International Journal of Electronics Signals and Systems

Volume 2 | Issue 3 Article 1

January 2013

IMPLEMENTING SHA-224/256 ALGORITHM FOR SECURE COMMITMENT SCHEME APPLICATIONS USING FPGA

V. VENKATA SAI KARTHIK
Audisankara College of Engineering & Technology. Nellore Dt, AP, India, vsaikarthik5712@gmail.com

T. VENKATA SRIDHAR
Audisankara College of Engineering & Technology. Nellore Dt, AP, India,
venkatasridhar.ece@audisankara.com

IMPLEMENTING SHA-224/256 ALGORITHM FOR SECURE COMMITMENT SCHEME APPLICATIONS USING FPGA

[1]V.VENKATA SAI KARTHIK & [2]T.VENKATA SRIDHAR

[1&2]Audisankara College of Engineering & Technology. Nellore Dt, AP, India
Email: vsaikarthik5712@gmail.com & venkatasridhar.ece@audisankara.com

Abstract - This paper uses the similarity between SHA-224 and SHA-256 algorithms to design the SHA-224/256 IP core oriented Digital Signature. The IP core uses parallel structure and pipeline technology to simplify the hardware design and improve the speed by 26%. Finally this IP core is implemented on the Altera's FPGA EP2C20F484C6 chip. And its simulation result can run rightly under the 100MHz frequency. This IP core can be widely used in the data integrity and consistency verification, pseudo random number generation and other areas of cryptography.

Keywords- *Digital Signature; SHA-224/256; IP core; FPGA.*

I. INTRODUCTION

SHS (Secure Hash Standard) is a hash algorithm (FIPS PUB 180-1), released by United States National Institute of Standards and Technology (NIST) in 1995. Because the algorithm is collision-resistance and non-reversible, it is widely used in the information security field at present, which are more well-known SSL, IPSec and PKCS. But as people study the algorithm in-depth, its security has also been questioned and threatened[1][2]. This has prompted NIST release the latest SHS specifications (FIPS PUB 180-3) in October 2008. With the previous version (FIPS PUB 180-2 CHANGE NOTICE, August 2002), the biggest difference is that SHA-224 algorithm has been formally included in the SHS standard. Because SHS algorithm itself is a very complex algorithm, its calculation is to a larger quantity, and each iteration needs to rely on the previous calculation, it is often used hardware implementation to increase the processing speed[3] . This paper uses the similarity between SHA-224 and SHA-256 algorithm and hardware description language to design and implement the time division multiplexing SHA-224/256 IP core. The IP core will not only be able to generate digital signature to protect the information integrity and security, but also generate the double-key of 3DES algorithm to provide a more reliable, safe, and convenient keys. So it has a broadapplication prospects. A typical application of SHA in the digital signature algorithm is shown in Fig.1.

II. COMPUTER BUS MEMORY SYSTEM DESIGN

SHA-224 and SHA-256 are the two kinds of algorithms in the SHS standard (FIPS PUB 180-3). They can handle input messages whose length is less than 2^{64} bits, but the outputs are separately compressed into 224 bits and 256 bits. SHA-224 algorithm and SHA-256 algorithm have only two differences: first, the initialized hash values are different; second, the results of SHA-224are needed to be truncated.

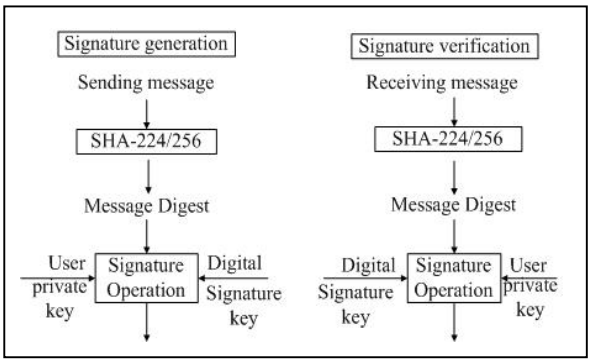

Figure1. Application Diagram of SHA-224/256 In Digital Signatures

SHA-256 algorithm has two steps to complete the calculation. The first step is to preprocess the input message to be filled and divided, generating 512 bits blocks. The second step is to calculate the hash value, that is to say, every block operates to produce the final results. After dividing blocks, every block messages can be processed by the following methods. And the details are described in reference [4].

