Hardware Complexity of SHA-1 and SHA-256 Based on Area and Time Analysis

Jun-Cheol Jeon
Dept. of Information Security
Woosuk University
Jeonbuk, Korea
jcjeon@ws.ac.kr

Kang-Joong Seo, Kee-Won Kim
Dept. of Information Security
Woosuk University
Jeonbuk, Korea
seokjdb@paran.com, nirk@paran.com

Abstract— ***This paper presents the analysis of a gate-level hardware complexity of SHA-1 and SHA-256. There are several kinds of SHA series' analysis on a hardware point of view but their analyses can be relatively measured according to the given equipments and facilities. In this paper, we provide a logical approach on hardware complexity analysis in area and time angle defined by the number of transistors needed for its construction and the time needed for the signal change to propagate through gates.***

Keywords- Hardware complexity; Area and time complexity; SHA-1; SHA-256

I. INTRODUCTION

Any message authentication or digital signature mechanism can be viewed as having fundamentally two levels [1]. At the lower level, there must be some sort of function that produces an authenticator: a value to be used to authenticate a message. This lower-level function is then used as primitive in a higher-level authentication protocol that enables a receiver to verify the authenticity of a message. Data integrity assurance and data origin authentication are essential security services in financial transactions, electronic commerce, electronic mail, software distribution, data storage and so on. Cryptographic hash functions are utilized to achieve these security services [2].

The following cryptographic hash functions, all based on the so called MD4 initially proposed in [3] have received the greatest attention: MD5 [4], SHA-1 [5] and SHA-256 [6], RIPEMD-160 [7], and HAVAL [8]. Most widely used hash functions in real applications are message-digest algorithm MD5 and secure hash algorithm SHA-1. For better security, three new hash functions, SHA-256, SHA-384 and SHA-512, referred to as SHA-2, with the security matching the security of AES with complexity of the best attack as 2^{128}, 2^{192} and 2^{256}, respectively, have been announced by the National Institute of Standards and Technology (NIST) [6]. The functional characteristics of SHAs are presented in Table 1.

By the way, in applications where speed is important and very large amounts of data have to be authenticated (e.g., electronic financial transactions, software integrity), hardware implementations are the natural solution. Thus A variety of research results have been presented based on various different platforms [2, 9-12].

In order to decide which scheme is better than another scheme based on hardware complexity point of view, more absolute and concrete analysis than typical hardware implementation methods is required. Thus we provide a hardware complexity analysis of SHA-1 and SHA-256 based on area and time complexity defined by the number of transistors and the time needed through gates. In this paper, we only analyze two well-known schemes because the complexities for the rest schemes can be easily analyzed by the same way as we provide.

The rest of this paper is organized as follows. Section 2 describes the area and time complexity of each gate and the complexity of a 32-bit improved super block carry-lookahead adder (ISBCLA), which is common adder in SHA series. The hardware complexity analysis of SHA-1 and SHA-256 is

TABLE I. COMPARISON OF FUNCTIONAL CHARACTERISTICS OF HASH FUNCTIONS

Hash functions	SHA-1	SHA-224	SHA-256	SHA-384	SHA-512
Size of hash value	160	224	256	384	512
Complexity of the best attack	2^{80}	2^{112}	2^{128}	2^{192}	2^{256}
Message size	$<2^{64}$	$<2^{64}$	$<2^{64}$	$<2^{128}$	$<2^{128}$
Message block size	512	512	512	1024	1024
Word size	32	32	32	64	64
Number of words	5	8	8	8	8
Number of digest rounds	80	64	64	80	80
Number of constants	4	64	64	80	80
Round-dependent operations	f_t	None	None	None	None

presented in Section 3. Finally, Section 4 presents our conclusion.

II. Area and Time Complexity

We are usually trying to find the design that will best satisfy a given set of design requirements when we implement arithmetic unit design. We consider construction simplicity, defined by the number of transistors needed for its construction and the time needed for the signal change to propagate through gates [13]. Let T_{XGATE} and A_{XGATE} be the gate delay and the number of transistors of GATE with X input. The following Table 2 shows the number of transistors and propagation time according to the gates.

