Design of a High Speed SHA-1 Architecture Using Unfolded Pipeline for Biomedical Applications

Eun-Hee Lee, Seok-Man Kim, Je-Hoon Lee, Kyoungrok Cho
Dept. of ECE, Chungbuk National University
12 Gaeshin-dong Heungduk-ku, Cheongju City, Korea 361-763
ehlee@hbt.cbnu.ac.kr, krcho@cbnu.ac.kr

ABSTRACT

Hash functions are a special family of cryptographic algorithms, which are applied widely wherever message integrity and authentication issues are critical. It is used in many security protocol like that TLS, SSL, PGP, SSH and IPSec. In this paper, we propose a high performance Hash coding architecture that computes next round coefficient in the present round in parallel. We optimize the critical path and hence the clock frequency by reorganizing of computation ordering of the functional blocks. In addition, we employ a pipeline architecture that computes Hash code for 512 bits block data in each clock hence high throughputs becomes possible. The proposed architecture provides more than 10% performance improvement as well as a 20% reduction in hardware size when compared with the previous implementations [2, 5, 8, 9, 10, 11 and 12].

Keywords: Hash function, security throughput, high speed, cryptographic, SHA-1

1. INTRODUCTION

In recent years effective and secure communications in networks and programs within the consumer electronics area is emerging as a critical part of systems implementation. Cryptographic Hash functions are important in modern cryptography. A typical technique is to extract Hash code of messages, IP packets and disk files which results in a large, possibly variable-sized amount of data to be converted into a small but fixed datum. If a message is signed in by a Hash function, the computation load, memory and transmission load become low. In other words, if a Hash code is made from an input data, it is impossible to find the original input data from the Hash code. Data origin authentication and message integrity can be obtained if public-key or symmetric-key encryption is applied to the Hash code. Hash functions have key features such as one-way function property, collision resistance and their high speed coding.

Data sizes and communication speeds are dramatically increasing every year. Therefore, low throughput of Hash functions can be a bottleneck in the digital and/or communications systems. In order to solve this problem, unfolding [4] and pipeline technique [5] are applied. Unfolding technique decreases the need for clock cycles and a pipeline technique makes the critical path shorten. However, unfolding factor increases critical path delay in a computation loop.

In this paper, we present an optimized architecture for Hash coding that computes the next stage coefficient in the present stage in parallel. This paper is organized as follows: In section II past implementation strategy of the SHA-1 is presented. Sections III provides our proposed architecture. Section IV shows results of throughput and area of the new SHA-1 architecture when compared with other implementations. Finally, conclusions are in section V.

2. CONVENTIONAL SHA-1

Secure Hash Algorithm (SHA) and its improve version [1] was developed by NIST (National Institute of Standards and Technology). According to SHA standard, SHA-1 requires 4 iterations of 20 operations that results in a total of 80 operations to generate the message digest for 512 bits data. The architecture of two consecutive operations of Hash function is illustrated in Figure 1

where each and all a_t, b_t, c_t, d_t, e_t are 32-bit which resulting in 160 bit Hash code output. Eq. (1) shows the model for one single operation. The scrambling constant k_t, and a non-linear function f_t are updated in each round operation. The constants k_t and w_t are used for t iteration of the Hash computation. $ROTL^n(x)$ indicates rotation of x by n positions to the left. $f_t(x,y,z)$ is a non-linear function of SHA-1 which is usable on operation t.

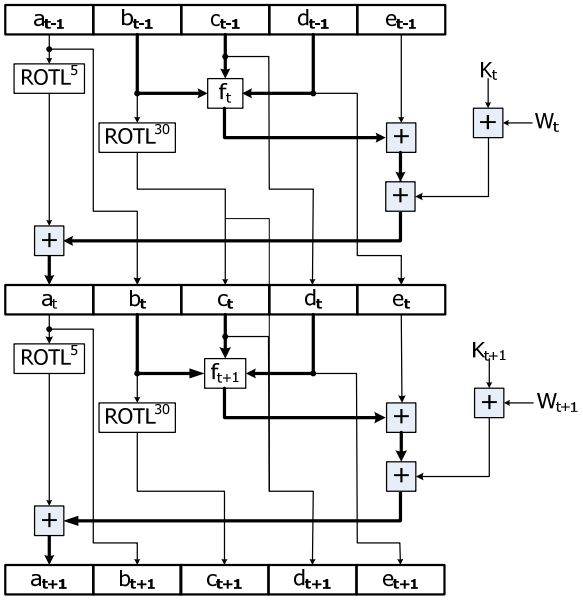


Figure 1. Architecture of 2 consecutive SHA-1

$$
\begin{aligned}
a_t &= ROTL^5(a_{t-1}) + f_t(b_{t-1}, c_{t-1}, d_{t-1}) + e_{t-1} + w_t + k_t \\
b_t &= a_{t-1} \\
c_t &= ROTL^{30}(b_{t-1}) \\
d_t &= c_{t-1} \\
e_t &= d_{t-1}
\end{aligned} \qquad (1)
$$

