High-performance Implementation of a New Hash Function on FPGA

Demin Han, Renfa Li, Juanli Zeng

Embedded Systems & Networking Laboratory, Hunan University, Changsha, China

handemin1987@gmail.com

Abstract— **Skein has the advantages of higher security (resisting against traditional attacks), faster speed and selectable parameters. Therefore, it becomes a strong competitor for next generation secure hash algorithm standard (SHA-3) which will be used widely in communication and security for substitution of SHA-2. The problems of existing works lie in implementation for only one structure and lack detailed comparison of different structures. Based on analysis of the algorithm, we accomplished three structures (iterative, 4-unrolled and 8-unrolled) of Skein and ported the designs to FPGA respectively. Finally detailed analysis and comparison with our different structures and other implementations are provided from aspects of hardware resource and performance. The results show that our implementation has better performance and takes up less hardware resources than existing works under the same structure. Our implementation can meet the requirement of real-time and high performance field.**

***Keywords*— Hash Function, SHA-3, Skein, Threefish, FPGA**

I. INTRODUCTION

Security in communication has become more and more important with the development of computer network and communication technology. Encryption and authentication are popularly used to guarantee the security in communication by avoiding the information leak and verifying the data integrity separately. Authentication is a useful way to provide trust of each other and accuracy of information, which is composed of data integrity verifying, digital signature and cryptographic protection. Hash algorithm is an efficient authentication method with fast speed, high security, and it is successfully applied to software and hardware platform. Software implementation could not meet the requirement of high-speed data processing application generally, such as high-speed router and switch supporting IPSec. In addition, we often need coprocessor of hash algorithm which is implemented on FPGA or ASIC hardware platform to help system meet real-time in some embedded security systems. Hash algorithm is an indispensable component in modern information security field.

Secure Hash Algorithm (SHA) is published by National Institute of Standards and Technology (NIST) firstly in 1993 [1]. From then on, SHA-0, SHA-1 and SHA-2 was proposed and being widely used in many application such as electronic commerce and Internet banking. The key idea of hash algorithm is mapping a long message into a fixed length message digest. Hash algorithm plays an important role in cryptographic field. SHA is wildly used in wireless network communication, digital signature and so on.

Unfortunately, traditional hash function such as MD5 and SHA-1 were successfully attacked so far and SHA-2 is facing threat now [2]-[4]. Therefore NIST began to select the next generation secure hash algorithm SHA-3 in Nov 2007 [5]. Skein has strong competitiveness and becomes one of the final five candidate algorithms after the second round contest. Skein is proposed by cryptographic experts come from Microsoft, University of California, and University of Washington. Skein has the novel structure (UBI and Threefish), which has the characters of high security, fast speed and flexible optional parameters [6].

Skein has been implemented on FPGA and ASIC by some researchers[7]-[12].Baldwin[7] and Homsorikamol[8] realized the 4-unrolled structure of Skein-512 and they compared the designs with other candidate algorithms on throughput and area, but they did not analyze the different structures of Skein. Tillich [9] realized the 8-unrolled mode of Skein-256, Skein-512 and Skein-1024 on FPGA and ASIC, but there is also no comparison in different structure. From the above we can see that the existing implementation of Skein focused on the single version and structure, lacking detailed analysis and comparison among different structures of Skein-256 and Skein-512.

In this paper, we realized iterative, the 4-unrolled and the 8-unrolled structure of Skein-256 and Skein-512 separately. We give out detailed analysis and comparison of our different structures and other implementations. The experimental results show that our implementation has better performance and takes up less hardware resources than existing works under the same structure.

The rest paper is organized as follows. Section II introduces the Skein algorithm. Section III describes our FPGA-based hardware implementations of Skein. Section IV presents the results and comparisons with different architecture and the other existing works. Finally, conclusions are drawn in Section V.

II. PRELIMARY OF SKEIN

According to internal state space, Skein has three versions including Skein-256, Skein-512 and Skein-1024. Skein-512 is a recommendation version, it has high security and can satisfy most requirement; Skein-256 has smaller state space and need lesser computing, so it can be used in embedded system which resources are limited; Skein-1024 is a complementary version of Skein-512. The differences among versions lie in rounds of compression function, mix function's number and mix function's arguments. The following is an introduction of Skein based on Skein-512.