We analyze a structure of addition modulo 2^{32} which is a common structure for SHA-1 and SHA-256. The following shows a configuration of a 32-bit improved super block carry-lookahead adder (ISBCLA) which has a good performance of variable adders [14]. We eliminate all input and output values since we only need a gate level complexity as shown in Fig.1.

TABLE II. THE AREA AND TIME VALUE OF BASIC GATES

The number of transistors (*tr*)		The time of gate delay (*ns*)	
A_{2AND} / A_{2OR}	6	T_{2AND} / T_{2OR}	2.4
A_{3AND} / A_{3OR}	8	T_{3AND} / T_{3OR}	2.8
A_{4AND} / A_{4OR}	10	T_{4AND} / T_{4OR}	3.2
A_{5AND} / A_{5OR}	12	T_{5AND} / T_{5OR}	3.6
A_{2XOR}	14	T_{2XOR}	4.2
A_{NOT}	2	T_{NOT}	1
A_{FF}	18	T_{FF}	3.8

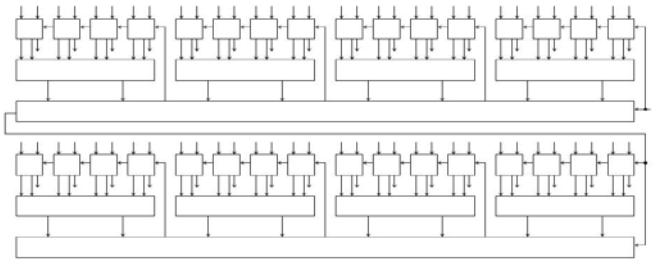

Fig. 1 Configuration of a 32-bit improved super block carry-lookahead adder (ISBCLA)

The adder is composed of three levels and each level consists of different logical circuits. ISBCLA consists of 32 blocks at the first level, 8 blocks at the second level, and 2 blocks at the third level. The following circuits show the detail configurations of each three-level.

One of 32 blocks at the first level shown in Fig.2 is composed of four 2-input AND, two 2-input XOR and two 2-input OR gates, and its critical path is one 2-input XOR, one 2-input AND and one 2-input OR gate. At the second level, one of 8 blocks shown in Fig.3 is composed of one 2-input AND, one 3-input AND, two 4-input AND and one 4-input OR gates, and its critical path is one 4-input AND and one 4-input OR gate. Meanwhile, the circuit configuration at the third level is a little more complicate. One of two blocks shown in Fig.4 is composed of one 5-input AND, two 4-input AND, three 3-input AND, four 2-input AND, one 5-input OR, one 4-input OR, one 3-input OR, and one 2-input OR gates. However it has only one 5-input AND and one 5-input OR gate as its critical path.

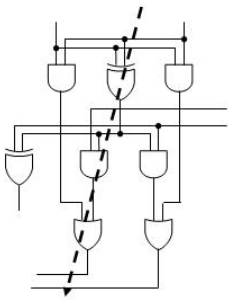

Fig. 2 Configuration of first-level (block) for ISBCLA

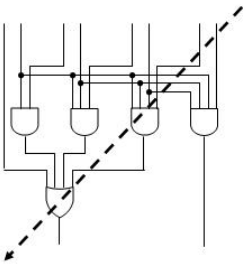

Fig. 3 Configuration of second-level (block) for ISBCLA

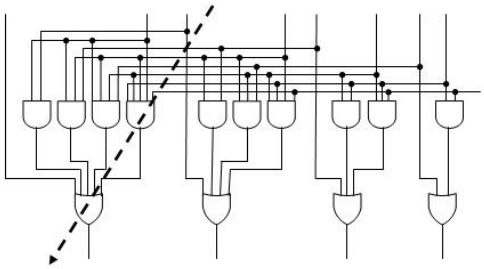

Fig. 4 Configuration of third-level (block) for ISBCLA

One block of each level of the area and time complexity for ISBCLA is noted in Table 3. Thus the addition modulo 2^{32} has an area complexity of 2,584 *tr* ((64×32)+(38×8)+(116×2)) and time complexity of 22.6 *ns* (9+6.4+7.2).

