World Applied Sciences Journal 16 (Special Issue on Recent Trends in VLSI Design): 22-28, 2012
ISSN 1818-4952
© IDOSI Publications, 2012

Single Chip Implementation of Serial Structure SHA-1

[1]Yogesh Pratap, [2]Manoj Kumar and [2]Jitendra Kumar

[1]Semiconductor Device Research Laboratory,
University of Delhi South Campus, Benito Juarez Road, New Delhi 110021, India
[2]Centre for Development of Advanced Computing (CDAC), Noida (U.P.), India

Abstract: Due to continuous growing network, the network security problem also increasing day by day, the demand of secure communication has been highlighted because of people is using service such as Internet e-commerce, e-government mobile network, online transactions, net banking and many more. Secure Hash Algorithm (SHA) is the one of the most important algorithm for network security. Basically it is used for message digest. SHA-1 is the version of SHA series with a 160-bits message digest. SHA-1 creates a digest of 160 bits from taking a multiple block message and each block length is 512-bits. Like other SHA series algorithms SHA-1 also have initial hash value of 160-bits storing into five register having 32-bit memory each. This paper presents layout implementation of serial architecture secure hash algorithm-1(SHA-1) which is area efficient with respect to simple parallel architecture of secure hash algorithm-1. The design has been written in Verilog HDL (hardware description language) and synthesized using TSMC 180 nanometer technology library which is capable of operating at 426MHz frequency. The area of the automated generated layout is 8956.55mm^2 and clock speed is 42 MHz.

Key words: SHA-1 · MD5 · Security · IPSEC · SSL · LVS · DRC

INTRODUCTION

With the advancement of technology the security issues also increase. Likewise we use the on chip cryptographic algorithm on SIM cards. To provide the better security and complexity to our design we use different algorithms together. The key benefit of single chip implementation of Secure Hash Algorithm is that it provides the better integrity and better performance. Cryptography algorithms are becoming more necessary to ensure secure data transmission, which can be used in several applications [1]. For secure, high performance, cost effective and better speed communication we have implemented Secure Hash Algorithms into a single chip. System-on chip (SOC) integrates a variety of components of a computer or other electronic system into a single integrated circuit (chip). Also we can say we are integrating a large design into a single integrated circuit. It may contain digital, analog, mixed-signal; all on one chip [2]. Almost all the embedded systems are the application area of System-On chip. This single SHA-1 chip has created using VLSI technology. Very-large-scale integration (VLSI) is the process of creating integrated circuits by combining more than ten thousands of transistors into a single chip.

Secure communication is one of the most important parameter nowadays. Cryptographic algorithms (like AES, DES etc.) are used for secure communication. These algorithms are used encryption and decryption techniques to protect our information so that in any communication or computer network only authorized people can send or retrieve information. Information security has three major goals for message authentication that is referred as security goals. They are confidentiality, interiority and availability. Secure hash algorithm (SHA-1) is one of the algorithms of SHA series which is used for internet protocol security (IPSEC) and Secure sockets layer protocol (SSL) as the basic for message authentication code technique HMAC because of having hash function. In IPSEC protocol they each packet of data authenticated between two sending and receiving points. Hence we also call it end to end security. Verilog hardware

Corresponding Author: Yogesh Pratap, Semiconductor Device Research Laboratory, University of Delhi South Campus Benito Juarez Road, New Delhi, 110021, India. E-mail: yogi.pratap87@gmail.com.

description language has been used for describing the SHA-1 algorithm and a Mentor tool has been used for generation of chip layout.

The paper is organized as follows: a hash function is described in section 2. Secure hash algorithm (SHA-1) is presented in section 3. The proposed serial architecture of SHA-1 is described in section 4. Result is shown in section 5 and the last section is conclusion section.

Hash Function: Hash function produces a fixed-size message by taking a variable size message. The fixed size output is referred as hash code H (M). It is very important cryptographic primitive which is used in digital signature scheme. It can be applied having any size of data.