Inputs a_{t-1}, b_{t-1}, c_{t-1}, d_{t-1}, e_{t-1} illustrated in Figure 1 go through two consecutive SHA-1 functional blocks that result in outputs a_{t+1}, b_{t+1}, c_{t+1}, d_{t+1} and e_{t+1}. Outputs a_t, b_t, c_t, d_t, e_t from the first SHA-1 computation block become inputs for the second SHA-1 computation block. Signals b_t, c_t, d_t, e_t except a_i, are derived from the inputs a_{t-1}, b_{t-1}, c_{t-1}, d_{t-1} respectively. This implies that c_{t+1}, d_{t+1} and e_{t+1} can be derived from a_{t-1}, b_{t-1}, c_{t-1} respectively. Calculations for a_{t+1} and b_{t+1} require inputs d_{t-1} and e_{t-1}, respectively. Also, k_t, w_t and k_{t+1}, w_{t+1} are known constants available prior the commencement of computation. Therefore the two computations can be performed concurrently. This approach is illustrated in Figure 2. The dotted area on Figure 2 indicates part of SHA-1 block that operates in parallel and highlights the critical path which has only an addition. The b_{t+1} with three addition levels is computed in parallel with a_{t+1} that is two addition. Thus, to compute a_{t+1} completely, an extra addition stage is required.

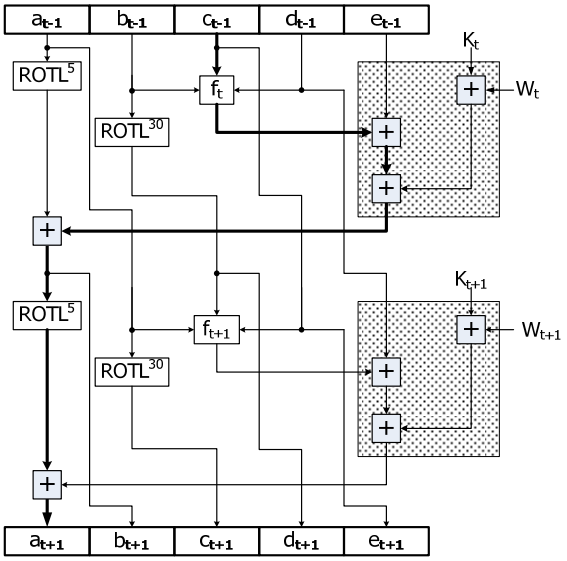


Figure 2 SHA-1 operation block that unfolding factor is 2

Although, the architecture in Figure 2 reduces the maximum operating frequency due to increasing critical path, the throughput is significantly increased because # of operations and # of used clocks are reduced by half. The throughput is given as Eq. (2).

$$Throughput = \frac{\#bits \times f_{operation}}{\#operations} \qquad (2)$$

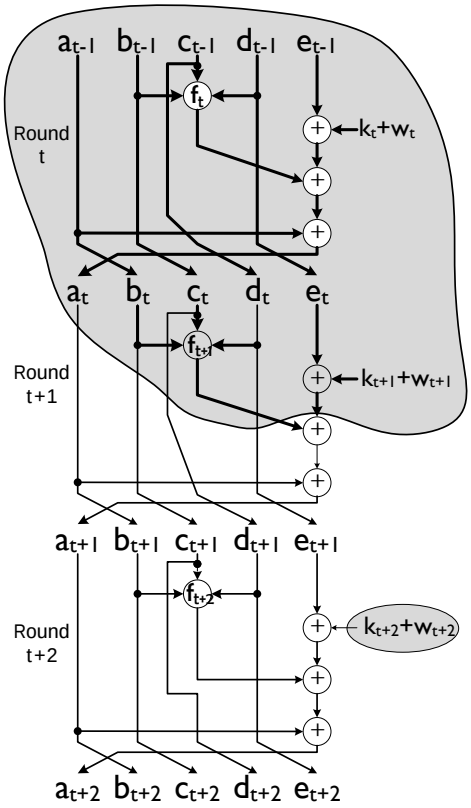
From the Eq. (2), the expected operating frequency for Figure 2 will be higher about 25% since the critical path

has been changed from three to four addition levels comparing to non-concurrent implementation. However, Hash code in the Figure 2 is computed in only 40 clock cycles instead of 80 in the non-concurrent implementations. This computation leads to the result that throughput of the Figure 1 increases by 35%.