Skein consists of three parts which are Threefish, Unique Block Iteration (UBI) and optional argument system. Among these parts, Threefish is compression function which determines state space; UBI is used to map arbitrary input to fixed output size. Optional argument system can provides optional features without imposing any overhead. The data process flow of Skein is shown in Figure 1, it consists of three UBI modules, and every UBI is composed of Threefish invocation chain that processes data block. Configuration UBI can be replaced by precomputed initialization vector (IV).

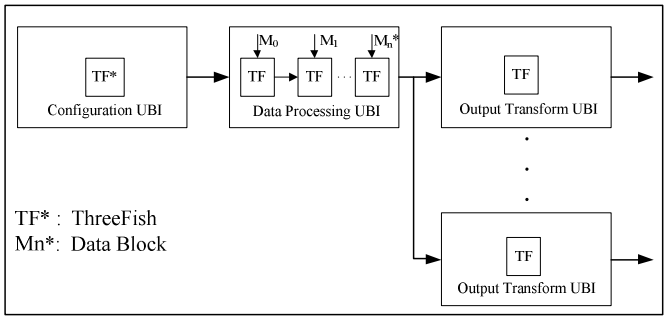

Figure 1. Data process flow of Skein

Threefish is the most important part of Skein, it is a tweakable block cipher based on 64-bit words. MIX is a non-linear mixing function which only include three operations: modulo addition (+), exclusive-or (^) and constant rotations (<<<) . MIX has two inputs (x_0, x_1) and generates two outputs (y_0, y_1), it can be expressed as follows formally:

$$y_0 = (x_0 + x_1) \bmod 2^{64} \quad (1)$$

$$y_1 = (x_1 <<< R_{(d \bmod 8),\, j)}) \wedge y_0 \quad (2)$$

In the formula (2), $R_{(d \bmod 8),\, j)}$ is a constant related to current round. Every four rounds need a subkey injection. Subkey is generated using Key, Tweak and Counter by key schedule which makes the data block mixed completely. Key is state space of key schedule. Tweak is combination presentation of current parameter which size is 128-bit. Counter's range is related to version of Skein. Subkey can be described by following formulas:

$$k_{Nw} = C_{240} \wedge k_0 \wedge \ldots \wedge k_{Nw-1} \quad (3)$$

$$t_2 = t_0 \wedge t_1 \quad (4)$$

$$k_{s,i} = k_{(s+i) \bmod (Nw+1)} \qquad i = 0, \ldots, N_w - 4 \quad (5)$$

$$k_{s,i} = k_{(s+i) \bmod (Nw+1)} + t_{s \bmod 3} \qquad i = N_w - 3 \quad (6)$$

$$k_{s,i} = k_{(s+i) \bmod (Nw+1)} + t_{(s+1) \bmod 3} \qquad i = N_w - 2 \quad (7)$$

$$k_{s,i} = k_{(s+i) \bmod (Nw+1)} + s \qquad i = N_w - 1 \quad (8)$$

In above formulas, $k_0 \ldots k_{Nw}$ represent the Key state space. C_{240} is a constant which makes k_{Nw} not to be zero. The t_0 and t_1 are the low and high 64-bit of Tweak respectively. The s variable is counter which represents current round number.

III. OUR IMPLEMENTATION OF SKEIN

The key of implementation lies in Threefish which includes almost all operations of Skein. The procedure of processing data block is controlled by control unit.

A. Threefish

Threefish is the core component of Skein. The design of its data path will determine the area and performance of Skein. The Threefish of Skein-512 needs 72 rounds to generate result, and the 72 rounds has a regular pattern of 9×8. Every round needs four different MIX. Threefish needs 32 different MIX, so there are 8 groups of MIX functions. How to connect the 8 groups MIX functions decides the architecture of Threefish and number of clock cycle. Generally there are three architectures of Threefish: iterative, 4-unrolled and 8-unrolled according to the connection of different MIX. The Threefish architecture of 8-unrolled Skein-512 is shown as Figure 2. There are 32 MIX functions running and two subkey injection in every cycle, but 4 and 16 MIX functions running and one subkey injection for iterative and 4-unrolled architecture respectively.

