# A High-Performance Parallel Computation Hardware Architecture in ASIC of SHA-256 Hash

Xiaoyong Zhang*, Ruizhen WU*, Mingming Wang*, Lin Wang*

*Intel Mobile Communications Technology (Xi'an) Ltd, Xi'an Shaanxi Province China

**xiao-yong.zhang@intel.com, ruizhen.wu@intel.com, mingming.wang@intel.com, lin.b.wang@intel.com**

*Abstract*— **The SHA-256 is playing an important role in various applications, such as e-transactions and bitcoins. To achieve more profits, the SHA-256 computation capacity is a main research direction of Hashing Algorithm. In this paper, a high-performance hardware architecture of SHA-256 hash is proposed. The computation of SHA-256 is rescheduled based on hardware characterises. Three pipelines are used to replace the critical path in the round functions which can shorten the long critical path, and divide the computation chain into independent parts. Multi-computation of SHA-256 is working in parallel pipelines, indicating that the computation capacity can be 3 times of standard SHA-256 implementation. The proposed SHA-256 hardware architecture has been implemented and synthesized with Intel 14nm technology. Simulation and synthesis results show the proposed SHA-256 hashing throughput can be improved by 3 times with 50.7% power reduction, at an area cost of 2.9 times compared to the standard implementation.**

*Keywords*— **SHA-256; VLSI; High speed; Low power; Crypto currencies**

## I. INTRODUCTION

Secure hashing algorithm is used to ensure the data integrity and authenticity while being stored and transferred Hash functions take input data of arbitrary length and convert them into some fixed data, called as hash value or message digest. The SHA-256 of hashing algorithm is playing an important role in various applications. Almost all e-transactions, high-throughput designs of security schemes are needed. Bitcoin is a new popular use of SHA-256, as the POW ("Proof of Work" [1]) mentions in the Bitcoin protocol: the POW requests a huge number of SHA-256 computation to find a proper 32-bit number to satisfy the protocol requirement, the first finder is awarded by bitcoin, which means the computation capacity of SHA-256 is the main research direction.

The SHA-256 hash architecture acts more and more important nowadays thus several improved designs are proposed. To embed a security engine in an RFID tag, two compact SHA-256 designs are presented, a low area design and a low power design [2]. To achieve the improvement several adder cycles and adder selectors were added in the round computation which made it very suitable for power-area balanced applications.

One application of ideas and techniques from functional languages to the model-driven design and synthesis of hardware artifacts for SHA-256 was proposed in [3]. The co-design of hardware and software not just made the SHA-256 algorithm easier to implement, but also gave a more effective way to optimize the performance of SHA-256 from software to hardware based on designer's need.

However the most challenging request of SHA-256 is high processing speed and low power in hardware. An optimized pipelined architecture of SHA-256 hash function has been implemented in [4] which used custom data path that enforces the reuse of modules based on which novel processor architecture was implemented. [5] proposed a SHA-256 unfolding design based on reconfigurable hardware modules. The complex linear computing of SHA-256 was reconfigured by new added computation modules. [6] implemented SHA-256 architecture based on operation rescheduling to minimize the critical path delay. [7] proposed a more effective hardware to control the SHA-256 computation, which is using the finite state machine (FSM).

The purpose of this paper is to provide a high performance parallel computation hardware architecture in ASIC of SHA-256 hash. The organization of this paper is: Section 2 describes the classic SHA-256 algorithm; Section 3 presents the proposed SHA-256 parallel computation hardware architecture. The implementation results and comparison with other designs are in Section 4. The last section provides the conclusions.

## II. SHA-256 ALGORITHM

This section describes the function of SHA-256 hash. A detailed description of the SHA-256 hashing algorithm can be found in the official NIST standard [8]. The SHA-256 computation can be divided into 2 steps. The first step is to pre-process the original messages. It involves message padding and expanding the message for the round computation. The padding means appending bits according to some rules until the total length is integer of 512-bit. Afterwards every 512-bit will be expanded to 64*32 bit for SHA-256 round computation.

Here we use "t" to indicate the number of transformation rounds.

