nd residue calculation circuit which is composed of 6 RBN adders.

Fig. 3 and Fig. 4 show the circuit diagram and the time chart of the structure for discontinuous input data, respectively.

5. Reconfigurable Architecture

Let us refer the structure for continuous input data as Structure 1, and the structure for discontinuous input data as Structure 2.

A reconfigurable structure, that is, a changeable structure between Structure 1 and Structure 2 depending on the form of input data will be presented here, which is designed as follows;

1. (N+N/2+Q) unit cells defined for Structure1 are arrayed.
2. In the continuous mode, interconnection lines are connected so as to realize Structure 1.
3. In the discontinuous mode, interconnection lines are changed so as to realize Structure 2.

Fig. 5 shows the reconfigurable structure for a unit cell of the multiplier circuit.

By this structure, a high-speed encryption can be realized for both continuous and discontinuous input.

6. Encryption Rate

6.1 Encryption Rate of the Continuous Structure

Let us define the continuous encryption rate, V_c as the data size possible to be processed per a second (bit/s) when the plaintext data is continuously input. Let us express the delay time of a single gate as t_g, the gate delay of a 1-bit RBN adder as t_{A0}, the gate delay of a latch as t_{F0} and the clock interval of the pipelined operation as t_0. Then, we have $V_c=2N/(4N^2t_0)=1/(2Nt_0)$, $t_0=t_{A0}+t_{F0}$.

From Ref. [7], we have $t_{A0}=6t_g$ and $t_{F0}=2t_g$. So, $t_0 \cong 8t_g$ and $V_c \cong 1/(16t_g)$. Thus, we can state as follows;

[Result 1] The encryption rate for continuous input data of the proposed processor is approximately $1/(16t_g)$ (bits/s), where t_g denotes the delay time of a single gate.

Assuming t_g=0.1ns as a typical value, we have V_c=0.62 (Gbits/s).

6.2 Encryption Rate of the Discontinuous Structure

Let us define the discontinuous encryption rate, V_d as the data size possible to be processed per a second (bit/s) when the plaintext data is only N bits.

Let us express the line delay of an interconnection line with the length of a 1-bit RBN adder as t_{L0}, the gate delay of a 1-bit RBN multiplier as t_{M0}, the read out time of a RAM as t_{R0}, the operation time of the multiplier circuit as T_M, the operation time of the i-th residue calculation circuit as T_{Ri}, the unit clock interval as t_0 and the overall encryption time of Structure 2 for N-bit data as T_P.

Assuming $t_{L0} \cong t_{A0}/100$, we obtain $T_M \cong 124t_g$, $T_{R1} \cong 112t_g$, $t_0=T_{R2} \cong 26t_g$.

We will take $T_M=5t_0$, $T_{R1}=5t_0$, $T_P=2N(T_M+T_{R1}+(R-1)t_0)=2N(5+5+4)t_0=28Nt_0$.

Then, $V_d=N/T_P=N/(28Nt_0)=1/(728t_g)$. Thus, we can state the following result.

[Result 2] The encryption rate for discontinuous input data of the proposed processor is approximately $1/(728t_g)$ (bits/s), where t_g denotes the delay time of a single gate.

Assuming t_g=0.1ns as a typical value, we have V_d=13 (Mbits/s).

6.3 Comparison with the Previous Studies

Numerous studies have been reported on the implementation of RSA encryption processors. As a typical processor, we will consider that presented in Ref. [3], since the number of gates through the critical path determining the operation time seems to be minimized in the basic sense.

The continuous encryption rate V_3 of the processor in Ref. [3] can be estimated as $V_3 \cong 1/(24t_g)$. Then, $V_3/V_c=24t_g/(16t_g)=3/2$. Thus, we can state the following result.

[Result 3] The encryption rate for continuous input data of the proposed processor is approximately 3/2 times that of conventional processors.

The gate delay through the critical path in the processor T_4 and the encryption rate for discontinuous input V_4 are estimated as $T_4 \cong 24N^2t_g$ and $V_4 \cong 1/(24Nt_g)$. Then, we have $V_d/V_4 \cong 24\times1024/728=33$.

So, we can state the following result.

[Result 4] The encryption rate for discontinuous input data of the proposed processor is approximately 33 times that of conventional processors.

7. Conclusions

In this paper, a high-speed RSA encryption processor based on RBN arithmetic, table-look-up and reconfigurable architecture has been presented. The following results are obtained; (1) The encryption rate for continuous input of the proposed processor is approximately 3/2 times that of conventional processors. (2) The encryption rate for discontinuous input is approximately 33 times that of conventional processors..