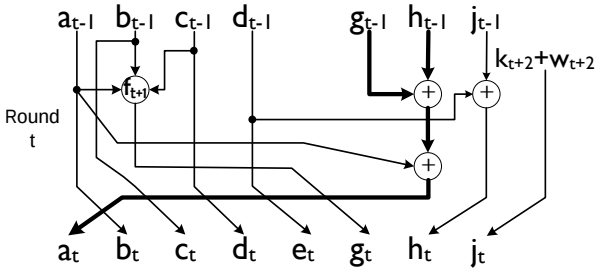
Critical path delay is increased by the unfolding factor thus degrading the throughput. Therefore improvement comes through SHA-1 architecture which employs the pipeline technique to improve higher throughput. SHA-1 architecture is based on four stages pipeline [5]. The SHA-1 architecture requires a different non-linear function every 20 clock cycles. Taking on design elements from [4], the four functional blocks are iteratively used during every 20 clock cycles. The architecture allows parallel operation of the four functional blocks that introduces a 20 cycle latency which is four times the throughput of those reported in [4] [5] [11] and [12]. Furthermore, power dissipation and area penalty are kept low compare with the implementation in [3].

3. THE PROPOSED DESIGN

The conventional implementations [6][7][8][11] and [12] employ four stages pipeline for a function block. Throughput is increased by a factor of 4 since a Hash code is produced every 20 clock cycles instead of 80. The computation of a_t forms the critical path as it requires three additions and one non-linear function delay as shown in Figure 1. Decreasing the critical path delay makes higher throughput. The modified architecture for SHA-1 is illustrated in Figure where Figure 3(a) shows a conventional architecture with three consecutive SHA-1 blocks. The signals a_{t-1}, b_{t-1}, c_{t-1}, d_{t-1} are initial constant of H0~H4. The w_t and k_t also are predictable. Except for a_t, the other signals b_t, c_t, d_t, e_t are derived directly from the inputs a_{t-1}, b_{t-1}, c_{t-1}, d_{t-1} respectively. In addition, e_{t+1} is derived directly from the input c_{t-1}. The signal of $t+2$ round is able to be obtained in t round. f_{t+1} is a computation of b_t, c_t, d_t which is derived through inputs a_{t-1}, b_{t-1}, c_{t-1}. This means that f_{t+1} used in $t+1$ round is able to compute in t round. The critical path for conventional approach is $3Ta$, where Ta delay of adder block. On the other hand, the optimized circuit illustrated in Figure 3(b) shows the critical path being only $2Ta$ being reduced by one adder path delay. Figure 3(b) is an optimized circuit of the shaded area in Figure 3(a).



(a) A conventional architecture with three



(b) A optimized architecture

Figure 3. Design flow for the proposed architecture

Figure 4 shows architecture with an unfolding factor two that computes SHA-1 in 40 clock cycles. Even though it has an unfolding factor of two, the critical path is $3Ta$. Comparing with Figure 3(b) where the unfolding factor is one with 80 clocks, the critical path delay adds an

additional Ta. The architecture with unfolding factor two uses 40 Ta clock cycles that results in an increase of throughput by some 35%. If we apply pipeline technique to SHA-1, the throughput increases by a factor of 4.

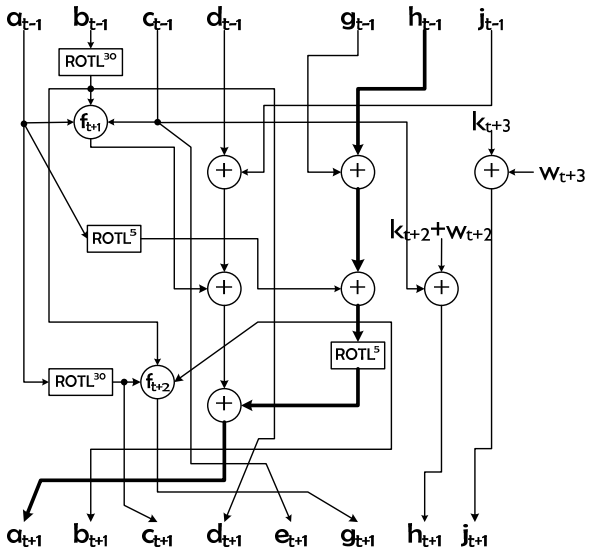


Figure 4 proposed SHA-1 operation block

However, it takes 40 clock cycles in four stage pipelines as one round uses 10 clock cycles. Thus, we have to wait 10 clocks to get the Hash code of the next data block. We modified the architecture that each pipeline stage requires a clock cycle. Although the architecture increases area, the throughput will be significantly higher.