When 0≤t≤15: $W_t = input\ message$

When 16≤t≤63:

$$W_t = \sigma_1^{\{256\}}(W_{t-2}) + W_{t-7} + \sigma_0^{\{256\}}(W_{t-15}) + W_{t-16} \quad (1)$$

The first 16 $W_t$ are input messages. And after that the others are from iterative operation. In equation (1) the σ is calculated by:

$$\sigma_0^{\{256\}}(x)=ROTR^7(x)\oplus ROTR^{18}(x)\oplus SHR^3(x) \quad (2)$$

$$\sigma_1^{\{256\}}(x)=ROTR^{17}(x)\oplus ROTR^{19}(x)\oplus SHR^{10}(x) \quad (3)$$

In equation (2) and (3), the $ROTR^n(x)$ means a right rotation of x by n bits, and $SHR^n(x)$ means shift right of x by n bits.

The whole SHA-256 computation is showed in figure 1.

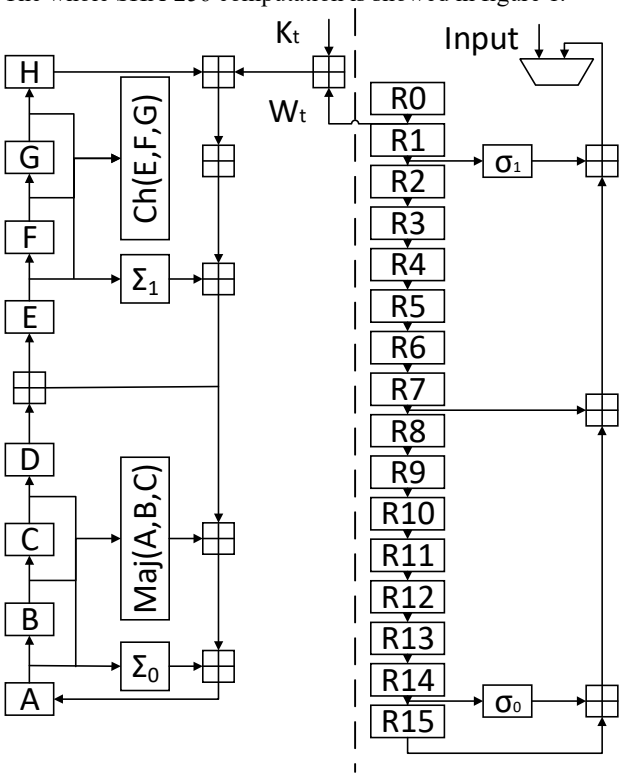


**Figure 1.** SHA-256 hashing algorithm

The second step is called round computation shown in left part of dotted line in figure 1 is to obtain the "a"~ "h", which can be calculated by:

$$T_1 = h + \sum_1^{\{256\}}(e) + Ch(e,f,g) + K_t^{\{256\}} + W_t \quad (4)$$

$$T_2 = \sum_0^{\{256\}}(a) + Maj(a,b,c) \quad (5)$$

$$h = g \quad (6)$$

$$g = f \quad (7)$$

$$f = e \quad (8)$$

$$e = d + T_1 \quad (9)$$

$$d = c \quad (10)$$

$$c = b \quad (11)$$

$$b = a \quad (12)$$

$$a = T_1 + T_2 \quad (13)$$

The first round of a, b, c, d, e, f, g and h are assigned by the initial value of SHA-256 definition. The $K_t$ is a constant in 32-bit and 64 values overall. And the four function computation are showed in equation (14)-(17):

$$Ch(x,y,z) = (x \wedge y) \oplus (\neg x \wedge y) \quad (14)$$

$$Maj(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) \quad (15)$$

$$\sum_0^{\{256\}}(x) = ROTR^2(x) \oplus ROTR^{13}(x) \oplus ROTR^{22}(x) \quad (16)$$

$$\sum_1^{\{256\}}(x) = ROTR^6(x) \oplus ROTR^{11}(x) \oplus ROTR^{25}(x) \quad (17)$$

The $\oplus$ represent bitwise XOR operation, the $\wedge$ represent bitwise AND operation and the $\neg$ bitwise complement operation.

