A 6.3nJ/op Low Energy 160-bit Modulo-Multiplier for Elliptic Curve Cryptography Processor

Hyejung Kim, Yongsang Kim, and Hoi-Jun Yoo
Department of EECS, Korea Advanced Institute of Science and Technology (KAIST)
373-1, Guseong-dong, Yuseong-gu
Daejeon, 305-701, Republic of Korea
E-mail : seeseah@eeinfo.kaist.ac.kr

***Abstract*— A low energy modulo-multiplier is proposed for elliptic curve cryptography (ECC) processor, especially for authentication in mobile device or key encryption in embedded health-care system. The multiplier uses only two 40-bit multipliers to execute 160-bit operation based on the Montgomery modulo-multiplication algorithm. Partial products of multiplication are accumulated with shift registers to get final 160-bit MSB of output value. One modulo-multiplication is executed with 20 clock cycles at 40MHz operating frequency. It consumes 6.3nJ for each modulo-multiplication at 1V supply voltage. It is implemented by using 0.18-µm CMOS process and has 0.7mm x 1.0mm area.**

I. INTRODUCTION

Recently, security, such as user authentication and key encryption, is steadily becoming more important issue in data communication. The related research is getting popular for mobile device or private data transmission in embedded health-care system. Elliptic Curve Cryptography (ECC) is attractive for the mobile applications because of its small key size and high security level, as shown in Table I [1]. Although it requires more computational power, overall system power consumption is reduced. That is because total power consumption mainly depends on communication power rather than computational power [2] and ECC reduces the bit length for transmitting and receiving.

To date, a large number of ECC hardware have been implemented [3–6]. Xu proposed a fast ECC processor with hardware accelerator on ARM7 platforms [3]. Orlando proposed a scalable architecture in terms of area and speed specially suited for memory-rich hardware platforms such as field programmable gate arrays (FPGA) [4]. Satoh implemented an ECC processor able to operate in both GF(p) and GF(2^m) [5]. And McIvor proposed a new unified modular inversion algorithm with a full-word multiplier that offers much fewer clock cycles [6]. However, these architectures consume relatively high power to compute modular multiplication which is a key operation for cryptography. Meanwhile, mobile device or embedded health-care system requires energy efficient architecture due to their limitation of battery power.

TABLE I. COMPARISON OF KEY SIZE [1]

Bits of Security	Symmetric key algorithm	DSA	RSA	ECC
80	2TDEA	L=1024 N=160	k=1024	f=160
112	3TDEA	L=2048 N=224	k=2048	f=224
128	AES-128	L=3072 N=256	k=3072	f=256
192	AES-192	L=7680 N=384	k=7680	f=384
256	AES-256	L=15360 N=512	k=15360	f=512

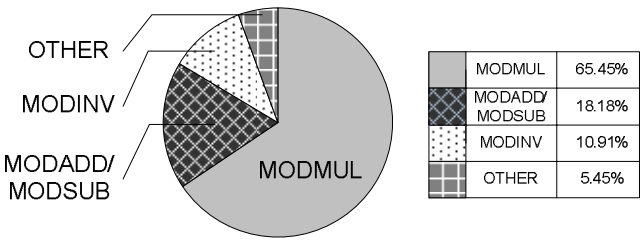

Figure 1. Percentage of processing time of ECC pipeline

This paper proposes a low energy modulo-multiplier which is implemented based on the Montgomery modulo-multiplication algorithm. In ECC encryption and decryption operation, modulo-multiplication (MODMUL) is most frequently used and has high complexity as shown in Fig. 1. Therefore, reduction of energy consumption of MODMUL operation is the way to reduce total energy consumption of overall system dramatically. To achieve low energy ECC operation, the proposed modulo-multiplier uses two 40-bit length multipliers which make energy optimum and computes 160-bit x 160-bit partial modulo-multiplication with small-bit registers. The details of the proposed work will be explained in the following sections.