III. Analysis of SHA-1 and SHA-256

In this section, we analyze the complexity of SHA-1 and SHA-256 using the 32-bit adder, ISBCLA as shown in the above section. In order to compare two schemes based on an equivalent condition, we consider only a single 512-bit block computation.

TABLE III. THE AREA AND TIME VALUE OF BASIC GATES

	Area complexity (*tr*)		Time complexity (*ns*)	
1-level	$4 \times A_{2AND}$	24	$1 \times T_{2AND}$	2.4
	$2 \times A_{2XOR}$	28	$1 \times T_{2XOR}$	4.2
	$2 \times A_{2OR}$	12	$1 \times T_{2OR}$	2.4
sub-sum		*64*		*9*
2-level	$2 \times A_{4AND}$	20	$1 \times T_{4AND}$	3.2
	$1 \times A_{3AND}$	8	$1 \times T_{4OR}$	3.2
	$1 \times A_{2OR}$	10		
sub-sum		*38*		*6.4*
3-level	$1 \times A_{5AND}$	12	$1 \times T_{5AND}$	3.6
	$2 \times A_{4AND}$	20	$1 \times T_{5OR}$	3.6
	$3 \times A_{3AND}$	24		
	$4 \times A_{2AND}$	24		
	$1 \times A_{5OR}$	12		
	$1 \times A_{4OR}$	10		
	$1 \times A_{3OR}$	8		
	$1 \times A_{2OR}$	6		
sub-sum		*116*		*7.2*

A. Analysis of area and time complexity of SHA-1

SHA-1 needs a 5-word (160-bit) register for *IV*, intermediate and final message digests, a 16-word (512-bit) register for an input message, a 4-word (128-bit) register for K_t, and 80-word (2560-bit) for W_t. Fig. 5 illustrates how the 32-bit word values W_t are derived from the 512-bit message. Thus, at the first 16 steps of processing, the value of W_t is equal to the corresponding word in the message block. For the remaining 64 steps, the value of W_t consists of the circular left shift by one bit of the XOR of four of the preceding values of W_t. It needs 3 × 32 XOR gates to produce the rest of W_t ($16 \le t \le 79$). We do not consider their time complexity since they can be pre-computed and preloaded.

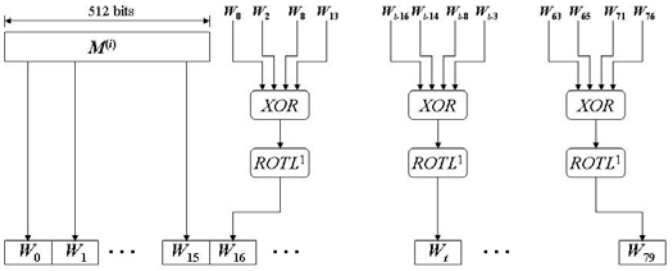

Fig. 5 Creation of 80-word input sequence for SHA-1 processing of single block

In SHA-1, three different functions are used, i.e., $Ch(x, y, z)=(x \wedge y) \oplus (\neg x \wedge z)$, $Parity(x, y, z)= (x \oplus y \oplus z)$, and $Maj(x, y, z)=(x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$. The functions have the following complexities.

Fig. 7 shows SHA-1 operation in a single step. A single step consists of a function, f_t, which is one of three functions in Table 4 and four additions. We do not consider shift operations since it is easily implemented by wiring with a negligible cost in hardware implementation.

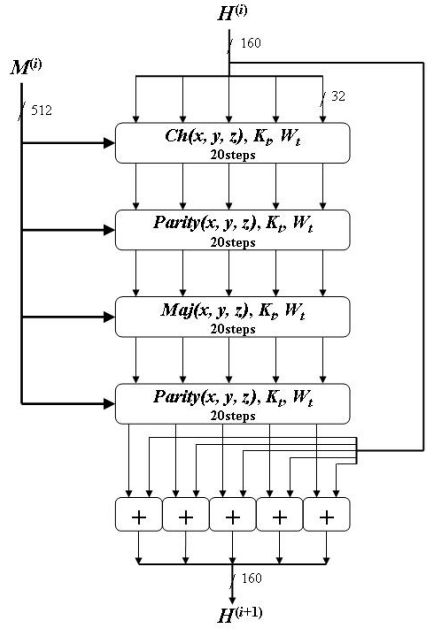

Fig. 6 SHA-1 processing of a single 512-bit block (SHA-1 compression function)