Each round of (6)-(13) can generate 8 hash values, they were showed in the right part of figure 1 and calculated by:

$$H_0^{(i)} = a + H_0^{(i-1)} \quad (18)$$

$$H_1^{(i)} = b + H_1^{(i-1)} \quad (19)$$

$$H_2^{(i)} = c + H_2^{(i-1)} \quad (20)$$

$$H_3^{(i)} = d + H_3^{(i-1)} \quad (21)$$

$$H_4^{(i)} = e + H_e^{(i-1)} \quad (22)$$

$$H_5^{(i)} = f + H_5^{(i-1)} \quad (23)$$

$$H_6^{(i)} = g + H_6^{(i-1)} \quad (24)$$

$$H_7^{(i)} = h + H_7^{(i-1)} \quad (25)$$

The final output is obtained by the 64th round hash value as below:

$$output = H_0||H_1||H_2||H_3||H_4||H_5||H_6||H_7 \quad (26)$$

## III. PROPOSED DESIGN

From the SHA-256 hashing algorithm we can see what limits the computation speed most is equations (4)-(17). In fact the equations (1)-(3) can be done quite earlier but has to wait a very long time for (4)-(17) to finish a round of whole SHA-256 computation. The proposed design is to optimize the equations (4)-(17) in order to get a better performance in 3 steps.

### *A. Critical path analysis*

To analyse the critical path of SHA-256 we unfold the computation steps, which are showed in figure 2.

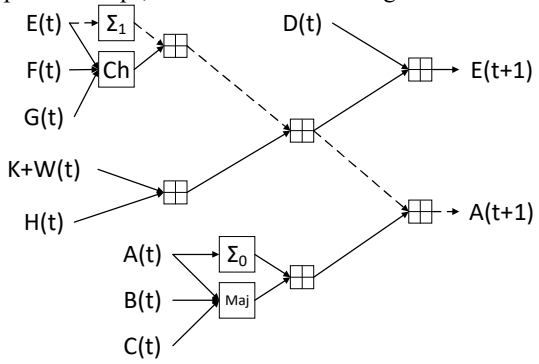


**Figure 2.** Critical path

The symbols in figure 2 are same as figure 1, and the "+" represents the addition modulo $2^{32}$. As known the most problematic characteristic of SHA-256 is the addition modulo $2^{32}$, which slows down the speed heavily versus the other

calculation steps [9]. Based on this fact, we find the most critical path in SHA-256 round calculation. As figure 2 shows, the most critical path is showed in dotted line, which means from calculation e(t) to obtain a(t+1) in each round is the longest path worth optimizing (same as other long path with 3 "+" calculations).

### *B. Break Critical path*

To optimize the SHA-256 computation it needs to break the critical path into shorter paths thus make it possible for the whole computation chain to work at a higher frequency. For this need, we consider each addition modulo $2^{32}$ calculation as one basic unit of SHA-256 calculation in each round. To separate all calculations we insert FFs to each minimum unit.

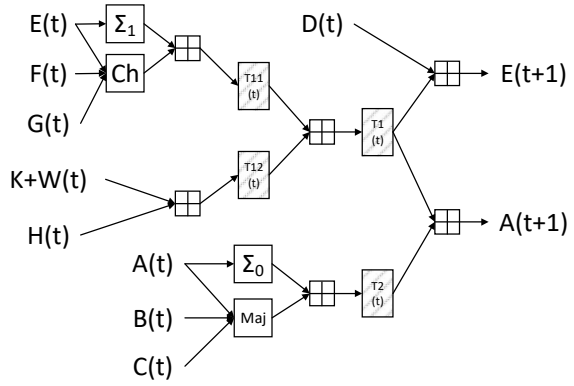


**Figure 3.** FFs insertion

As figure 3 shows, we insert 4 32-bits FFs (The grey rectangles) in each round of SHA-256 to separate all basic units. With the insertion, there is only one addition modulo $2^{32}$ calculation between each two FFs. And the T1 (t) and T2 (t) mean the FFs for T1 and T2 calculation as what equation (4) and (5) show, n means the round number of whole SHA-256 computation. T11 (t) and T12 (t) are intermediate results for T1, "t" the same as before.

Because the long path is broken， we can run almost 3 times faster than before, but each round will take 3 cycles now.