978-1-4244-1684-4/08/$25.00 ©2008 IEEE

II. MATHMATICAL BACKGROUND

The Elliptic Curve Cryptographic Processor (ECCP) requires scalar point multiplication for arbitrary elliptic curves [7–8]. For fields GF(p), the curves are defined by

$$E: y^2 = x^3 + ax + b, \text{ where } 4a^3 + 27b^2 \neq 0 \pmod{p}. \quad (1)$$

To implement ECCP, it is strongly recommended to select parameters from among the example parameters listed in [9]. For example, we choose 160-bit prime number, p in (2), as elliptic curve domain parameters over GF(p).

$$p = 0x\_ffffffff\_ffffffff\_ffffffff\_ffffffff\_7fffffff = 2^{160} - 2^{31} - 1. \quad (2)$$

Modulo operation is complex because it involves division operation. Montgomery proposed that division operation can be replaced by simple shift operation which is comparatively easy to perform in hardware [10]. In these days, the Montgomery modulo-multiplication is one of the most efficient modulo-multiplication algorithms available. It is defined as follows:

$$\begin{aligned} Mont(X, Y) &= XYr^{-1} \bmod n. \\ &= [XY + (XY \times n' \times n)] \text{ div } r. \end{aligned} \quad (3)$$

In (3), X and Y are inputs and n is a modulo number which has been mentioned as p in (2). A positive integer, r, greater than n and relative prime to n, is usually 2^m for some positive integer m and satisfies $rr^{-1} - nn' = 1$. 40-bit n' is selected among the several values available for hardware efficiency.

III. MODULO MULTIPLIER ARCHITECTURE

A. Bit Length Optimization

For energy optimization, we reduce the number of clock cycles and gate counts of the modulo-multiplier. The proposed modulo-multiplier minimizes the product of two factors which refers energy consumption of one modulo-multiplication.

Fig. 2 describes conventional 160-bit full-word Montgomery modulo-multiplication. It shows that there are three 160-bit scalar multiplications, one addition and one division to compute one MODMUL in (3). The division can be implemented by just throwing half of low part of result. And addition has relatively small overhead in hardware complexity and computation time. Therefore, the numbers of gates and operation clock cycles mainly depend on the multiplication which is executed three times in one MODMUL.

Since the target data rate is 10Mbps[11], 40MHz operating frequency is selected to achieve the throughput with 160-bit ECC key operation. The relationships between the number of clock cycles and the gate counts through bit length are shown in Fig. 3. The number of clock cycles per operation decreases as the bit length is increased. A 160-bit multiplication needs six cycles while 10-bit multiplication requires 768 cycles per one conventional MODMUL operation. To the contrary, the gate counts are increased. Shown in Fig. 3, the energy factor, the product of the number of clock cycles and the gate counts, is minimized with the multiplier bit width between 40 and 80 bits. Since 80-bit multiplication cannot be computed in one clock cycle at 40MHz operating frequency, the proposed architecture uses two 40-bit multipliers for a low energy MODMUL operation. Though the gate counts increase, the number of clock cycles is reduced to half. This helps additional optimization of the energy factor.

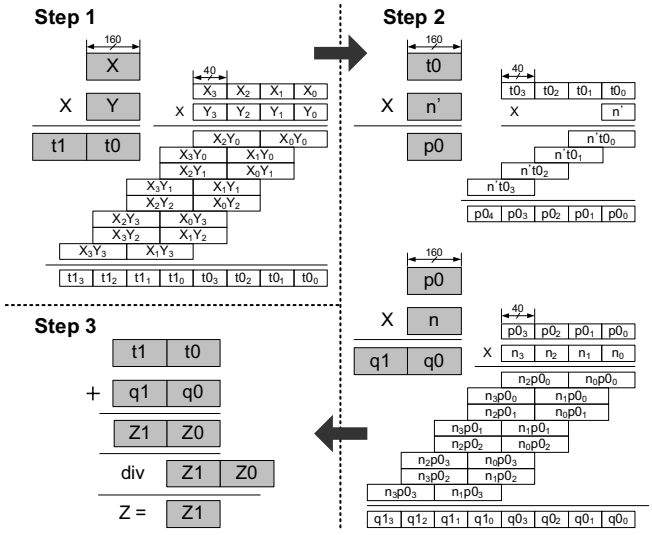

Figure 2. Conventional 160-bit full-word modulo-multiplication

B. Datapath of Modulo-Multiplication

The proposed datapath of modulo-multiplier is described in Fig. 4. Two 160-bit operands, X and Y, are inputted to the iteration path and after 20 iterations, 160-bit output Z comes out. Iteration path is composed of two radix-4 Booth multipliers, one carry save adder (CSA), three carry propagation adders (CPAs), a shifter and six pipeline registers. All of datapath are 40-bit-wide.