(1) Giving K_0, K_1, …, K_{63} sixty-four 32-bits K the initial value.

(2) Giving H_0, H_1, H_2, H_3, H_4, H_5, H_6, H_7eight 32-bits variables the specified initial hash values. Every block messages is to do the from step (3) to(7).

(3) Divide the 512bits block into sixteen 32-bits words W_0, W_1, …, W_{15}.

(4) For i = 16 to 63

S_0=ROTR^7(Wi-15)

⊕ROTR^{18}(Wi-15) SHR^3(Wi-15)

S1=ROTR^7(W_{i-2})⊕ROTR^{19}(W_{i-2}) ⊕SHR^{10}(W_{i-2})

Wi=Wi-16^{+S}0+Wi-7+S_1

(5) Initialize the hash value, a = H_0, b = H_1, c = H_2, d = H_3, e = H_4, f = H_5, g = H_6, h = H_7.

(6) For i = 0 to 63

S_0=ROTR^2(a)⊕ROTR^{13}(a) ⊕ROTR^{22}(a)

maj=(a∧b) ⊕(a∧c) ⊕(b∧c)

t_2=s_0+maj

S1=ROTR^6(e)⊕ROTR^{11}(e) ⊕ROTR^{25}(e)

Ch=(e∧f) ⊕(¬e∧g)

t_1=h+s_1+ch+K_t+W_t

h=g, g=f, f=e, e=d+t_1, d=c, c=b, b=a, a=t_1+t

(7) Add the hash values a, b, c, d, e, f, g, h respectively to the variables H_0, H_1, H_2, H_3, H_4, H_5, H_6 and H_7.

(8) Output 256-bits compressed code H_0|| H_1|| H_2|| H_3|| H_4|| H_5|| H_6|| H_7.

The signs ∧ , ⊕, ¬ , + respectively represents bitwise AND, XOR, NOT and 32-bits addition operation. And ROTR^m(Wn) represents that Wn rotates right m bits, SHR^p(Wq) represents that Wq rotates right p bits. The sign | represents bitwise connect.

As can be seen from the description of the algorithm, the core of the whole algorithm is the second step calculating the hash values. The first step can be achieved by the upper software. Therefore, several issues need to be solved for the calculation of hash values.

- Determine the data bus width. Because the message length handled by the algorithm is variable, the external data bus width and the corresponding control core. From the third step and sixth step of the algorithm, the relationship between production and consumption among them entirely can be handled by the parallel architecture.
- The multiplexing of IP core, a group of registers is used to achieve the time-division multiplexing of SHA-224 and SHA-256 algorithms.
- Performance and area optimization, pipelining and parallel computing architecture will be used to design simple structure and fast IP core.

III. SYSTEM DESIGN AND IMPLEMENTATION

Every sub-module of the entire IP core is designed according to the data flow of the SHA-256 algorithm.

First, determine the interfaces of the IP core. Considering the portability of the IP core, 32-bits data bus and 11-bits control bus. Control bus includes clock signal, reset signal, control enable signal, function selection signal, control signal and state signal. Next, According the relationship between production and consumption of data flow, the IP core can be divided into the Data pool, ALU(Arithmetic Logic Unit), Register files and Counter four parts(shown in Fig.2). Data pool is used to save the constant and W_t in the algorithm, including the initial hash value, key value, and the values of the input words and the expansion words. ALU is used to complete the arithmetic and logic operations. Register files are used as the dedicated registers to save the values of a , b , c , d , e , f , g , h. Counter is added 1 in every clock rising edge arrives to meet the iterative control.

When input the corresponding data and control signal to the IP core, the IP core does iterative processing in a block (512 bits).

