High Speed and Throughput Evaluation of SHA-1 Hash Function Design with Pipelining and Unfolding Transformation Techniques

Shamsiah binti Suhaili[1], Takahiro Watanabe[2] and Norhuzaimin Julai[1]
[1]*Faculty of Engineering, Universiti Malaysia Sarawak, 94300 Kota Samarahan.*
[2]*Graduate School of Information, Production and System, Waseda University, Kitakyushu-shi, Fukuoka, 808-0135 Japan.*
sushamsiah@unimas.my

***Abstract*—In recent years, designing of SHA-1 hash function has become popular because it was important in security design application. One of the applications of SHA-1 hash function was HMAC where the architecture of SHA-1 needed to be improved in terms of speed and throughput in order to obtain the high-performance design. The objective of this project was to design high speed and throughput evaluation of SHA-1 hash function based on a combination of pipelining and unfolding techniques. By using both techniques in designing the architecture of SHA-1 design, the speed of SHA-1 hash function can be increased significantly as well as throughput of the design. In this paper, five proposed SHA-1 architectures were designed with different stages of pipelining such as 1, 4 and 40 stages. The results showed the high-speed design of SHA-1 design can be obtained by using 40 stages pipelining with unfolding factor two. This design provided a high-speed implementation with maximum frequency of 308.17 MHz on Arria II GX and 458.59 MHz on Virtex 5 XC5VLX50T. Furthermore, the throughput of the design also increased about 150.269 Gbps and 223.618 Gbps on Arria II GX and Virtex 5 XC5VLX50T respectively. Thus, high-speed design of SHA-1 hash function was successfully obtained which can give benefit to society especially in security system data transmission and other types of hash functions.**

***Index Terms*—FPGA; Pipelining; SHA-1 Hash Function; Unfolding Transformation.**

I. INTRODUCTION

Cryptography is the the science of writing messages in secret codes; no one can read an encrypted message except the intended receiver. There are three types of cryptographic algorithms: symmetric cryptographic (secret key), asymmetric cryptographic (public key), and hash function. Cryptographic hash function or known as message digest algorithm is an algorithm that translates a random string of characters into hash code output [1]. Hash function is widely used for cryptographic application and it is useful for message authentication because it is strong enough against collision. There are different types of hash function such as SHA-1, MD5, RIPEMD160, and others. This paper focuses on cryptographic SHA-1 hash function because it is simple and is widely used in digital signature applications, Keyed-hash Message Authentication Codes (HMACs) and Hash-based Key Derivation Functions (Hash-based KDFs) [2]. Efficient implementation of cryptographic hash function on reconfigurable hardware is one of the problems that need to be solved. Thus, high performance of hash function design in terms of speed, and area is important to improve the throughput of the hash function design since nowadays all the security system need fast implementation. Maximum frequency needs to be considered in designing hash function in order to increase the speed of the design. Besides, efficient hash function also has small area implementation. Thus, in order to improve the speed and area, this paper proposes the architecture of SHA-1 40 stages pipelining with unfolding factor 2 transformations. Besides, this method can also be implemented to other hash functions to improve the speed of the design.

There are many researchers have been designed SHA-1 algorithm based on FPGA. Some of them tried to increase the frequency and throughput of the SHA-1 design [3-14]. J. Kim [3] proposed unfolding transformation with 4 stages pipelining whose throughput was 7.4 Mbps. E. H. Lee [10] and H. Michail [12] proposed SHA-1 design whose throughputs were 5.9 Gbps and 4.7 Gbps respectively with 4 stages pipelining. L. Jiang [5] improved the throughput by applying the unfolding transformation with 40 stages pipelining. This structure gave 76.2 Gbps of throughput but large area implementation which was about 32048 total registers and 33764 ALUTs. Thus, this paper proposed a combination of pipelining and unfolding transformation to improve the performance of SHA-1 hash function.