Hash function is defined as:

H = H (M)

Where

H = Hash value

M = Variable length message

H (M) = Fixed length hash value

Hash function is secure against any cryptographic attacks because of having thee important property given below.

One way property:

- If H (M) = h, for any given value h, it is computationally infeasible to find M.

Weak collision resistance:

- If H (a) = H (b), for any given block b, it is computationally infeasible to find a ≠ b.

Weak collision resistance:

- If H (a) = H (b), for any given block b, it is computationally infeasible to find pair (a, b) [3].

From Figure 1

- C Vo = IV initial n-bit value
- C Vi = f (CVi-1, Yi-1), 1 <= I <= L
- H (M) = CVL

Where the input to the hash function is a message M consisting of the blocks Yo, Y1... YL1.

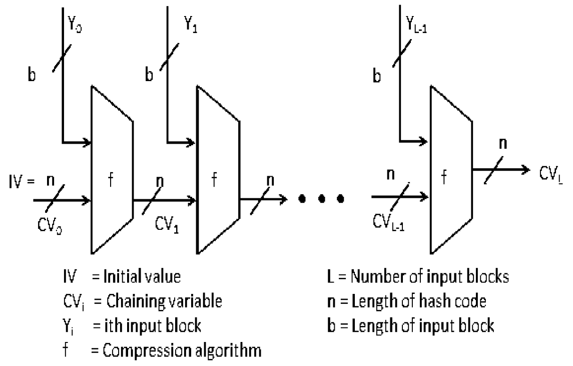

Fig. 1: Structure of secure hash code