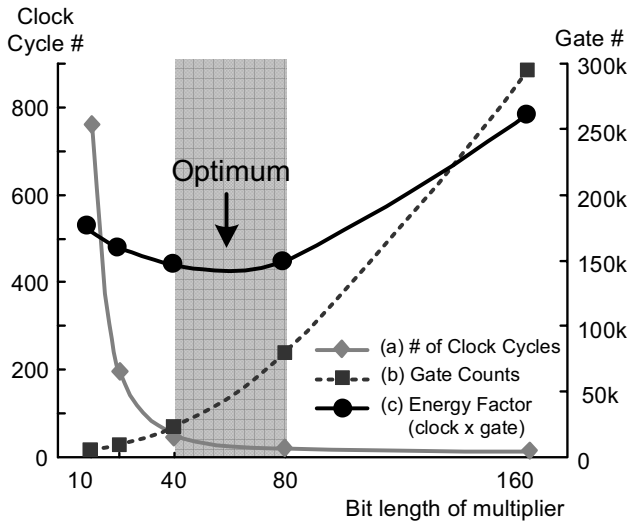

Figure 3. Analysis of energy optimization

The 160-bit operands are divided into four 40-bit partial operands. Three multiplications in step 1, 2 of Fig. 2 are sequentially executed by two 40-bit scalar multipliers, MUL A, MUL B. After that, the addition in step 3 is computed in pipeline by the CSA and three CPAs. In conventional architecture, one needs 320-bit CPA and 320-bit registers to compute 160bit-160bit multiplication. Moreover, it is necessary to keep up registers to store intermediate value until overall operation is completed. Those are 800-bit registers which are burdensome to keep energy consumption low. However, as shown in Fig. 2, only 160-bit MSB of the final result is needed. In this point of view, the proposed architecture uses a simple shifter and only six 40-bit registers to make 160-bit addition output.

The detailed procedure is illustrated in Fig. 5. Total MODMUL operation is divided into four stages. In the first stage, two 40-bit multipliers operate with the operand X and the least significant 40 bits of operand Y. Among the 200-bit outputs, 40-bit LSB, $t0_0$ in the black box in Fig. 5, is ready to make final result. However, 160-bit MSB, $t1_0$, $t0_3$–$t0_1$ in the gray box are intermediate values. Next multiplication with n' in step 2 of Fig. 3 is performed with only $t0_0$. The output of last multiplication with n in step 2 is dealt in a similar way. The fixed result, 40-bit LSB $q0_0$, is thrown away, and the intermediate value, 160-bit MSB $q1_0$, $q0_3$–$q0_1$ are shifted and stored into output registers Rz_3–Rz_0. In the second stage, the input of one multiplier is changed to next 40-bit of operand Y. The outputs of multipliers are added with previous intermediate value $t1_0$, $t0_3$–$t0_1$. After accumulation, the new value of $t0_1$ is also completed. As the sequences are followed as previous stage, the output registers of stage 2 are filled with the next intermediate values, $q1_1$, $q1_0$, $q0_3$, $q0_2$. The third stage operates in a similar way. With the shifter and six 40-bit pipeline registers, accumulation and shift operation is performed. After the last multiplication with n, the fixed output $q0_2$ is thrown away and other intermediate values are stored. Finally, in the stage 4, all of partial modulo-multiplication is finished and then we get final 160-bit MSB output. It takes 20 clock cycles at 40MHz operating frequency. The proposed architecture computes 160-bit multiplication with only 240-bit registers rather than 800-bit registers in the pipeline. This achieves 70% reduction in terms of gate size and energy.