TABLE IV. THE AREA AND TIME VALUE OF BASIC GATES

	Area complexity (*unit*)			Time complexity (*ns*)		
$Ch(x, y, z)$	$2 \times A_{2AND}$	12		$1 \times T_{2AND}$	2.4	
	$1 \times A_{2XOR}$	14		$1 \times T_{2XOR}$	4.2	
	$1 \times A_{NOT}$	2		$1 \times T_{NOT}$	1	
Sub-sum			*28*			*7.6*
$Parity(x, y, z)$	$2 \times A_{2XOR}$	28		$2 \times T_{2XOR}$	8.4	
Sub-sum			*28*			*8.4*
$Maj(x, y, z)$	$3 \times A_{2AND}$	18		$1 \times T_{2AND}$	2.4	
	$2 \times A_{2XOR}$	28		$2 \times T_{2XOR}$	8.4	
Sub-sum			*46*			*10.8*

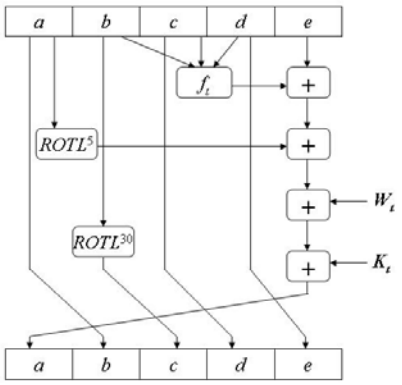

Fig. 7 SHA-1 operation in a single step

Thus the area complexity for SHA-1 is calculated as follows: W_t generation - 1,344 (3×32×14), three different functions - 3,264 ((28×32)+(46×32)+(28×32)), four serial

additions - 2,584 (demands only one adder), 4-word register for K_t - 2,304 (128×18), 80-word register for W_t - 46,080 (2,560×18), 16-word register for input message - 9,216 (512×18), 5- word register for *IV*, intermediate and final digest - 2,880 (160 × 18), and five parallel additions for the end of the round - 12,920 (5×2,584). Therefore, the total required area complexity for SHA-1 is 80,592.

Meanwhile, the critical path in a single step is to read register, f_t function, four additions, and to write register. Thus, the time complexity for SHA-1 is calculated as follows: four functions for 80 steps - 704 (20×(7.6+8.4+10.8+8.4)), four additions for 80 steps - 7,232 (80×4×22.6), read and write register for 80 steps - 608 (80×2×3.8), and five parallel additions for the end of the round - 2,584. Therefore, the total required time complexity for a single 512-bit block is 11,128.

B. Analysis of area and time complexity of SHA-256

SHA-256 needs a 8-word register (256-bit) for *IV*, intermediate and final message digests, a 16-word (512-bit) register for an input message, a 64-word (2048-bit) register for K_t, and 64-word (2048-bit) for W_t.

Fig. 8 illustrates how the 32-bit word values W_t are derived from the 512-bit message. Thus, at the first 16 steps of processing, the value of W_t is equal to the corresponding word in the message block. For the remaining 48 steps, the value of W_t consists of the addition of four of the preceding values of W_t. It demands two functions and each function needs two XOR operations as follows. $\sigma_0^{256}(x) = ROTR^7(x) \oplus ROTR^{18}(x) \oplus SHR^3(x)$, $\sigma_1^{256}(x) = ROTR^{17}(x) \oplus ROTR^{19}(x) \oplus SHR^{10}(x)$. Thus, it needs three adder and 4 × 32 XOR gates to produce a W_t so that the area complexity to produce the whole W_t ($16 \le t \le 63$) are 3 × 2,584 + 4 × 32 × 14 = 9,544. We also do not consider their time complexity because of the reason mentioned in the above Section 3.1. Fig. 9 shows SHA-256 processing.