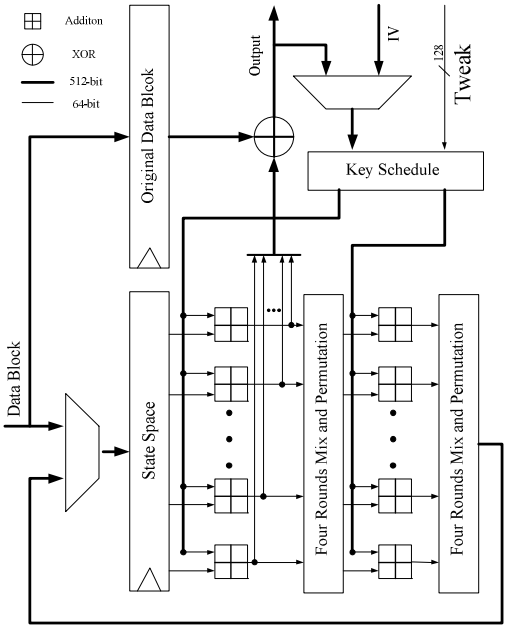

Figure 2. Threefish architecture of 8-unrolled Skein-512

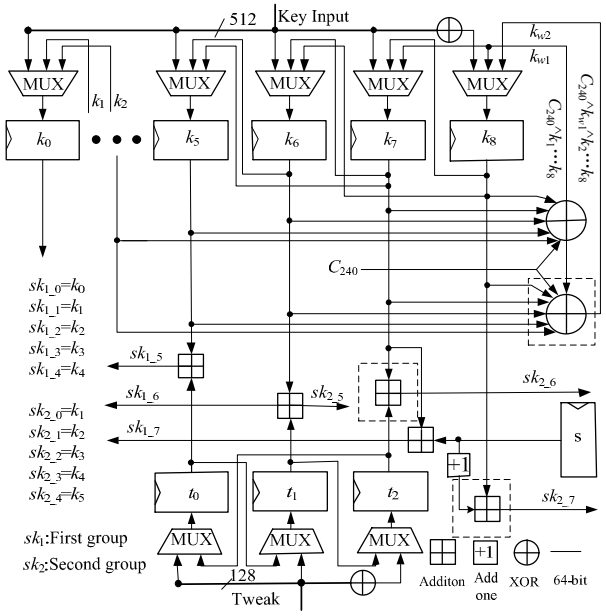

Figure 3. Key schedule architecture of 8-unrolled Skein-512

The key schedule is the most complex unit of Threefish. The iterative and 4-unrolled architecture has the same key schedule unit. The key schedule of 8-unrolled architecture needs some additional operations in order to generate two groups subkey every clock cycle. We present the key schedule architecture of 8-unrolled Skein-512 in Figure 3. The variables in Figure 3 correspond to formulas (3)-(10). We use two groups shifer registers (k_n and t_n) to achieve the functions. The state space of key schedule changed every cycle.

Note that there are three dotted boxes in Figrue 3. They are the additional operations mentioned above. We need another two formulas to describe the added important operations as follows:

$$k_{w1} = C_{240} \wedge k_0 \wedge \ldots \wedge k_{Nw} \quad (9)$$

$$k_{w2} = C_{240} \wedge k_{w1} \wedge \ldots \wedge k_{Nw} \quad (10)$$

B. Control Units

The main parts of control unit are three Finite State Machines (FSM) as shown in Figure 4.

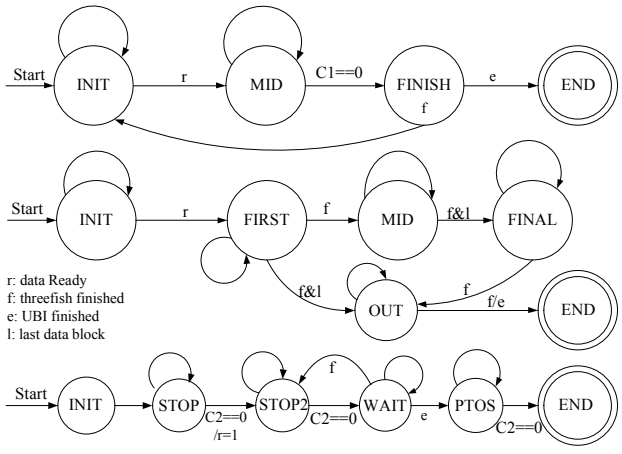

Figure 4. FSM of control unit

In Figure 4, The first is Threefish FSM which control generation of subkey and operations of MIX; the second is UBI FSM, it control generation of Tweak and transformation of three UBI; the third is interface FSM used to load data block to Skein and output final hash value. The interface of Skein can transform data between serial and parallel because the width of internal data is 512-bit but external is 64-bit. Another function of interface is to padding the data block to reach the size of state space. There are synchronous signals to coordinate three FSM to complete the whole flow of Skein.

