Implementation of high-speed SHA-1 architecture

Eun-Hee Lee[1], **Je-Hoon Lee**[1a)], **Il-Hwan Park**[2], **and Kyoung-Rok Cho**[1b)]

[1] *BK21 Chungbuk Information Tech. Center, Chungbuk Nat'l University*
San 12, Gaeshin-dong, Heungduk-ku, Cheongju 361–763, Rep. of Korea

[2] *National Security Research Institute, 161 Gajeong-dong, Yuseong-ku, Daejeon, Rep. of Korea*

a) *leejh@hbt.cbnu.ac.kr*

b) *krcho@cbu.ac.kr*

Abstract: This paper proposes a new SHA-1 architecture to exploit higher parallelism and to shorten the critical path for Hash operations. It enhances a performance without significant area penalty. We implemented the proposed SHA-1 architecture on FPGA that showed the maximum clock frequency of 118 MHz allows a data throughput rate of 5.9 Gbps. The throughput is about 26% higher, compared to other counterparts. It supports cryptography of high-speed multimedia data.

Keywords: cryptography, secure hash algorithm, hardware design

Classification: Integrated circuits

References

[1] NIST, *FIPS PUB 180-2, Secure Hash Standard (SHA–1)*, 1996.

[2] I. Yiakoumis, et al., "Maximizing the Hash function of authentication codes," *Potentials, IEEE*, vol. 25, no. 2, pp. 9–12, March 2006.

[3] N. Skavos, et al., "An ultra high speed architecutre for VLSI implementation of Hash functions," *Proc. ICECS*, pp. 990–993, 2003.

[4] J. M. Diez, et al., "Hash algorithms for cryptographic protocol: FPGA implementation," *Proc. TELFOR'2002*, 2002.

[5] H. Michail, et al., "Holistic methodology for designing ultra high-speed SHA-1 Hashing cryptographic module in hardware," *Proc. EDSSC 2008.*

[6] Y. K. Lee, et al., "Throughput optimized SHA-1 architecture using unfolding transformation," *Proc. ASAP*, pp. 354–359, 2006.

1 Introduction

In network supporting multimedia data service, the information security becomes a critical issue. Hash functions are widely used in message authentications and integrity protections for many wireless protocols because it dose not require the processed data to be retrieved. SHA-1 (secure Hash algorithm-1) is a representative algorithm of the SHA family [1]. Many SHA-1 implementations have been proposed to enhance the performance [2, 3, 4, 5, 6]. However, low throughput of secure data processing is a bottleneck for data

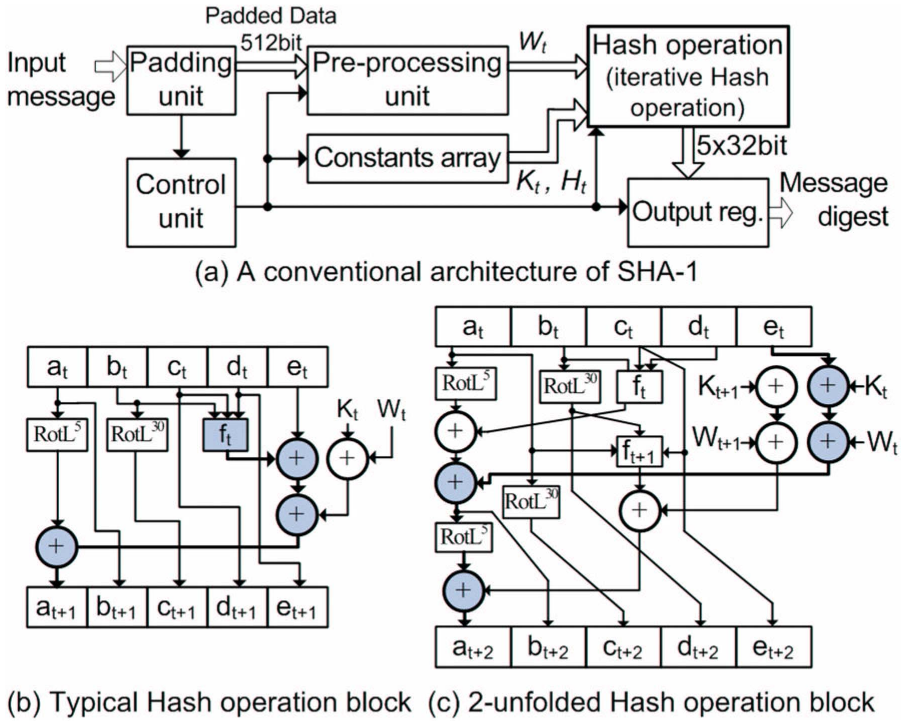
DOI: 10.1587/elex.6.1174
Received July 06, 2009
Accepted July 22, 2009
Published August 25, 2009