4. RESULTS AND COMPARISONS

We evaluate the proposed SHA-1 architecture on XININX 2 Family FPGA xc2v1000. The design was fully verified using a large set of test messages that is the test example proposed by the standard [1]. In Table 1, the proposed implementations are compared to the implementation of [2], [5], [8], [9], [10], and [11]. The proposed_1 with 4 stage pipelines and the unfolding factor two in Figure 4 shows maximum operating frequency and throughput 98.3MHz and 5.0Gbps, respectively. As a result, there is 10% increase of the throughput compared to the previous best implementation. The proposed_2 is a fully pipelined architecture with unfolding factor of two that each pipeline stage requires one clock cycle. The architecture shows 10 times higher throughput than the conventional ones.

Table 1 Performance characteristics and comparisons

Implemen -tation	Freq. (MHz)	Throughput (Mbps)	Area (CLBs)
[2]	38.6	900.0	1550
[5]	55.0	1339	2245
[8]	55	2816	-
[9]	98.7	2526.7	950
[10]	127.4	3261	-
[11]	41.5	3541	4258 Slices
[12]	91	4715.0	1212
Proposed_1	98.3	5032.9	3442 Slices
Proposed_2	98.3	50329	28741 slices

5. CONCLUSION

SHA-1 is a popular Hash algorithm and is suitable for high speed crypto graphing. In this paper, we proposed a new architecture reducing critical path of SHA-1 function block that enhances speed of Hash algorithm. As the results, the proposed architecture shows more than 10% performance up and 20% smaller hardware size compared with the previous implementations. The high speed SHA-1 is able to speed up either table lookup or data comparison tasks for biomedical applications. The SHA-1 can be used to generate a condensed message. The high-speed Hash algorithm strengthens the security of mobile communication and internet service.

6. REFERENCES

[1] FIPS PUB 180-2, Secure Hash Standard (SHA-1), National Institute of Standards and Technology (NIST), 1996

[2] J.M., Diez, S., Bojanic, C., Carreras, and O., Nieto-Taladriz, “Hash Algorithms for Cryptographic Protocols: FPGA Implementation”, in Proc. of TELEFOR, 2002

[3] S., Dominikus, “A Hardware Implementation of D-4 Family Hash Algorithms”, in Proc. of ICECS, pp. 24-27, 2003

[4] G., Selimis, N., Sklavos, and O., Koufopavlou, “VLSI Implementation of the Keyed-Hash Message Authentication Code for the Wireless Application Protocol”, in Proc. of ICECS, pp. 24-27, 2003.

[5] N., Skavos, G., Dimitroulakos, and O., Koufopavlou, “An Ultra High Speed Architecture for VLSI Implementation of Hash Functions”, in Proc. of ICECS, pp. 990-993, 2003

[6] R. Lien, T. Grembowski, K. Gaj, “A 1 Gbit/s Partially Unrolled Architecture of Hash Functions SHA-1 and SHA-512”, in Proc. of LNCS,vol. 2964, pp. 324-338, 2004

[7] H.E. Michail, A.P. Kakarountas, Georgee N. Selimis, and Costas E. Goutis, " Optimizing SHA-1 Hash Function for High Throughput with a Partial Unrolling Study", in Proc. of PATMOS, pp. 591-600, 2005

[8] H.Michail, A.P. Kakarountas, O. Koufopavlou and C.E. Goutis, “A Low-Power and High-Throughput Implementation of the SHA-1 Hash Function", in Proc. of ISCAS, pp. 23-26 May, 2005

[9] A.P. kakarountas, G. Theodoridis, T. Laopoulos, and C.E Goutis "High-Speed FPGA Implementation of the SHA-1 Hash Function", in Proc. of IDAACS, pp. 211-215, September 2005.

[10] Michail, H.E. Kakarountas, A.P. Milidonis, A.S. Panagiotakopoulos, G.A. Thanasoulis, V.N. Goutis, C.E. “Temporal and System Level Modifications for High Speed VLSI Implementations of Cryptographic Core” in Proc. of ICECS, pp. 1180-1183, 2006

[11] Y. K. Lee, H. Chan and I.Verbauwhede, "Throughput Optimized SHA-1 Architecture Using Unfolding Transformation", in Proc. of ASAP, pp. 354-359, 2006.

[12] Michail, H. Goutis, C., “Holistic Methodology for Designing Ultra High-Speed SHA-1 Hashing Cryptographic Module in Hardware”, in Proc. of EDSSC, pp. 1-4, 2008.