IV. RESULTS AND COMPARISONS

In this paper, we use Verilog hardware description language to implement our three architectures and synthesize the design using Xilinx ISE 13.1. For fair comparison, we choose a similar FPGA device (xc5vlx30t-3ff323) with other existing works. We will give related analysis and comparison from several key parameter of performance with different architectures and other's works, such as frequency, throughput and area. The throughput can be defined as follows:

$$\text{Throughput}=\frac{\text{Frequency}\times\text{StateSpace}}{\text{Cycles}} \quad (11)$$

In formula (11), the unit of Frequency is MHz; the size of StateSpace is 256-bit or 512-bit; Cycles is related to architecture and version of Skein, the unit of Throughput is Mb/s.

The hardware resource and performance of the three architectures of Skein-256 and Skein-512 are shown in Figure 5. This is the comparison among our different architecture and version implementations. The horizontal axis represents different architecture of two versions, the left vertical axis represents the number of hardware resource and the right vertical axis represents frequency, the number above middle column represent throughput. From Figure 5, we can find that the same versions have almost the same Register. The difference mainly lies in number of LUT. The iterative architecture has the highest frequency, but has the lowest throughput, because the iterative architecture takes up 4 times and 8 times cycles than 4-unrolled and 8-unrolled architecture respectively.

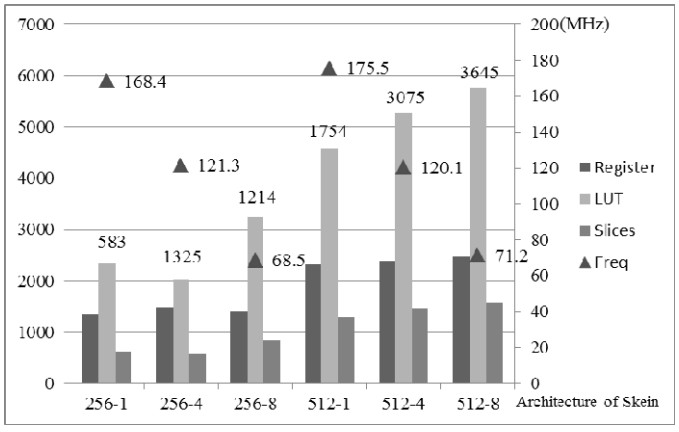

Figure 5. Resource and performance of different architecture and version

TABLE 1. FPGA HARDWARE IMPLEMENTATION COMPARISIONS OF SKEIN-512

	FPGA Platform	Threefish Architecture	Periods [cycles]	Frequency [MHz]	Throughput [Mb/s]	Area [slices]	Thp. /Area
Baldwin [7]	xc5vlx330t-2	4-unrolled	22*N*	83.65	1945	1786	1.09
Homsirikam [8]	Virtex-5	4-unrolled	19*N*	119.1	3209	1716	1.87
Tillich [9]	xc5vlx110-3	8-unrolled	10*N*	69.04	3535	1632	2.17
This work	xc5vlx30t-3	Iterative	74*N*	175.5	1214.3	1284	0.95
This work	xc5vlx30t-3	4-unrolled	20*N*	120.1	3074.6	1458	2.1
This work	xc5vlx30t-3	8-unrolled	10*N*	71.2	3645	1561	2.34

TABLE 2. FPGA HARDWARE IMPLEMENTATION COMPARISIONS OF SKEIN-256

	FPGA Platform	Threefish Architecture	Periods [cycles]	Frequency [MHz]	Throughput [Mb/s]	Area [slices]	Thp. /Area
Tillich [9]	xc5vlx110-3	8-unrolled	10*N*	68.4	1751	937	1.87
Gaj[10]	Vitex-5	8-unrolled	9N	49.8	1416	1312	1.1
Matsuo[11]	xc5vlx30-3	4-unrolled	21*N*	115	1402	854	1.64
Kobayashi[12]	xc5vlx30-3	4-unrolled	20*N*	115	1482	854	1.74
This work	xc5vlx30t-3	Iterative	74*N*	168.4	582.5	619	0.94
This work	xc5vlx30t-3	4-unrolled	20*N*	121.3	1552.6	578	2.68
This work	xc5vlx30t-3	8-unrolled	10*N*	68.5	1753.6	845	2.08

Because of additional operations of generating subkey, the 8-unrolled architecture takes up largest hardware resource, but it has the highest throughput which about two times than iterative architecture. In addition, we can find that the Slice of Skein-512 takes up about 1.5 times than Skein-256.