II. FUNDAMENTALS OF SHA-1 ALGORITHM

SHA-1 hash function algorithm takes arbitrary input message with the length of the message less than 2^{64} bits. The output of hash function is called message digest or hash code. SHA-1 hash function produces 160 bits output of hash code with the processing of 512 bits message inputs and 160 bits initial value. There are several steps need to be considered in order to produce 160-bit SHA-1 hash output. First, the message needs to be padded, so its length is congruent to 448 modulo 512. After that, the input message will be followed by single 1-bit input at the end of the message, and then 0 bits are padded until the last 64-bit which is the length of the message. After padding the message, the inputs of 64 bits are appended to the message. Therefore, overall message padding is 512 bits together with the length of the message. Table 1 shows buffer initialization of SHA-1 hash function in hexadecimal. There are five different inputs buffer initialization such as *A*, *B*, *C*, *D*, and *E*. These inputs are fixed for any kind of SHA-1 hash function.

After initialization process, the next step is to process the 512-bit input message which is in 16 blocks with 32-bit per block. Figure 1 illustrates 80 steps of message processing

with 20 steps for four different non-linear function of SHA-1 hash function. Each step function has the same structure but different non-linear function, *f*, constant value, *K* and message, *W*. Table 2 and Table 3 show the non-linear function and constant value, *K* of SHA-1 hash function respectively.

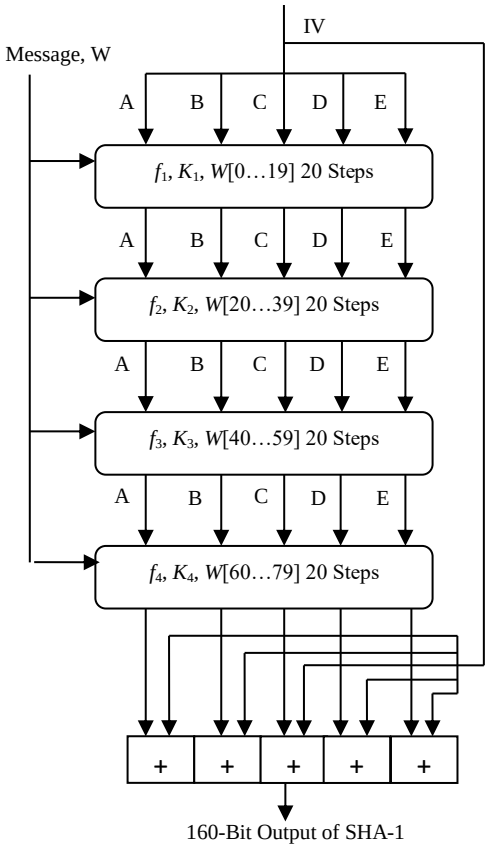

Figure 1: SHA-1 Algorithm

Table 1
Buffer Initialization of SHA-1 Algorithm

Register	Buffer Initialization (Hex)
A	32'h67452301
B	32'hefcdab89
C	32'h98badcfe
D	32'h10325476
E	32'hc3d2e1f0

Table 2
Non-Linear Function of SHA-1 Algorithm

Step Number	Function F(B,C,D)
$0 \le t \le 19$	$(B \wedge C) \oplus (\neg B \wedge D)$
$20 \le t \le 39$	$B \oplus C \oplus D$
$40 \le t \le 59$	$(B \wedge C) \oplus (B \wedge D) \oplus (C \wedge D)$
$60 \le t \le 79$	$B \oplus C \oplus D$

Table 3
Constant, K_t of SHA-1 Algorithm

Step Number	Constant, K_t (Hexadecimal)
$0 \le t \le 19$	32'h5a827999
$20 \le t \le 39$	32'h6ed9eba1
$40 \le t \le 59$	32'h8f1bbcdc
$60 \le t \le 79$	32'hca62c1d6