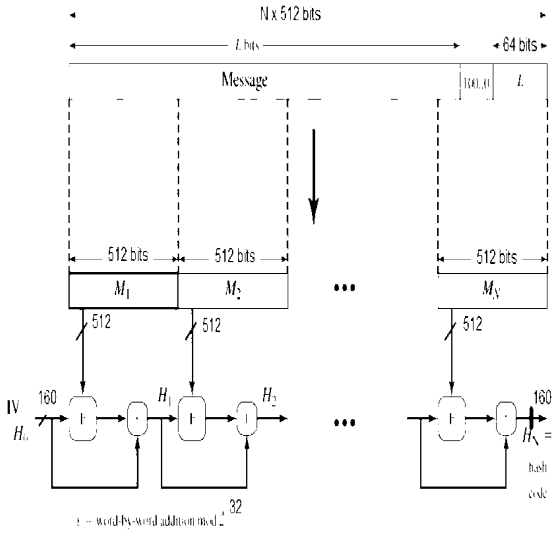

Fig. 2: Message Digest Generation

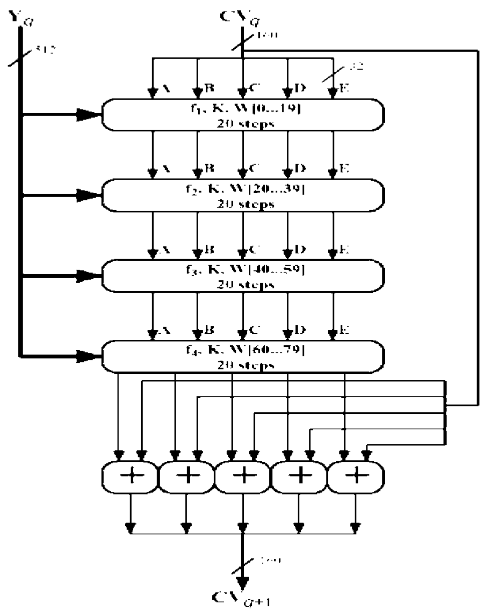

Fig. 3: Compression function

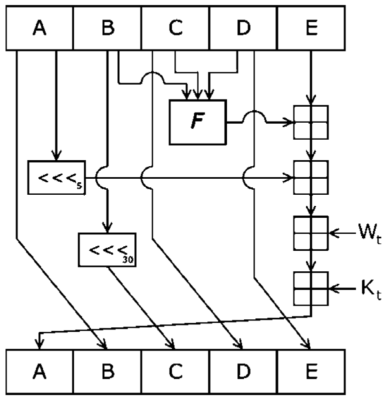

Fig. 4: Sha-1 round function

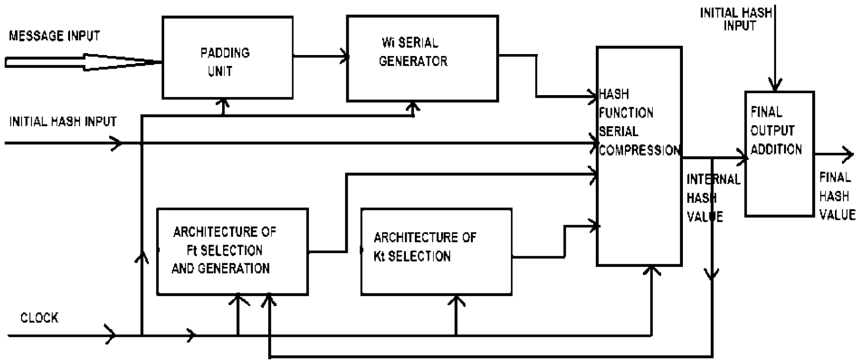

Fig. 5: Block diagram of serial SHA1

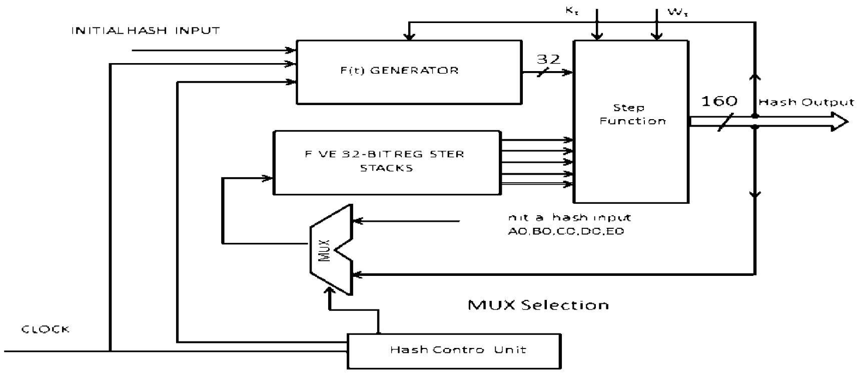

Fig. 6: Architecture of Serial Compression function

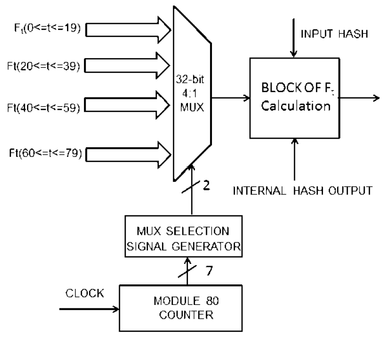

Fig. 7: Architecture of function generation

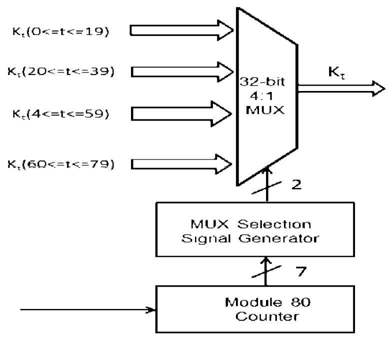

Fig. 8: Architecture of Kt selection

Table 1: Function and constant value

Step	Function name	Function value	Value of Kt
(0=t=19)	F1 = f(t,B,C,D)	(B ^ C) v (B'^D)	5A827999
(20=t=39)	F2 = f(t,B,C,D)	(B xor C xor D)	6ED9EBA1
(40=t=59)	F3 = f(t,B,C,D)	(B ^ C) v (B^D) v(C^D)	8F1BBCDC
(60=t=79)	F4 = f(t,B,C,D)	(B xor C xor D)	CA62C1D6