Table 1. and Table 2. show the detailed comparison of Skein-512 and Skein-256 with existing works respectively. Our implementation of Skein-256 8-unrolled architecture takes up about 100 slices less than [9] but has the similar frequency and throughput. Our 8-unrolled architecture of Skein-512 has the highest throughput (3645Mb/s) among the works.

The iterative architecture has poor throughput but has similar area with 4-unrolled architecture. We recommend that the high-performance hardware implementation of Skein should adopt 4-unrolled or 8-unrolled architecture which have good performance and acceptable area.

V. CONCLUSIONS

SHA-3 standard will be selected from five candidates in 2012 by NIST. In this paper we focus on a competitive candidate: Skein. We present our design of Skein's key components and implement three architectures of Skein-256 and Skein512 on FPGA respectively. And above all, we give out comprehensive comparison of different architectures and other's works. The result shows that our Skein-512 8-unrolled design can achieve a throughput of 3645Mb/s which can be used in high-performance field.

ACKNOWLEDGMENT

This work is supported by "the Fundamental Research Funds for Chinese Central Universities". Thank Embedded Systems & Networking Laboratory for providing development tools and platform.

REFERENCES

[1] Secure Hash Standard (SHS), National Institute of Standards and Technology (NIST), FIPS PUB 180-3, 1993, Available: http://csrc.nist.gov/publications/fips/fips180-3/fips180-3_final.pdf.

[2] X. Wang, H. Yu, "How to Break MD5 and Other Hash Functions," in Advances in Cryptology – EUROCRYPT 2005, LNCS 3494, p.561.

[3] X. Wang, Lisa Yin and H. Yu, "Finding Collisions in the Full SHA-1," in Proc. 25th Annual Cryptology Conference- CRYPTO 2005, LNCS 3621, p.17.

[4] W.E. Burr, "Cryptographic hash standards: where do we go from here?," Security & Privacy, IEEE , vol.4, no.2, pp.88-91, March-April 2006.

[5] NIST, Cryptographic Hash Algorithm Competition [Online], Available: http://csrc.nist.gov/groups/ST/hash/sha-3.

[6] N. Ferguson, S. Lucks, B. Scheneier, D. Whiting, M. Bellare, etc., The Skein Hash Function Family [Online]. Available: http://www.schneier.com/skein.html, 2010.

[7] B. Baldwin, N. Hanley, M. Hamilton, L. Lu, A. Byrne, M. O'Neill, and W. P. Marnane, "FPGA Implementations of the Round Two SHA-3 Candidates," in Field Programmable Logic and Applications (FPL) Conference , 2010 , pp.400-407.

[8] E. Homsirikamol, M. Rogawski, K. Gaj, Comparing Hardware Performance of Fourteen Round Two SHA-3 Candidates Using FPGAs. Cryptology ePrint Archive, Available: http://eprint.iacr.org/ , 2010/445.

[9] S. Tillich. Hardware Implementation of the SHA-3 Candidate Skein. Cryptology ePrint Archive , Available: http://eprint.iacr.org/ , 2009/159.

[10] K. Gaj, E. Homsirikamol and M. Rogawski, *Fair and Comprehensive Methodology for Comparing Hardware Performance of Fourteen Round Two SHA-3 Candidates Using FPGAs*, Lecture Notes in Computer Science, Springer , 2010, volume 6225/2010.

[11] S. Matsuo, M. Knezevic, P. Schaumont, I. Verbauwhede, A. Satoh, K. Sakiyama,and K. Ota, "How can we conduct "fair and consistent" hardware evaluation for SHA-3 candidate?," in Second SHA-3 Candidate Conference, Aug. 2010.

[12] K. Kobayashi, J. Ikegami, M. Knezevic, E.X. Guo, S. Matsuo,etc. "Prototyping platform for performance evaluation of SHA-3 candidates," in IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 2010, pp.60-63.