Message inputs for SHA-1 algorithm are taken from the first 15 messages which are directly from data input and the rest of message input will be using the following Equation (1). *ROTL*^{a}(b) denotes rotation to the left of *b* by *a* position. Equation (2) provides the output for five different registers *A, B, C, D, and E*. Figure 2 illustrates the compression function of SHA-1 algorithm. The output *A* is the crucial part where it consists of non-linear function, input message, *W*, constant, *K*, input *E* and shift *A* to the left by 5. Finally, buffer initialization input will be added to the last output rounds to produce message digest of SHA-1 hash function.

$$W_t = ROTL^1(W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) \quad (1)$$

$$T = (A \ll 5) + f(B,C,D) + W_t + K_t + E$$
$$A = T,\ B = A,\ C = B \ll 30,\ D = C,\ E = D \quad (2)$$

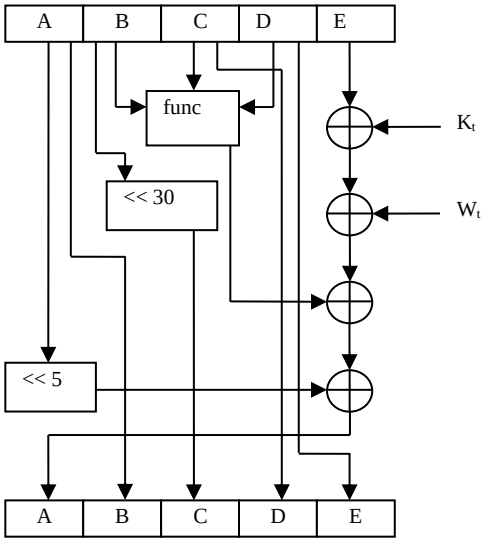

Figure 2: Compression Function of SHA-1 Algorithm

III. PROPOSED SHA-1 DESIGN

In this project, iterative and pipelining techniques are applied to SHA-1 hash function algorithm in order to increase the speed of the design [15]. Several methods have been proposed such as iterative, inner-round pipelining and outer-round pipelining. In order to improve the throughput of SHA-1 design, unfolding transformation technique was proposed [16]. Unfolding algorithm is one of the techniques that can be used by DSP application to obtain a new program that performs more than one iteration of the original program. The number of iterations is introduced as *unfolding factor, J*. By applying these techniques to SHA-1 architecture, the high performance of SHA-1 hash function can be obtained.

There are five proposed SHA-1 to be designed and evaluated as follows: (1) iterative SHA-1, (2) inner-round pipelining SHA-1, (3) 4 stages pipelining SHA-1, (4) 40 stages pipelining SHA-1 and (5) 40 stages pipelining with unfolding transformation factor 2 SHA-1. These architectures are designed using Verilog code and synthesized with both Altera and Xilinx tools. Figure 3 shows the architecture of 4 stages pipelining with five different inputs. This figure can be used to construct the architecture of 40 stages pipelining with unfolding factor two where each stage consists of two rounds. Some modification has been made in terms of input and output of the modules to obtain the output of specific proposed design. This design starts with buffer initialization which is init module. In this module, there are five different

fixed values of buffer initialization values as mentioned in Table 1. In order to execute schedule module, 80 rounds of counter module need to be designed. Schedule module is a module that executes message input to step function. The message input for the first 15 data input of SHA-1 comes from 512-bit message input which has been divided into 15 blocks of 32-bit message input. In this schedule module, the message after 15 rounds will be calculated using the formula given in Equation (1).