encryption in the multimedia service network. Y. Lee et al. [6] proposed a SHA-1 architecture employing unfolded transformation, which combines the several iterations of SHA-1 into a single cycle. This architecture enhances performance by reducing the number of required cycles for one block Hash, which achieves 3.5 Gbps as a result. H. Michail et al. [5] proposed holistic methodology that shows the maximum throughput over 4.7 Gbps with 91 MHz clock frequency. This paper proposed a high-speed SHA-1 architecture aiming high throughput. It is based on the unfolded transformation performing two Hash operations in a cycle. We shorten the critical path of Hash operation by pre-computation of the coefficients and its parallelism.

2 SHA-1 algorithm and related works

SHA-1 algorithm takes input message with a maximum length of less than 2^{64} bits and produces a 160-bit message digest. The conventional architecture of SHA-1 is shown in Fig. 1 a. In Fig. 1 a, the input message split in 80 × 32-bit words and it requires 4 rounds of Hash operation and each round performs 20 operations iteratively. Main differences among the rounds are a scrambling constant, K_t, a nonlinear operation, F_t, and 32-bit dataword, W_t for each Hash operation.

H. Michail presented cost function analysis for the unfolded SHA-1 algorithm. As a result, the best achieved throughput/area ratio was obtained by partially unfolding two operations. Y. Lee presented an unfolded SHA-1 design employing two unfolded SHA-1 as shown in Fig. 1 c instead of the conventional SHA-1 as shown in Fig. 1 b. It reduces the number of cycles by half because it performs two Hash operations in one cycle. However, it



(a) A conventional architecture of SHA-1

(b) Typical Hash operation block (c) 2-unfolded Hash operation block

Fig. 1. Conventional architecture of SHA-1

© IEICE 2009
DOI: 10.1587/elex.6.1174
Received July 06, 2009
Accepted July 22, 2009
Published August 25, 2009

increases the critical path delay in Hash operation block. The critical path in two-unfolded Hash operation has delay of 4 additions as shown in Fig 1 c.

3 The proposed SHA-1 architecture

The proposed SHA-1 design employs two-unfolded architecture. We propose coefficient pre-computation of Hash function and parallelism available in two Hash operation blocks. The outputs of Hash operation are shown in Eq. (1).

$$\begin{aligned}
a_{t+2} &= RotL^5\{RotL^5(a_t) + f_t(b_t, c_t, d_t) + e_t + W_t + K_t\} \\
&\quad + f_{t+1}(a_t, RotL^{30}(b_t), c_t) + d_t + W_{t+1} + K_{t+1} \\
b_{t+2} &= RotL^5(a_t) + f_t(b_t, c_t, d_t) + e_t + W_t + K_t \\
c_{t+2} &= RotL^{30}(a_t) \\
d_{t+2} &= RotL^{30}(b_t) \\
e_{t+2} &= c_t
\end{aligned} \tag{1}$$

where $RotL^x(y)$ represents the left rotation of y by x, and $f_t(o, p, q)$ stand for non-linear function at time $t (t = 0, 2, 4, \ldots)$. The outputs c_{t+2}, d_{t+2}, e_{t+2} are directly derived from a_t, b_t, c_t, whereas the other outputs, a_{t+2} and b_{t+2}, require the computational result of a_{t+1} as shown in Eq. (1).

We propose pre-computation for Hash coefficients that makes higher parallelism in Hash operation. We newly define three parameters l_t, m_t, and n_t that are presented in Eq. (2). These terms are pre-computed before computing other parameters. Note that, l_t is used to compute a_{t+2} and b_{t+2}; m_t and