### *C. Reschedule with parallel pipeline*

In standard SHA-256 computation, variable "A" to "H" are calculated one after another. But most of them is just a bit shifting operation which is much easier compared to addition modulo $2^{32}$ calculation. Consequently, we reschedule the SHA-256 computation based on step B's basic units with parallel pipeline.

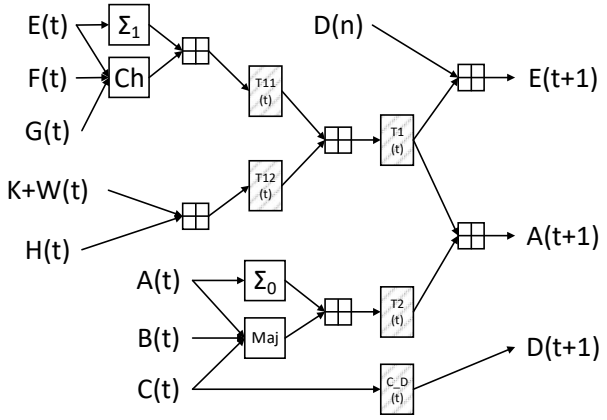


**Figure 4.** Variable update

As figure 4 shows, to achieve parallel computation, a key intermediate FF "C_D" is added, to preserve C and update D later. With this rescheduling the SHA-256 is separated into 3 individual steps to update "A" ~ "H", in subsequent 3 cycles. The functional diagram in sequence is showed in figure 5.

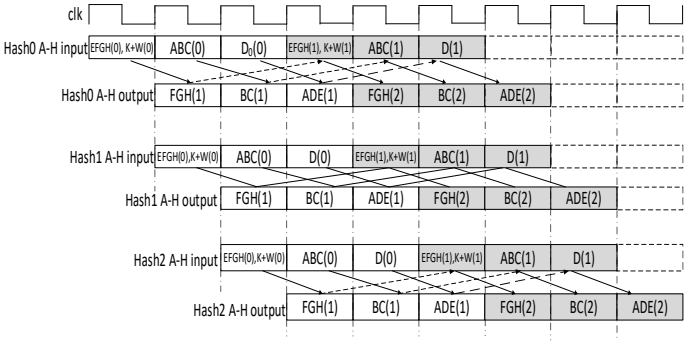


**Figure 5.** rescheduled parallel pipeline SHA-256

As the figure 5 shows, with parallel pipelines three SHA-256 hashes can be calculated at the same time. The hardware architecture is showed in figure 6.

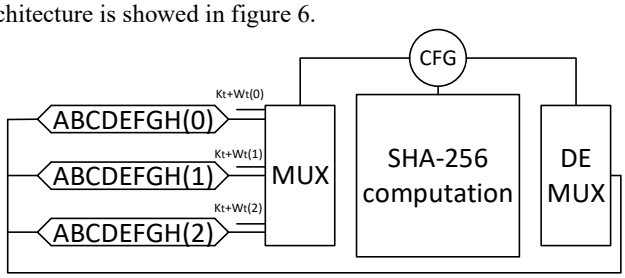


**Figure 6.** Parallel computation SHA-256 hardware architecture

The proposed architecture gains 3 times performance with a 3 times higher clock frequency, compared to the standard architecture.

## IV. RESULT AND DISCUSSION

The proposed high performance parallel computation of SHA-256 is successfully implemented in Verilog. The hardware architecture is fully verified at RTL level and synthesized with Intel 14nm technology lib. The comparison results are showed below:

**TABLE 1.** HARDWARE COMPARISON RESULTS

| | Clock(ps) | Area(μm2) | Power(mW) |
|---|---|---|---|
| **Standard** | 1959 | 4916.7 | 3.3794 |
| **Proposed** | 653 | 14272.6 | 6.855 |

It is clear to see that the proposed high performance parallel computation hardware architecture of SHA-256 is 3 times faster than the standard architecture as we expected.

The area cost to achieve this improvement is 2.90 times compare to standard SHA-256, which is because there are reused function modules to save area.

The power of proposed parallel SHA-256 is just 2.03 times of standard SHA-256 to have same 3 times output. That's because the sequential logic consumes much bigger power than the combinational logic, and the proposed architecture can exactly save much sequential logic.