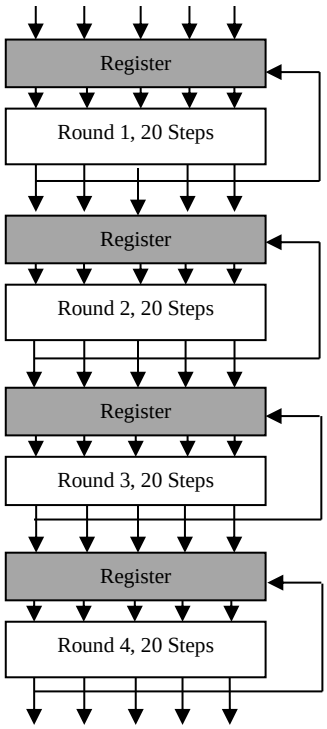

Figure 3: 4 Stages of SHA-1 Algorithm Architecture

A. Compression Function of Unfolding SHA-1 Algorithm

The main objective of this project is to obtain high-speed design. Therefore, four and forty stages of pipelining SHA-1 design are proposed. By adding registers at different stages of round can improve the maximum frequency of the design. Four registers are applied to A, B, C, D, and E of compression function with 20 rounds for each stage and forty registers are applied to A, B, C, D, and E of compression function with two rounds for each stage. In order to improve the throughput of SHA-1 design, unfolding transformation technique has been proposed. In this paper, the unfolding technique is designed based on factor 2. Figure 4 illustrates the structure of compression function of SHA-1 algorithm using unfolding transformation technique. From this figure, there are two non-linear functions such as Func and $Func_{t+1}$ with three different inputs. Furthermore, 8 addition operations are performed in parallel form in this architecture. It consists of four shifters such as circular left shift of A by 30, circular left shift of B by 30, circular left shift of A by 5 and circular left shift of Temp by 5. This process reduces the number of processing cycle of SHA-1 design from 80 cycles to 40 cycles where two hash operations are executed per cycle. Small number of latency can increase the throughput of SHA-1 design. Hence, efficient and high-performance design can be obtained. The outputs of SHA-1 unfolding algorithms are shown in the following equation. $ROTL^{a}(b)$ represents circular left shift or left rotation operation of b by a position to the left, and $func_t(p,q,r)$ means non-linear function at time t for three different input p, q and r.

$$\begin{aligned}
Temp &= ROTL^{5}(A_t) + func(B_t, C_t, D_t) + E_t + W_t + K_t \\
A_{t+2} &= ROTL^{5}\{Temp\} + \\
&\quad func_{t+1}\left(A_t, ROTL^{30}(B), C_t\right) + D_t + W_{t+1} + K_{t+1} \\
B_{t+2} &= Temp \\
C_{t+2} &= ROTL^{30}(A_t) \\
D_{t+2} &= ROTL^{30}(B_t) \\
E_{t+2} &= C_t
\end{aligned} \tag{3}$$