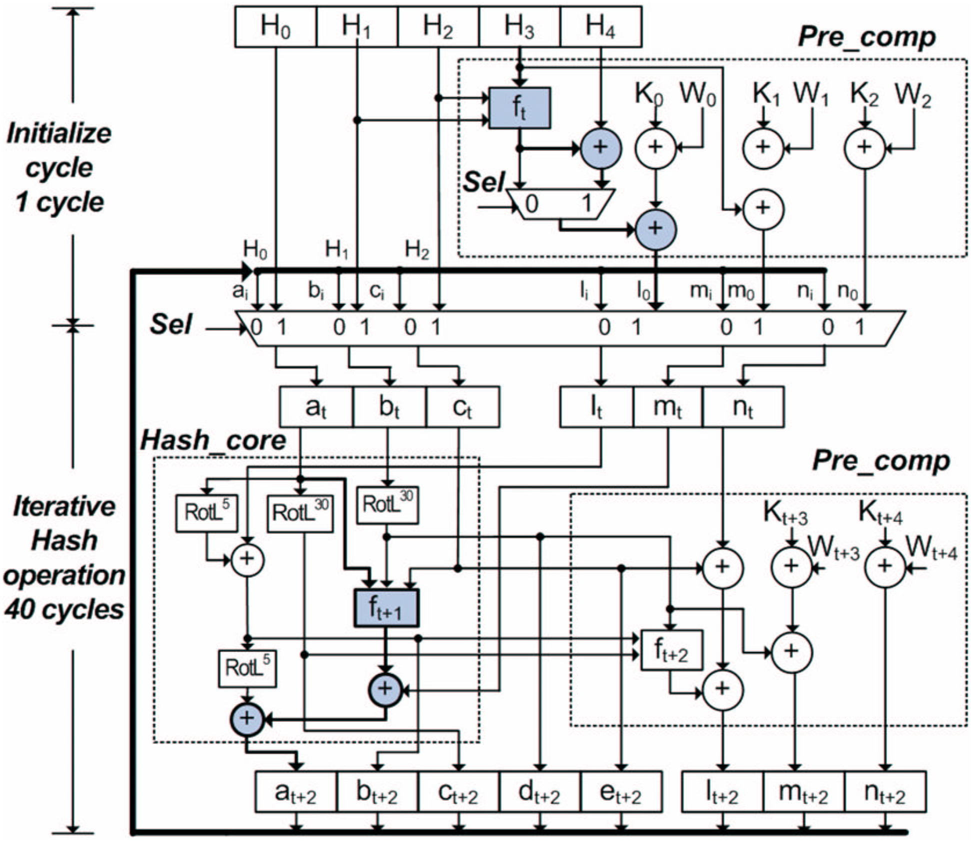


Fig. 2. The proposed SHA-1 Hash operation block

© IEICE 2009
DOI: 10.1587/elex.6.1174
Received July 06, 2009
Accepted July 22, 2009
Published August 25, 2009

n_t are used to compute a_{t+2} and l_{t+2}, respectively.

$$\begin{aligned} l_t &= f_t(b_t, c_t, d_t) + e_t + W_t + K_t \\ m_t &= d_t + W_{t+1} + K_{t+1} \\ n_t &= W_{t+2} + K_{t+2} \end{aligned} \tag{2}$$

Thus, a_{t+2} and b_{t+2} in Eq. (1) can be modified to Eq. (3) using pre-computed parameters. The pre-computed parameters are fed with other inputs such as a_t, b_t, and c_t simultaneously. The critical path delay computing a_{t+2} is dramatically decreased because l_t and m_t is pre-computed.

$$\begin{aligned} a_{t+2} &= RotL^5\{RotL^5(a_t) + l_t\} + f_{t+1}(a_t, RotL^{30}(b_t), c_t) + m_t \\ b_{t+2} &= RotL^5(a_t) + l_t \end{aligned} \tag{3}$$

The proposed SHA-1 architecture is composed of Pre_comp and Hash_core as shown in Fig. 2. Pre_comp is responsible for the pre-computation of the newly defined terms l_t, m_t, and n_t before next stage of Hash operation. Among these parameters, l_t has the longest delay that is two additions and one non-linear function, f_t. Hash_core is responsible for the computation of $n - th$ Hash operation using the values from a_t to e_t as well as the previous outputs of Pre_comp. In these outputs, a_{t+2} has the longest delay that is two additions and one non-linear function. In addition, there is not any data dependency between Pre_comp and Hash_core that it makes possible parallel computation. Consequently, the critical path for Hash operation block is converged to the delay of two additions and one non-linear function, f_t.