## V. CONCLUSIONS

In this paper, we proposed a high performance parallel computation hardware architecture in AISC of SHA-256. The standard SHA-256 is rescheduled based on hardware characterises. Consequently, the critical path of SHA-256 is found and pipelined by DFFs added. Based on the comparison with standard SHA-256 designs, the proposed design can have 3 times speed improvement at 2.90 times area cost and gain 50.7% improvement in power.

## REFERENCES


[1] (2008) The Bitcoin website. [Online]. Available: http://bitcoin.org/

[2] X. Cao and M. O'Neill, "Application-oriented SHA-256 hardware design for low-cost RFID," IEEE International Symposium on Circuits and Systems, 2012, pp. 1412-1415.

[3] W. L. Harrison, A. M. Procter and G. Allwein, "Model-driven design & synthesis of the SHA-256 cryptographic hash function in rewire," International Symposium on Rapid System Prototyping (RSP), 2016, pp. 1-7.

[4] M. Padhi and R. Chaudhari, "An optimized pipelined architecture of SHA-256 hash function," 7th International Symposium on Embedded Computing and System Design (ISED), 2017, pp. 1-4.

[5] S. Suhaili and T. Watanabe, "Design of high-throughput SHA-256 hash function based on FPGA," 6th International Conference on Electrical Engineering and Informatics (ICEEI), 2017, pp. 1-6.

[6] I. Algredo-Badillo, C. Feregrino-Uribe, R. Cumplido and M Morales-Sandoval, "FPGA-based implementation alternatives for the inner loop of the Secure Hash Algorithm SHA-256," *Microprocessors & Microsystems,* vol. 37, pp. 750-757, Jun. 2013.

[7] H. Mestiri, F. Kahri, B. Bouallegue, and M. Machhout, "Efficient FPGA Hardware Implementation of Secure Hash Function SHA-2," *International Journal of Computer Network and Information Security,* vol. 7, pp. 9-15, Dec. 2014.

[8] *Secure Hash Standard (SHS),* N. I. of Standards and Technology, FIBS PUB 180-4, 2012.

[9] L. Dadda, M. Macchetti and J. Owen, "The design of a high speed ASIC unit for the hash function SHA-256 (384, 512)," Proceedings Design, Automation and Test in Europe Conference and Exhibition, 2004, pp. 70-75.


**Xiaoyong, Zhang** was born in China, Nov 5th 1980. Bachelor. The first bachelor degree was earned in automation, in School of Marine Science and Technology, Northwestern Polytechnical University, Shaanxi Province, China, in 2003, and the second bachelor degree was earned in electronic science and technology, in Institute of Microelectronics, Tsinghua University, Beijing City, China, in 2005.

He has worked in Xi'an, Shaanxi Province, China, since 2005, in the wireless department for Infineon Technology at first, which was acquired later by Intel in 2011. His current job is hardware design and validation.

**Ruizhen, Wu** was born in China, Jan 1st 1986. PhD. The PhD was earned in School of Microelectronics of XIDIAN University, Shaanxi Province, China, in 2014. The major field of study is Asynchronous Circuits design and 5G CODEC.

He has worked in Hangzhou, Zhejiang Province, China, since 2014, in the 2012 communication lab of Huawei at first, and Intel iCDG in Xi'an, Shaanxi Province since 2016.

**Mingming, Wang** was born in China, Oct 1st 1986. Master. The Master degree was earned in Computer Application Technology in Xi'an University of posts & Telecommunications, Shaanxi Province, China, in 2011, and the bachelor degree was earned in electronic science and technology, in Xi'an University of posts & Telecommunications, Shaanxi Province, China in 2008.

He has worked in Intel iCDG Xi'an, Shaanxi Province, China since 2011.

**Lin, Wang** was born in China, Dec. 1st. 1971. Master of Sci. The master degree was earned in Dept. of Electrical Engineering, Fudan University, Shanghai, China, in 1998. His majority is microelectronics and physics on semiconductor and semiconductor device.

He worked in Shanghai Nortel Semiconductor and Broadcom, focusing on communication chip development after his graduation.

He is now the Director of Digital Design in Intel Xi'an, Shaanxi Province, China, ever since 2005.