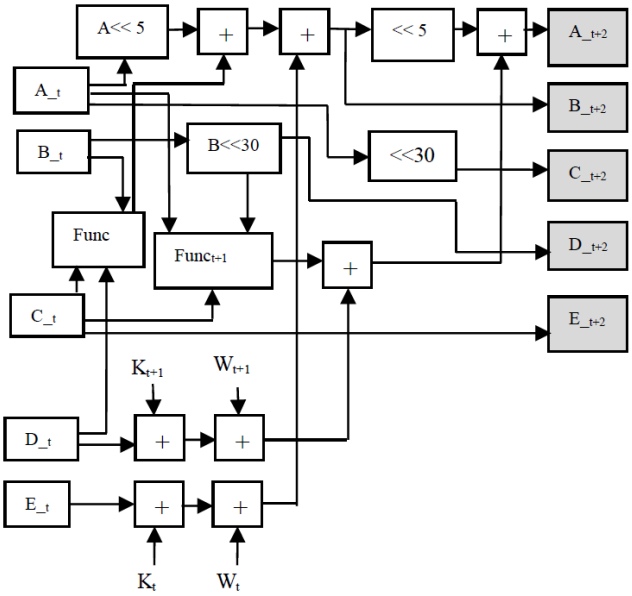

Figure 4: Unfolding Transformation of SHA-1 Algorithm Compression Function

IV. RESULTS AND DISCUSSION

The five SHA-1 designs (1) iterative, (2) inner pipelining, (3) 4 stages pipelining, (4) 40 stages pipelining and (5) 40 stages pipelining with unfolding factor 2 transformation were successfully designed using Verilog code. These designs were synthesized and implemented using both Altera Quartus II 15.0 and Xilinx ISE design suite 14.7. The proposed SHA-1 designs were simulated using ModelSim for both functional and timing simulation to verify the correctness of the design. TimeQuest timing analyzer was used to optimize the design by giving appropriate clock constraint of sdc (synopsys design constraint) file in order to meet the timing requirement. Table 4 and Table 5 show the five proposed SHA-1 design results obtained on Arria II GX and Virtex 5 XC5VLX50T after synthesis and implementation process. The following formula can be used to calculate the throughput of SHA-1 design. p denotes the number of pipeline stages in the SHA-1 design.

$$Throughput = \frac{512 \times Maximum\ Frequency \times p}{Latency} \tag{4}$$

The novelty of this design is using combination method between unfolding and the number of pipelining stages as shown in Table 4 and 5. Table 4 shows the result for

unfolding SHA-1 40 stages pipelining give high throughput which is about 150.27 Gbps on Arria II GX. The maximum frequency of this design improves significantly on Xilinx Virtex 5 XC5VLX50T with 458.60 MHz as shown in Table 5. Since the latency for this unfolding SHA-1 design reduces from 80 cycles to 40 cycles, the throughput of this design increases up to 223.62 Gbps. Besides, the total register of the SHA-1 unfolding transformation with 40 stages pipelining reduces for both Altera and Xilinx implementation.

Table 4
Synthesis and Implementation based on Altera Arria II GX

Design	Pipeline	Maximum Frequency (MHz)	ALUTs	Total Registers	Latency	Throughput (Mbps)
SHA1_ite	1	279.64	402	720	82	1746.04
SHA1_inner	1	281.53	411	681	82	1757.85
SHA1_4	4	316.76	976	715	82	7911.27
SHA1_40	40	321.54	6715	3229	82	80306.57
SHA1_Unfold40	40	308.17	9799	718	42	150269.56

Table 5
Synthesis and Implementation based on Xilinx Virtex5 XC5VLX50T

Design	Pipeline	Maximum Frequency (MHz)	Slice Register	Slice LUT	Latency	Throughput (Mbps)
SHA1_ite	1	233.899	720	897	82	1460.44
SHA1_inner	1	232.918	681	950	82	1454.32
SHA1_4	4	430.024	974	1332	82	10740.11
SHA1_40	40	427.546	5953	6465	82	106782.22
SHA1_Unfold40	40	458.593	3448	11994	42	223618.68

V. CONCLUSION

In conclusion, the five SHA-1 designs such as SHA-1 iterative, SHA-1 inner pipelining, SHA-1 4 stages pipelining, SHA-1 40 stages pipelining and SHA-1 40 stages pipelining with unfolding factor 2 transformation were designed using both Altera Quartus II and Xilinx ISE design suite 14.7 and evaluated. Simulation results showed that SHA-1 40 stages pipelining with unfolding factor 2 transformations provided the higher speed and throughput of SHA-1 design. By using the techniques of pipelining and unfolding, the throughput increased up to 159.3 Gbps on Arria II GX and 223.6 Gbps on Virtex 5 XC5VLX50T. The maximum frequency of unfolding SHA-1 40 stages pipelining design was 308.17 MHz and 458.6 MHz on Arria II GX and Virtex 5 XC5VLX50T respectively. This design only used 9799 combinational ALUTs and 718 total registers on Arria II GX. For Virtex 5 XC5VLX50T, the usage of slice LUT and slice register was about 11994 and 3448 respectively. Hence, by using appropriate FPGA device, the high frequency and small area implementation can be obtained. This proposed SHA-1 design provided better performance compared with traditional designs. As a result, our proposed technique can improve the performance of SHA-1 design which is useful for security system design and other hash functions.

ACKNOWLEDGEMENTS

This project was supported by Universiti Malaysia Sarawak (UNIMAS).

REFERENCES

[1] Federal Information Processing Standards Publication, Secure Hash Standard (SHS), FIPS PUB 180-4, Information Technology Laboratory National Institute of Standards and Technology Gaithersburg, Mar. 2012.