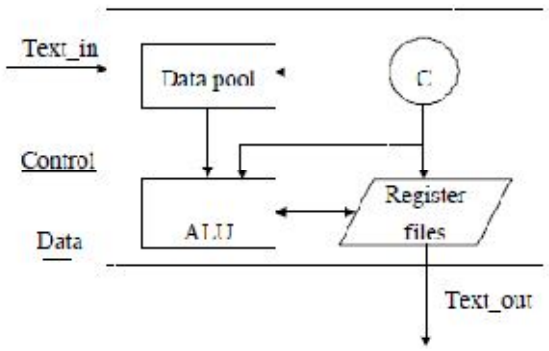

Figure 2 : IP Core Architecture

The counter is cleared after every 64 clocks to maintain synchronization between itself and the word. Its data flow is the following: Data pool gives Wt and Kt under the control of the Counter and sends them to ALU. ALU does the corresponding arithmetic and logic operations after receiving the data, and save the results to register files until the end of this iteration. At the beginning, the blocks, the words and the end, the Register files you need to provide the corresponding results for ALU or the output bus under the control of external control signals and the Counter

A. Data pool

The Data pool consists of look-up table unit and shift register unit. Look-up table unit is responsible for

looking up the key value of this iteration according to the counter value. Shift register unit is responsible for completing the expansion from 16 words to 64 words. There is sixteen 32-bits registers, respectively recorded as W_0 ,W_1,…, W14, Wt. When each 512-bits block is processed, these registers assign and flow according to the counter value.

The assignment will be done less than 16, which is assigning the first i 32-bits word to W_i and W_t .When the counter is greater than or equal to 16, the flow operation will be done, that is, the pipelining is used among the registers to transfer W_{i+1} to W_i ($0 \le i \le 13$) and W_{14} is equal to W_t after every clock in order to simplify the calculation of Wt circuit which is satisfying the following expression.

$$N\_W_1 = ROTR^7(W_1) \oplus ROTR^{18}(W_1) \oplus SHR^3(W_1)$$

$$N\_W_{14} = ROTR^{17}(W_{14}) \oplus ROTR^{19}(W_{14}) \oplus SHR^{10}(W_{14})$$

$$Wt = N\_W_{14} + W_9 + N\_W_1 + W_0$$

B. ALU

In the processing every word, logical operations in the every iteration may be a simple combination circuit, while the arithmetic only needs 32 bits adder to complete.

From the description of the algorithm, calculating 'a' value is the longest path (It has five additions). So CSA (Carry Save Adder) of the parallel structure is used to reduce the carry signal delay[5][6] brought by the number of additions in order to improve the entire IP core speed. Due to every summand is also the intermediate result of the logical operation, it is as the input of the second level CSA. And the final calculation results are given by the CPA (Carry-Propagate Adder). The addition structure of 'a' value is shown in Fig.3

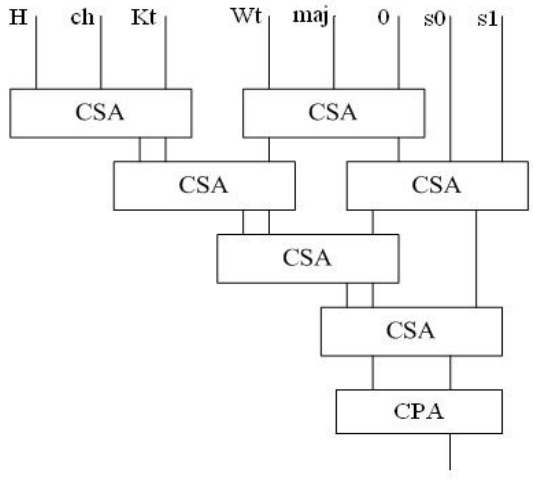

Figure3. The Addition Structure of 'a' Value

TABLE1. SYNTHESIS RESULTS

Project name	Default Comprehensive Result	Parallel CSA Structure
Total logic elements	2,160	2,646
Logic registers	1,124	1,124
Total pins	75	75
Total memory bits	4,073	4073
Actual fmax	81.06MHz	103.08MHz

IV. SYNTHESIS AND SIMULATION

In this design, this IP core is described by Verilog HDL language and has been implemented to FPGA Altera Cyclone EP2C35F672C6. Then it is synthesized and routed on the QuartusII 8.0. Finally it is simulated by ModelSim[7] to test if the IP core is correct.

A. Synthesis results

Table 1 shows the comparison data whether or not using the CSA adder (Default comprehensive option), in which the performance is increased by 26% and the resource consumption is also increased by 26% after using the CSA adder. Taking into account the internal structure of FPGA, using the HardCopy technology [8] turns the IP core to ASIC achieving that the power consumption will be further reduced and the performance and speed will be increased by almost 50% [9].