Secure Hash Algorithm-1: National institute of standards and technology (NIST) has developed secure hash algorithm (SHA). SHA series have algorithm SHA-1, SHA-256, SHA-384 and SHA-512 with different-different fixed output length values 160,256,384 an 512. SHA-1 algorithm takes input maximum length up to 2^{64}-bit and generate 160-bit output called message digest. In this algorithm input is processed in 512-bit blocks. Algorithm [4] processing includes the following steps:

Append Padding Bits: Padding is used in the algorithm to make message length congruent to 448 modulo 512 [length = 448(mod 512)]. It is necessary even if message having the desired length. Padding consist of a single 1- bit follow by necessary number of 0-bits,range of padding bits are 1 to 512.

Append Lenth: The block of 64 binary bits is added to the end of the original message after appending.

Initialize Hash Buffer: The predefined 160- bit buffer which represented as five 32-bits resisters (A, B, C, D and E) is used to hold the intermediate and final output digest hash value [3]. The predefined hash value in hexadecimal is

A	=	67452301
B	=	EFCDAB89
C	=	98BADCEF
D	=	10325476
E	=	C3D2E1F0

Process Message in 16 Words(512-bit)Block : The main module of algorithm consists of four rounds and each round having 20 steps of processing. The four round having same structure but each uses a different primitive logical function. These logic functions are described below in the table:

The logical operators (AND, OR, NOT, XOR) are represented by the symbols (^, v, ', XOR).

Each round has 160-bit buffer value A B C D E as input and updates the contents of the buffer. The buffer has the intermediate hash value at input of the first round that is Hi-1. A 32-bit value word (Wt) is used in each round derived from the current 512-bit block being processed (Mi). Each round also makes use of an additive constant Kt where 0<=t<=79 indicates one of the 80 rounds. These words represent the first thirty-two bits of the fractional parts of the cube roots of the first four prime numbers. The constants provide a "randomized" set of 32-bit patterns, which should eliminate any regularity in the input data [3]. The input of the first round is added to the output of the fourth round which gives buffer value A B C D E that calculates for the next 512-bit block. Here addition is modulo 2^{32}.

Round Function: Each primitive function takes three 32-bit words as input and produces a 32-bit word output. Each function performs a set of bit wise logical operations; that is, the nth bit of the three inputs.

The functions can be summarized as follows [3]:

- A, B, C, D, E _ $(E + f(t, B,C,D) + S^5 (A) + Wt + Kt)$,A, S^{30} (B),C,D

Where

- A, B, C, D, E = the five words of the buffer,
- T = step number; 0<= t <=79,
- f (t, B,C,D) = primitive logic function of logic t,
- S^k = circular left shift of the 32-bit argument by k bits
- Wt = a 32-bit word derived from the current 512-bit input block
- Kt = an additive constant; four distinct values are used
- '+ '= additive module of 2^{32}

Output: After all N 512-bit blocks have been processed, the output from the nth stage is the 160-bit message digest.

After summarize the behavior of SHA-1 as follows:

H_0 = IV
H_i = $SUM_{64}(H_{i-1}$, ABCDE)
MD = H_N

Where

- IV= initial value of the ABCDE buffer
- ABCDE= the output of the last round of processing of the ith message block

- N= the number of blocks in the message (including padding and length fields)
- SUM_{64}= Addition modulo 232 performed separately on each word of the pair of inputs
- MD= final message digest value

Serial Structure of SHA-1: This given below architecture shows the serial implementation of secure hash algorithm-1 having padding unit, Wi serial generation unit, hash serial compression function, architecture of Kt and function generation and selection. The above flow graph of SHA- 1 algorithm assumes that the sequence Wo, Wl, W2, W79 is implemented as an array of eighty 32-bit words.. In this architecture an input message is padded. The padding unit pads 1 and series of 0, the length of the message to the input message and generate blocks of 512-bit message. Wi serial generator generate W0 to W79 of 32-bit which is used in serial compression function. Function generation block generates function Ft using internal hash value.