[2] Q. Dang, Draft NIST Special Publication 800-107(Revised), Recommendation for Applications Using Approved Hash Algorithms, Computer Security Division, Information Technology Laboratory, Computer Security, Sep. 2011.

[3] J. Kim, H. Lee, Y. Won, "Design for High Throughput SHA-1 Hash Function on FPGA," *Fourth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 403-404, 2012.

[4] L. Miao, X. Jinfu, Y. Xiaohui, Y. Zhifeng, "Design and Implementation of Reconfigurable Security Hash Algorithms based on FPGA, Information Engineering," *ICIE'09 WASE International Conference, Taiyuan, Chanxi*, pp .381-384, 2009.

[5] L. Jiang, Y. Wang, Q. Zhao, Y. Shao, X. Zhao., "Ultra High Throughput Architectures for SHA-1 Hash Algorithm on FPGA," *Computational Intelligence and Software Engineering, CiSE 2009, International Conference, Wuhan*, pp. 1-4, 2009.

[6] P. Kitsos, N. Sklavos, and O. Kaufopavlou, "An efficient implementation of the digital signature algorithm," *Proceedings of 9th IEEE International Conference on Electronics, Circuits and Systems*, vol. 3, pp. 1151-1154, 2002.

[7] G. Selimis, N. Sklavos, and O. Koufopavlou, "VLSI Implementation of the Keyed-Hash Message Authentication Code for the Wireless Application Protocol," *Proceedings of the 10th IEEE International Conference on Electronics, Circuits and Systems*, vol. 1, pp 24-27, 2003.

[8] N. Sklavos, E. Alexopoulos and O. Koufopavlou, "Networking Data Integrity, High Speed Architectures and Hardware Implementations," *The International Arab Journal of Information Technology*, vol. 1, no. 0, pp. 54 – 59, Jul. 2003a.

[9] N. Sklavos, G. Dimitroulakos, and O. Koufopavlou, "An Ultra High Speed Architecture for VLSI Implementation of Hash Functions," *Proceedings of the 10th IEEE International Conference on Electronics, Circuits and Systems ICECS*, vol. 3, pp. 990-993, 2003b.

[10] E. H. Lee, J. H.Lee, I. H.Park, K. R.Cho, "Implementation of high-speed SHA-1 architecture," *IEICE Electronics Express*, vol. 6, no. 16, pp. 1174-1179, 2009.

[11] I. I. Yiakoumis, M. E. Papadonikolakis, H. E. Michail, A. P. Kakarountas, C. E. Goutis, "Maximizing the hash function of authentication codes," *IEEE Potentials*, vol. 25, no. 2, pp. 9 -12, Mar. 2006.

[12] H. Michail, C. Goutis, "Holistic Methodology for designing Ultra High-Speed SHA-1 hashing Cryptographic Module in hardware," *IEEE International Conference on Electron Devices and Solid-State Circuits, 2008, EDSSC*, 2008.

[13] J. H. Lee, S. C. Kim, Y. J. Song, "High-Speed FPGA Implementation of the SHA-1 Hash Function," *IEICE Trans. Fundamentals*, vol. E94-A, no. 9, pp. 1873 – 1876, Sep. 2011.

[14] S. Suhaili, T. Watanabe, "High Throughput Evaluation of SHA-1 Implementation using Unfolding Transformation", *ARPN Journal of Engineering and Applied Sciences*, vol. 11, no. 5, pp. 3350 – 3355, Mar. 2016.

[15] F. R. Henriquez, N. A. Saqib, A. D. Perez, C. K. Koc, "Cryptographic Algorithms on Reconfigurable Hardware," *Springer series on Signal and Communication*, pp. 58-59, 2006.

[16] K. K. Parhi. "VLSI Digital Signal Processing Systems: Design and Implementation," *John Wiley & Sons, Inc*. pp. 119-140, 1999.