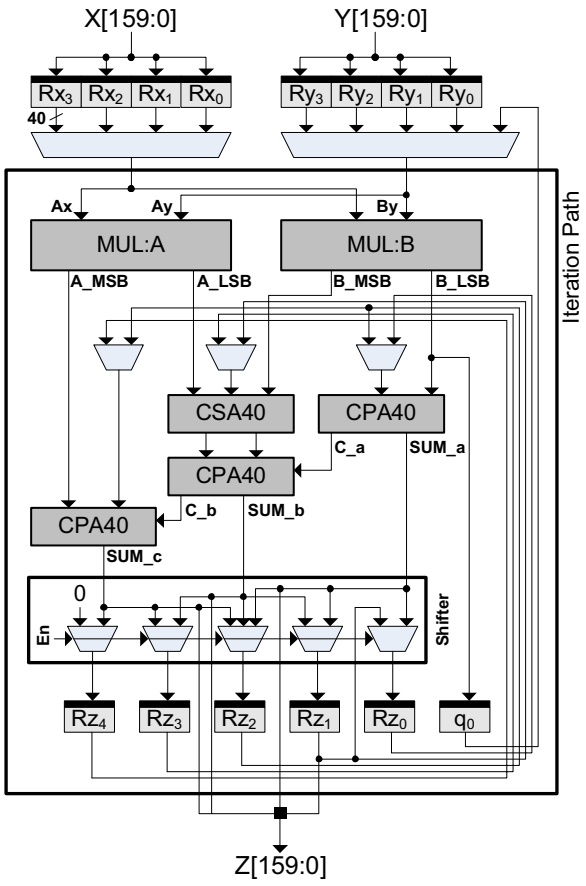

Figure 4. Datapath of the proposed 160-bit modulo-multiplier

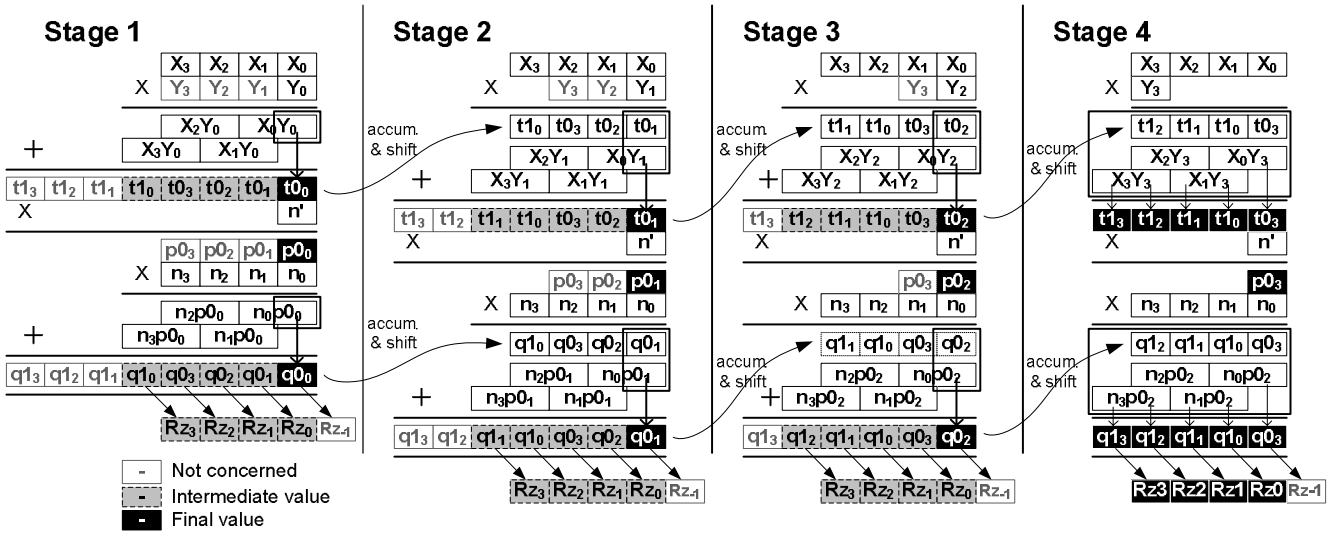

Figure 5. Proposed 160-bit partial modulo-multiplication

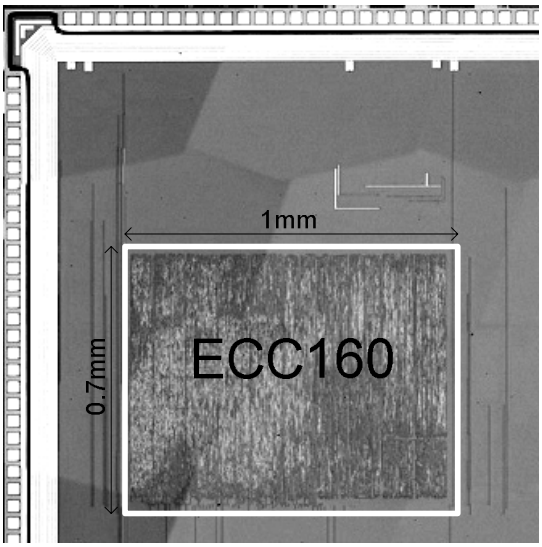

Figure 6. Chip photograph of the 160-bit modulo-multiplier

IV. IMPLEMENTATION RESULT

The low energy modulo-multiplier chip is implemented by using a 0.18-㎛ CMOS technology. Fig. 6 is a chip photograph of the modulo-multiplier which has 0.7mm x 1.0mm area. This multiplier operates at 40MHz at 1V supply voltage. It takes 0.5 ㎲ for one MODMUL operation. The energy consumption of modulo multiplier is 6.3nJ for one modulo-multiplication. Table 2 summarizes the performance results.