The value of function of each 20 round will be deferent - different. It will be given in the Table 1. For the first twenty round its value will be F1 and 20 to 39 rounds the value will be F2, 40to 59 the value is F3 and the last rounds the value assigned as F4 and Architecture of constant Kt selection selects the constant value for every 20 round. For first 20 rounds the value is K1 and 20 to 39 rounds Kt is equal to K2, for 40 to 59 rounds k3 and last 20 rounds k4. Hash serial compression function has 80 step functions generating internal hash value of 160-bits and finally 160-bit adder add internal hash value and initial hash input to produce final hash value. Here we take only input having length up to 448-bit.The hash control unit generate the control signals to control all the function block of structure.

The given below structures are show the internal sub module of serial architecture SHA-1 algorithm.

Architecture of Serial Compression Function: The given below figure describe internally structure of compression function. In this architecture step function generate internal hash value taking word (Wt) from word generation block and take constant from Kt selection block. In every single step function takes new value of A, B, C, D, E which is serially store in five 32-bit shifts register stacks. For the first step function MUX select the initial hash input value using MUX selection signal generated by hash control unit.

Architecture of Function Generation: The Ft from Table-1 depends upon the round number t and Fig. 7 shows the implementation architecture. The module 80 counters count the round cycle from 0, 1, 2, 79 and the two multiplexer selection signals are derived from the counter output and then the 32-bit 4-to-I multiplexer picks one of the four function name from table to be used in the architecture described in Fig. 4(single step function or round function) and calculate the function value of given below table. After 80 rounds, the internal hash value is added to the five constants A-E from to have the final message digest.

Architecture of Kt Selection: The constant Kt from table-1 depends upon the round number t and Fig. 8 shows the implementation architecture. The module 80 counters count the round cycle from 0, 1, 2, 79 and the two multiplexer selection signals are derived from the counter output and then the 32-bit 4-to-I multiplexer picks one of the four constants from Equation given below to be used in the architecture described in Fig. 4. (single step function or round function) After 80 rounds, the internal hash value is added to the five constants A-E to have the final message digest [4].

RESULTS

The simulation result of serial SHA-1 algorithm has been presented here. To make the simulation of the design easier, write the test bench file.The principle of this test bench is a black box that uses the top design, give it some predefined simulation vectors (input message, the output message digest and the clock) and verify if the output corresponds to the valid message digest.. The serial SHA-1 code has been written in Verilog HDL and simulated using Modelsim 5.7f.

Serial Sha-1 Simulation Result: Figure 9 shows the test bench simulation result using Modelsim 5.7f. We use predefined input/output values called test vectors provided by national institute of standards and Technology (NIST) that has been used to verify the functionality of proposed design.

The Text Vector:

One Block Message Sample: Input Message: "abc" that is in form of hexadecimal 61626380 and Initial hash value:

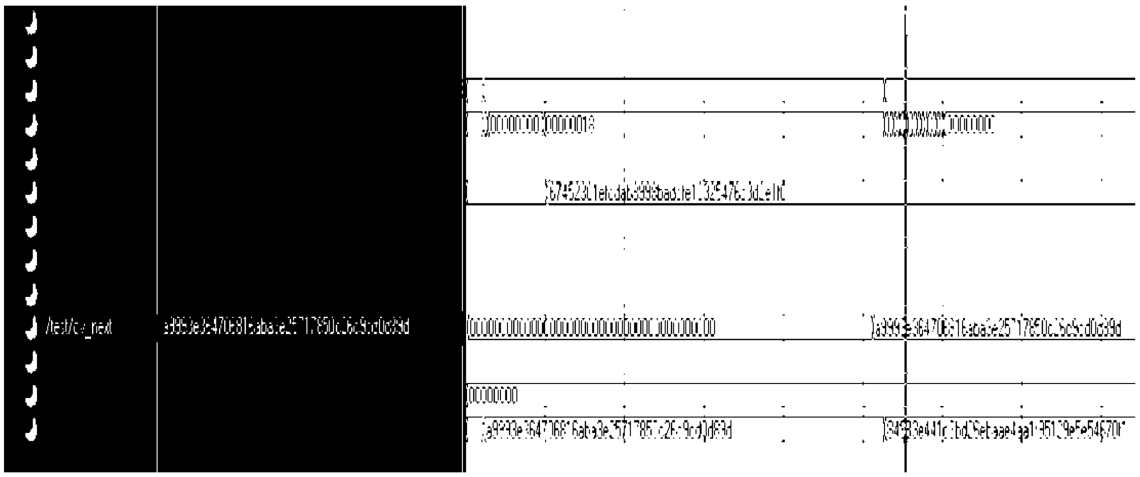

Fig. 9: Modelsim 5.7f Simulation result for serial SHA-1 text bench

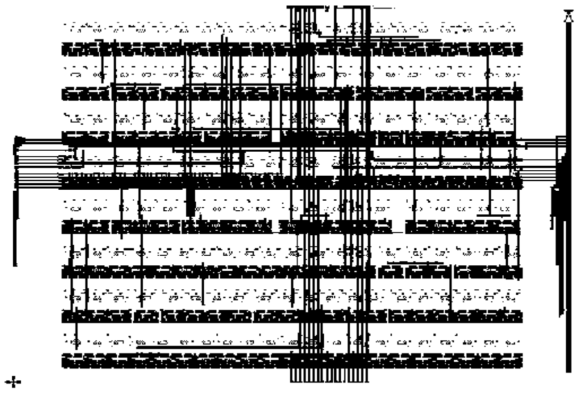

Fig. 10: Layout view of serial SHA-1 (by IC Station Layout editor)

A	=	67452301
B	=	EFCDAB89
C	=	98BADCFE
D	=	10325476
E	=	C3D2E1F0

Schematic and Layout Generation

Automatic Schematic Generation Using HDL: Synthesis process has done by Leonardo spectrum tool. It will generate the net list file. For generating the schematic we use the ASIC technology instead of FPGA technology. The net list file generated by the synthesis tool will import in IC design tool(IC Station da_ic) and set the library path also set the mapping file path. After loading the all files it will automatically generate the schematic.

Automatic Layout Generation Using HDL: For layout generation we use the Verilog netlist file and set the library path. From Automatic floor planner we have done the floor planning. By using IC station tool, the automatic layout generation of the serial SHA-1 Verilog Code is completed successfully.

After the successful complition of layout [passing all DRC (design rule check)] we varify the currectness of the layout using LVS(layout Vs schematic).

CONCLUSION

In this paper, hardware for serial architecture SHA-1 algorithm was designed, modeled and verified using the Verilog hardware description language. The Modelsim 5.7f tool was used for simulation and verification of the model. In addition, to generate the gate level schematic and layout we synthesize the code and generate the gate level net list file using the Leonardo Spectrum tool by Mentor Graphics. For generating the gate level schematic we used the Mentor Tool IC Design Architect and for layout IC Station tool was used. The LVS has also done of automated generated layout and schematic. The gate level net list generated during the synthesis phase using the TSMC 0.18 micrometer technology library is capable of operating at 426MHz frequency. The area of the automated generated layout is 8956.55mm^2 for serial SHA-1 algorithm which is less than parallel architecture SHA-1 algorithm. We expect the design to run at higher frequency if synthesized using a more efficient technology library.

REFERENCES

1. Ahmed Rady Ehab, EL-Sehely and A.M. EL-Hennawy, 2007. "Design and Implementation of area optimized AES algorithm on reconfigurable FPGA". IEEE Trans. Electron Devices, 39: 35-38.
2. Minling Zhu. and Xi Wang, Min Xu, 2011. "The Analysis for System-on Chip Application on Full Authority Digital Engine Control System", IEEE Trans. Electron Devices, 41: 2728 -2732.
3. William Stallings, 2009. "Cryptography and Network Security". PHI publication, Fourth Edition,
4. Guoping Wang, 2006. "An Efficient Implementation of SHA-1 Hash Function,", IEEE Trans. Electron Devices, 15: 575-579.