References

[1] Brickell E. F. “A survey of hardware implementations of RSA,” Advances in Cryptology-CRYPTO’89, Springer-Verlag, pp.368-370, 1990.

[2] Kameyama M., Wei S., Higuchi T. "Design of a RSA encryption processor based on signed-digit multivalued arithmetic circuits, "Systems and Computers Japan, Vol.21, pp.21-31, 1990.

[3] Eldridge S. E., Walter C. D. “Hardware implementation of Montgomery’s modular multiplication algorithm,” IEEE Trans. on Computers, Vol.42, No.6, pp.693-699, June, 1993.

[4] Chen P. S., Hwang S. A., Wu C. W., "A systolic RSA public key cryptosystem," Proc. ISCAS, Vol. 4, pp.408-411, 1996.

[5] Ishii S., Oyama K., Yamanaka K. "High-speed public key encryption processor," IEICE Japan Trans. (D-I), Vol. J80-D-I, No.8, pp.725-735, Aug., 1997.

[6] Yang C. C., Chang T. S., Jen C. W., "A new RSA cryptosystem hardware design based on Montgomery's algorithm," IEEE Trans. Circuits and Systems, Vol.45, No. 7, pp.908-913, July 1998.

[7] Tomabechi N., Ito T. "Design of a high-speed RSA encryption processor based on the residue table for redundant binary numbers," Systems and Computers in Japan, Vol.33, No.5, pp. 1-10, May 2002.