The proposed SHA-1 architecture requires 41 cycles to generate message digest. The first cycle is to initialize the newly defined terms l_t, m_t, and n_t. The other 40 cycles are required for iterative Hash functions. During this iterative Hash operation, the proposed SHA-1 processes Pre_comp and Hash_core in parallel. The outputs of Pre_comp and Hash_core are independent at same cycle because Pre_comp is the former computation of Hash_core. The select signal, *Sel* is used to identify either the first initialized cycle or the other iterative cycles. It selects the input of Hash_core as shown in Fig. 2. When *Sel* is *high*, Pre_comp generates the newly defined terms l_0, m_0, and n_0 using the constant values W_t and K_t. The initial values are from H_0 to H_4, which are the constant of SHA-1 algorithm. Thus, the proposed implementation needs total of 41 cycles for completing of Hash operations. Although it requires one extra cycle comparing to the conventional two unfolded SHA-1, it dramatically shortens the critical path delay in Hash operation block.

4 Performance analysis

We evaluate the proposed SHA-1 architecture on Xillinx Vertex-2 FPGA, xc2v1000 and compare it with the other counterparts. The design was fully verified using a large set of test messages that is the test example by the standard [1]. Table I shows the comparison results with the other counterpart SHA-1 implementations based on same FPGA technology, Xillinx Vertex-2 family [2, 3, 4, 5, 6]. For the better comparison, we referred to the data sheet of each FPGA chip and found out how many slices are occupying. The

DOI: 10.1587/elex.6.1174
Received July 06, 2009
Accepted July 22, 2009
Published August 25, 2009

proposed architecture with pipelines and the unfolding factor two in Figure 2 shows maximum operating frequency 118 MHz and throughput 5.9 Gbps.

In terms of circuit size, a conventional SHA-1 architecture without both pipelining and unfolding scheme [2] shows the smallest hardware size, 854 slices, among all listed works in Table I. The proposed SHA-1 design has about 2,900 Slices that it is smaller than the other ones employing both pipelining and unfolding architecture [5, 6]. As a result, the proposed SHA-1 reduces the area over 32% compare with these unfolded architectures.

The proposed SHA-1 architecture brings advantages in both maximum operational frequency and throughput as shown in Table I. The SHA-1 designs without pipelining and unfolding [2, 3] shows the maximum throughput to 1.3 Gbps. Instead of that, the SHA-1 employing the alternative of unfolding transformation or pipeline architecture shows 900 Mbps and 2.5 Gbps. The work presented by Y. K. Lee et al. and by H. Michail et al. showed 3.5 Gbps at the 41.5 MHz clock frequency and 4.7 Gbps at the 91 MHz clock frequency. Both of them employ unfolded transformation and pipeline architecture. Our implementation works flawlessly at 118 MHz clock that achieves maximum throughput of 5.9 Gbps. It is due to shorten the critical path of iterative Hash operation. The work presented by H. Michail et al. is the fastest architecture ever presented implementation. In the work, the longest path delay T_{pd} is 3 $\times T_{add}$ and one-non linear function. On the other hand, T_{pd} of the proposed SHA-1 implementation is 2 $\times T_{add}$ and one-non linear function. Even though our implementation requires 41 cycle including initializing cycle for pre-computation, it increases throughput by 26% comparing to the previous work [5].

Table I. Comparison results with the other SHA-1 designs

Design	*Frequency (MHz)*	*Throughput (Gbps)*	*Area (Slices)*	*Number of cycles*
[2]	162	1.0	854	80
[3]	55	1.3	8,980	80
[4]	39	0.9	1,550	22
[5]	91	4.7	4,848	40
[6]	42	3.5	4,258	24
$Proposed\ SHA-1$	118	5.9	2,894	41

5 Conclusions

SHA-1 is a popular Hash algorithm and suitable for high-speed crypto graphing. We proposed a new architecture of SHA-1 reducing critical path that enhances throughput of Hash algorithm. We implemented the proposed SHA-1 architecture on FPGA chips. The proposed implementation works at 118 MHz clock that gives the maximum throughput of 5.9 Gbps. As a result, the proposed architecture shows more than 26% better throughput with 32% smaller hardware size compared to the previous implementations. The high-speed SHA-1 is useful to generate a condensed message and may strengthen the security of mobile communication and internet service.

DOI: 10.1587/elex.6.1174
Received July 06, 2009
Accepted July 22, 2009
Published August 25, 2009

Acknowledgement

This research was financially supported by the Ministry of Education, Science Technology (MEST) and Korea Industrial Technology Foundation (KOTEF) through the Human Resource Training Project for Regional Innovation.

DOI: 10.1587/elex.6.1174
Received July 06, 2009
Accepted July 22, 2009
Published August 25, 2009