B. Timing simulation

Under that the simulation clock is 100MHz, its simulation waveforms are shown in Fig.3 (SHA-224) and Fig.4 (SHA-256), in which the input test string is : 12345678901234567890123456789012345678901234567890123456 (The length is 448 bits), the result of SHA-224

Algorithm is :

e1cb99de_19ad01ca_c1cad48b_f5230169_fd18aaab_1fb2b1ec_a48cd7d5, the result of SHA-256

algorithm is :

0be66ce7_2c2467e7_93202906_00067230_66617916_22e0ca9a_df4a8955_b2ed189c. This IP core achieves the desired purpose both in function and timing (consistent with the results of Freeware Hash & CRC [10]), while the delay and the glitch phenomenon in the simulation waveform can also accurately reflect the characteristics of the circuit delay.

.

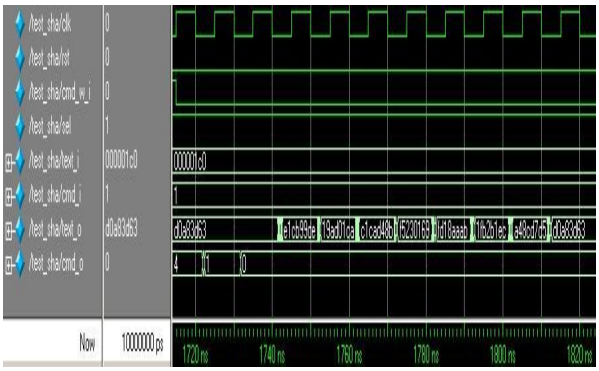

Figure4. The Simulation Result Of SHA-224

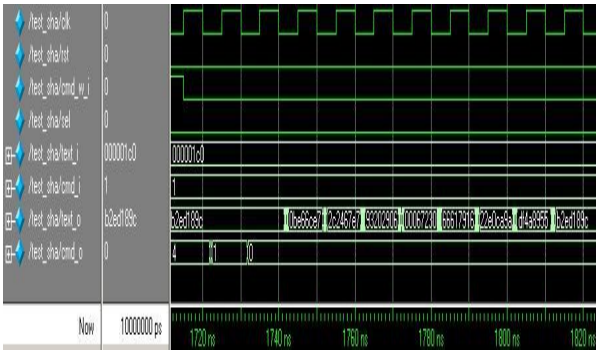

Figure5. The Simulation Result Of SHA-256

.V. CONCLUSION

This paper uses the similarity between SHA-224 and SHA-256 algorithms to design a time division multiplexing IP core. 32-bits data bus makes this design has a friendly data interface, and the whole design has a simple hardware structure and fast running speed and can be widely used in digital signatures and 3DES key generation systems.

REFERENCES

[1] Wang Xiaoyun, Yu Hongbo and Yiqun Lisa Yin, Efficient Collision Search Attacks on SHA-0[C], CRYPTO 2005[2]

[2] Wang Xiaoyun, Yiqun Lisa Yin and Yu Hongbo, Finding Collisions in the Full SHA-1[C], CRYPTO 2005[3]

[3] Huang Chun, Bai Guoqiang, Chen Hongyi. Fast Implementation of the hardware structure of SHA-1 algorithm[J]. Journal of Tsinghua University 2005(45)1, pp.:123-125.

[4] FIPS PUB 180-3, Secure Hash Standard[S], National Institute of Standards and Technology (NIST), 2008

[5] Jian Honglun. Proficient VerilogHDL: The example explanation of IC design core technology[M]. Electronics Industry Pres, 2005.10

[6] Yang Xiaohui, Dai Zibin. FPGA-based implementation of SHA-256 algorithm[J], Microcomputer Information, 2006(22)4-2, pp.146-148.

[7] Jiang Hao, Li Zheying. FPGA design flow based on a variety of EDA tools[J], Microcomputer Information, 2007(23)11-2, pp.:201-203

[8] HardCopy II Device Handbook, Volume 2[OL], http://www.altera.com.cn/literature/hb/hardcopy-ii/hc_h5v2.pdf

[9] IC Technology Seminar. FPGA modular design and Altera HardCopy II structured ASIC[J], World Electronic Components, 2007,6, pp.: 38-42

[10] febooti.com, Freeware Hash & CRC [OL],http://www.febooti.com/products/filetweak/members/hash-and-crc/

◈◈◈