Table 3 shows the hardware performance comparison result of modulo-multiplication with previous work in which the field sizes are around 160 bits. These numbers cannot be compared directly because the hardware platforms are different. The proposed multiplier with two 40-bit multipliers achieves the fastest operation time of 20 cycles at 40MHz operation frequency, that is, the smallest number of cycles, using 0.18-㎛ CMOS library. And the gate counts are also relatively small. The energy consumption, 6.3nJ per modulo-multiplication, is reduced to only 52.5% of [6].

TABLE II. PERFORMANCE RESULT

Process Technology		1-poly 4-metal 0.18-㎛ CMOS technology
Power Supply		1V
Operating Frequency		40MHz
Area		0.7mm x 1.0mm
MODMUL	**Latency**	20-cycle
	Energy Consumption	6.3nJ per operation

V. CONCLUSION

A low energy modulo-multiplier is proposed for ECC processor. It is implemented in a single chip by using a 0.18-μm CMOS technology. This multiplier operates at 40MHz at 1V supply voltage, and needs 0.5 ㎲ per modulo multiplication. Its bit length is optimized for energy minimization, and the partial modulo multiplication algorithm in pipeline reduces gate counts and the number of registers. These hardware optimizations extremely reduce energy consumption of the proposed modulo-multiplier. It consumes only 6.3nJ for one modulo-multiplication.

REFERENCES

[1] Elaine Barker, William Barker, William Burr, William Polk and Miles Smid, "Recommendation for key management – Part 1: General (Revised)", NIST Special Publication 800-57, pp. 61-64, Mar 2007.

[2] H. Kim, S. Choi, H. Yoo, "A low power compression processor for body sensor network system", International Workshop on Wearable and Implantable Body Sensor Networks, vol. 13, pp. 65-69, Mar 2007.

[3] S. Xu and L. Batina, "Efficient implementation of elliptic curve cryptosystems on an ARM7 with hardware accelerator", Proc. Information Security, pp. 266-279, 2001.

[4] G. Orlando and C. Paar, "A scalable GF(p) elliptic curve processor architecture for programmable hardware", Proceeding of Workshop on Cryptographic Hardware and Embedded Systems, N.2162 in Lecture Notes in Computer Sceience, pp. 356-371, May 2001.

[5] A. Satoh and K. Takano, "A scalable dual-field elliptic curve cryptographic processor", IEEE Trans. on Computer, vol. 52, no. 4, pp. 449-460, Apr 2003.

[6] C. McIvor, M. McLoone and J. McCanny, "Hardware elliptic curve cryptographic processor over GF(p)", IEEE Trans. on Circuits and Systems, vol. 53, no. 9, pp. 1946-1956, Sep 2006.

[7] Certicom Research, "SEC 1: Elliptic curve cryptography", Certicom Corp. Standards for Efficient Cryptography, ver 1.0, Sep 2000.

[8] D. Hankerson, A. Menezes and S. Vanstone, "Guide to elliptic curve cryptography", Springer, 2004.

[9] Certicom Research, "SEC 2: Recommende elliptic curve domain parameters", Certicom Corp. Standards for Efficient Cryptography, ver 1.0, Sep 2000.

[10] P. Montgomery, "Modular multiplication without trial division", Mathematics of Computation, vol. 44, pp. 519-521, 1985.

[11] S. Song, N. Cho, S. Kim, J. Yoo, S. Choi and H. Yoo, "A 0.9V 2.6mW body-coupled scalable PHY transceiver for body sensor applications", IEEE International Solid-State Circuits Conference, vol. 50, pp. 366-367, Feb 2007.

TABLE III. COMPARISON RESULT OF MODMUL

Reference	Field	Platform	Maximum frequency	Number of cycles	Energy per operation	Notes
This work	GF(2^{160}-2^{32}-1)	0.18-㎛ CMOS	40MHz	**20**	**6.3nJ**	Two 40-bit multipliers
[6] McIvor, et al.	GF(p) 256 bits	Xilinx XC2VP125-7	45.68MHz	32	12nJ	256-bit multiplier
[3] S. Xu, et al.	GF(2^{192}-2^{64}-1)	ASIC	50MHz	40	-	ARM7 + arithmetic unit
[4] Orlando, et al.	GF(2^{192}-2^{64}-1)	Xilinx XCV1000E-8	40MHz	35	-	192-bit multiplier
[5] Satoh, et al.	GF(p) 192 bits	0.13 ㎛ CMOS	137.7MHz	45	-	64-bit multiplier