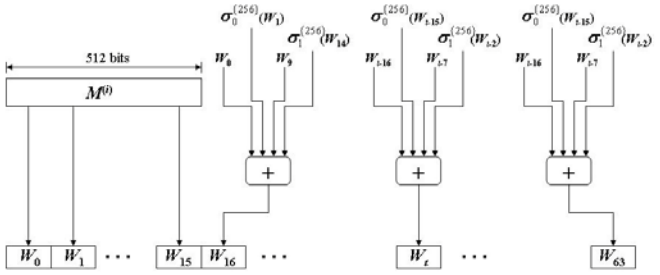

Fig. 8 Creation of 64-word input sequence for SHA-256 processing of single block

In SHA-256, two different functions are used, i.e., $\sum_0^{\{256\}}(x) = ROTR^2(x) \oplus ROTR^{13}(x) \oplus ROTR^{22}(x)$ and $\sum_1^{\{256\}}(x) = ROTR^6(x) \oplus ROTR^{11}(x) \oplus ROTR^{25}(x)$. Fig. 10 shows SHA-256 operation in a single step. A single step consists of four functions, and seven additions. Thus the area complexity for SHA-256 is calculated as follows: W_t generation - 9,544 ((3×2,584)+(4×32×14)), four functions - 4,160 (896+1,472+2×(2×32×14)), seven additions (two parallel additions) - 5,168 (2×2,584), Two 64-word registers for K_t and W_t - 73,728 (2×64×32×18), 16-word register for input message - 9,216 (16×32×18), 8-word register for *IV*, intermediate and final digests - 4,608 (8×32×18), and eight additions for the end of the round - 20,672 (8×2,584). Therefore, the total required area complexity for SHA-256 is 127,096.

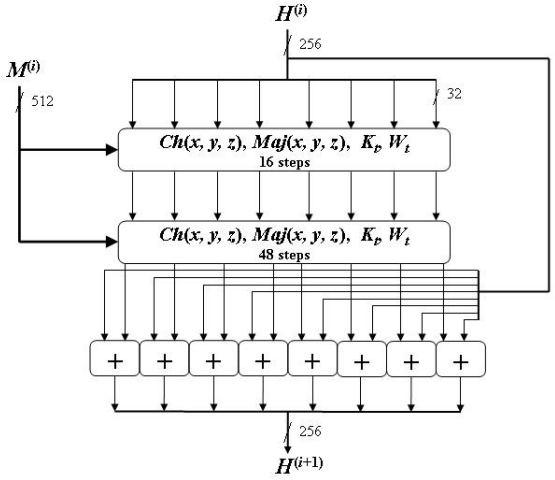

Fig. 9 SHA-256 processing of a single 512-bit block (SHA-256 compression function)

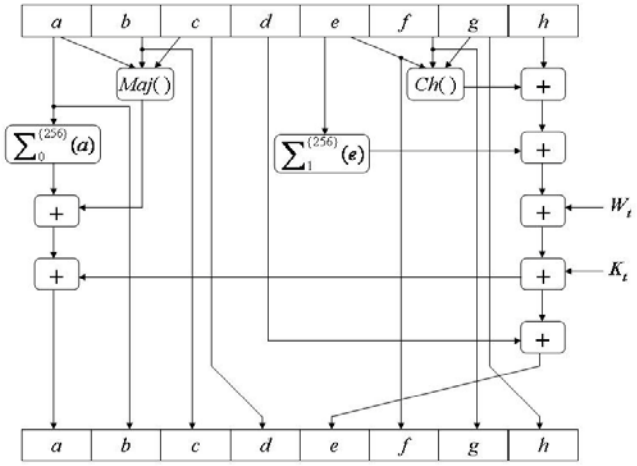

Fig. 10 SHA-256 operation in a single step.

Meanwhile, the critical path in a single step is to read register, *Ch*() function, five additions, and to write register. Thus, the time complexity for SHA-256 is calculated as follows: *Ch*() function for 64 steps (three parallel functions) – 486.4 (64×7.6), five additions for 64 steps (two parallel additions) - 7,232 (64×5×22.6), read and write register for 64 steps - 486.4 (64×2×3.8), and seven parallel additions for the end of the round - 2,584. Therefore, the total required time complexity for a single 512-bit block is 10,788.8 ≈ 10,789.