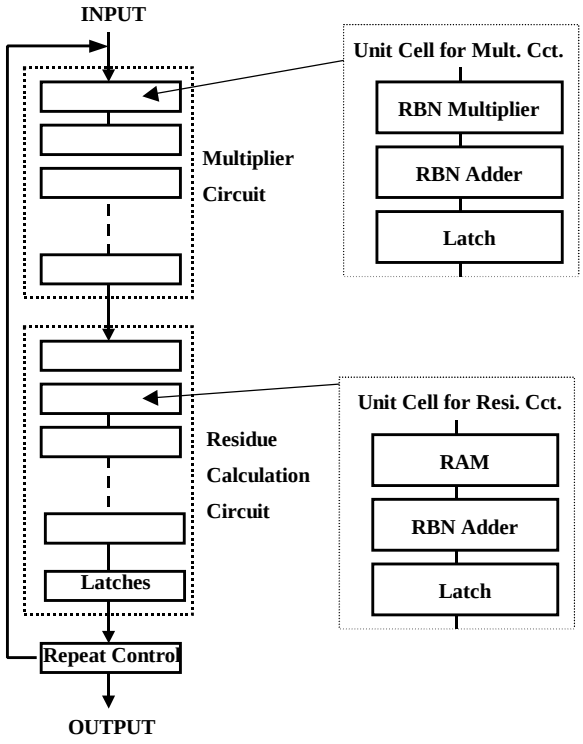

Fig.1 Circuit diagram of the structure for continuous input data

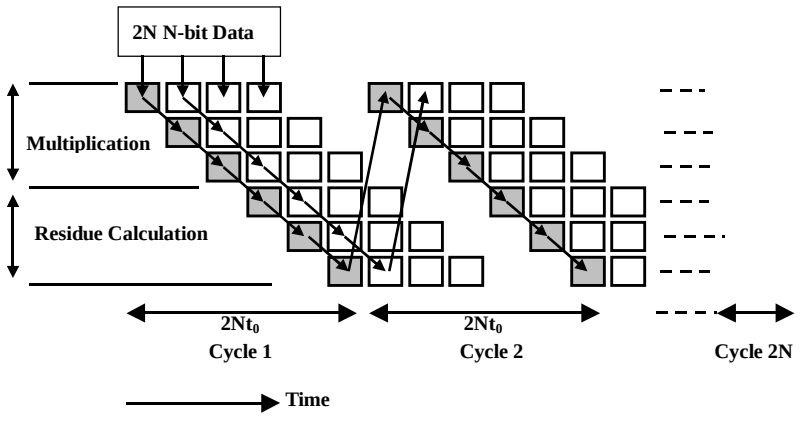

Fig.2 Time chart of the structure for continuous input data

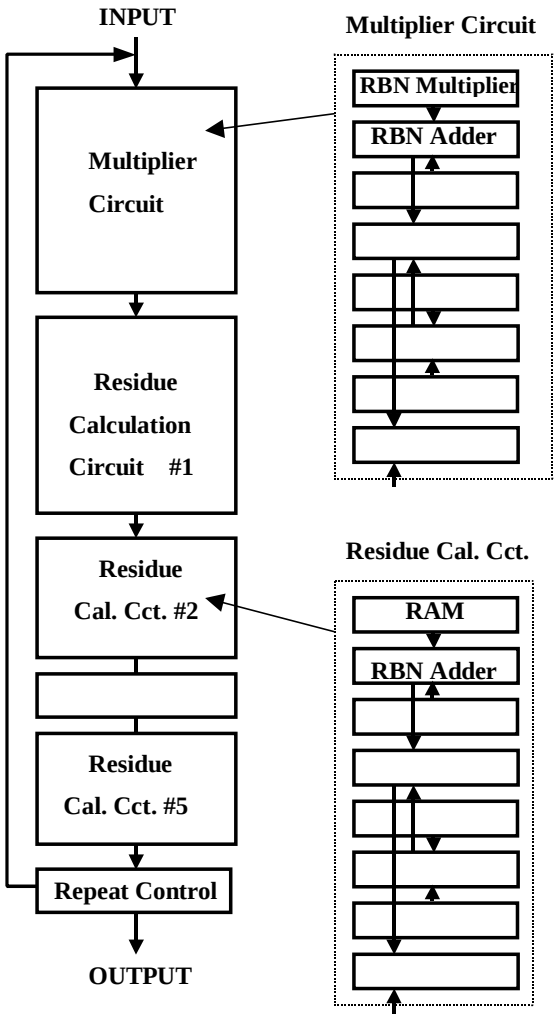

Fig.3 Circuit diagram of the structure for discontinuous input data

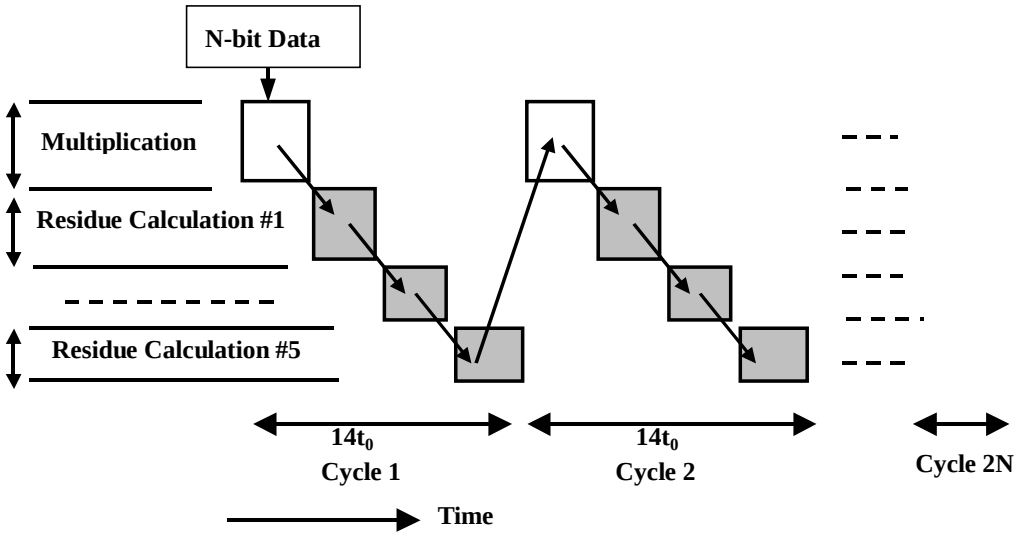

Fig.4 Time chart of the structure for discontinuous input data

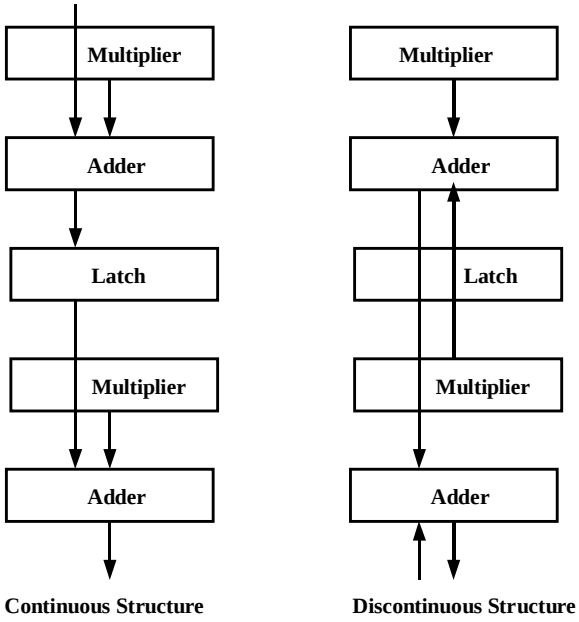

Fig. 5 Reconfigurable Structure
(a unit cell of the multiplier circuit)