TABLE V. THE AREA AND TIME VALUE OF BASIC GATES

	SHA-1	SHA-256
Area complexity	80,592	127,096
Time complexity	11,128	10,789
AT^2 value	$9{,}980\times10^9$	$14{,}794\times10^9$

In terms of area and time complexity in gate level, the best method, as noted in [15], is to evaluate the AT^2 value for each scheme. Area is assumed to be totally contributed by the number of transistors in gates and registers required to compute a find hash result. The cost due to time consists of the delay time of the gates and registers for proceeding a 512-bit input message block. As shown in Table 5, SHA-256 has nearly 50% more AT^2 value than SHA-1.

IV. CONCLUSION

In this paper, we have analyzed well-known hash functions, SHA-1 and SHA-256. In order to analyze a hardware complexity, we have adopted an area and time complexity measure. Typical analysis by implementation can be differently measured according to given equipments and facilities, but our approach does not be influenced on any circumstances. However, this analysis provides a little flexibility that a circuit designer can coordinate to minimize either area or time complexity though it does not give a big difference in the construction of a scheme. Therefore, we believe that our approach can be efficiently used for a logical analysis of hardware complexities.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2011-0014977).

REFERENCES

[1] Stallings, W.: 'Cryptography and Network Security: Principles and Practice' (Prentice Hall Inc., 2nd edn. 1999)

[2] Ahmad, I. and Das, A. S.: 'Hardware implementation analysis of SHA-256 and SHA-512 algorithms on FPGAs', Computers and Electrical Engineering, 2005, 31, pp. 345-360

[3] Rivest, R. L.: 'The MD4 message-digest algorithm'. Proc. Crypto'90, LNCS 537, 1991, pp. 303-311

[4] Rivest, R. L.: 'IETF RFC 1321: The MD5 Message-Digest algorithm', Available from http://www.ietf.org/rfc/rfc1321.txt

[5] Federal Information Processing Standard (FIPS): 'Publication 180-1, Secure Hash Standard (SHS)', U.S. Doc/NIST, April 1995

[6] Federal Information Processing Standard (FIPS): 'Publication 180-2, Secure Hash Standard (SHS)', U.S. Doc/NIST, May 2001.

[7] Integrity Primitives for Secure Information Systems: Final Rep. of RACE Integrity Primitives Eval. RIPE-RACE 1040, LNCS 1007, 1995

[8] Zheng, Y., Pieprzyk, J. and Sebery, J.: 'HAVAL - A One-Way Hashing Algorithm with Variable Length of Output'. Proc. Auscrypt'92, LNCS 718, 1993, pp. 83-104

[9] Dominikus, S.: 'A hardware implementation of MD5-family hash algorithm'. Proc. Int. Conf. on electronics circuits and systems. Dubrovnik, Croatia, 2002, pp. 1143-1146

[10] Satoh, A. and Inoue, T.: 'ASIC-hardware-focused comparison for hash functions MD5, RIPEMD-160, and SHS', INTEGRATION, the VLSI journal, 40, 2007, pp. 3-10

[11] Sklavos, N. and Koufopavlou, O.: 'On the hardware implementation of the SHA-2 (256, 384, 512) Hash functions'. Proc. Int. Conf. Circuits and Systems, 5, 2003, pp. 153-156

[12] Ting, K. K., Yuen, S. C. L., Lee, K. H. and Leong, P. H. W.: 'An FPGA-based SHA-256 processor'. Proc. Int. Work. Field Prog. Logic and Appl, LNCS 2438, 2002, pp. 576-585

[13] Gajski, D. D.: 'Principles of Digital Design' (Prentice-Hall International Inc., 1997)

[14] Omondi, A. R.: 'Computer Arithmetic Systems Algorithms, Architecture and Implementations' (Prentice Hall International Limited, 1994)

[15] Das, A. K. and Chaudhuri, P. P.: 'An Efficient On-Chip Deterministic Test Pattern Generation Scheme', Euromicro J., Microprocessing & Microprogramming, 1989, 26, pp. 195-204