

# Implementation of Secure Hash Algorithm-1 using FPGA

Ms. Vishakha Borkar<sup>1</sup>, Mrs. A.S.Khobragade<sup>2</sup>

PG Student [VLSI], Dept of Electronics Engg, PCE, Nagpur, India<sup>1</sup>

Associate Professor, Dept. of Elect & Tele Engg, PCE, Nagpur, India<sup>2</sup>

**Abstract:** Sharing of information over the internet becoming a critical issue. To secure the data lots of techniques are available. The present work will focus on the combination of hashing, cryptography to secure the data. Hash value will be obtained from original data. Secure hash algorithm is used for hash value. Then the data is encrypted by using cryptography algorithm. Now the hash value and encrypted data must be hidden in image or audio or video file to secure the data. At the receiver end the hash value is matched and data is decrypted by using decryption technique.

**Keywords:** FPGA, hash function, Secure Hash Algorithm-1 (SHA-1), Verilog HDL.

## I. INTRODUCTION

Today the use of internet for communication has greater than before. So the security of information is significant issue for safety. Cryptography is a method of securing the information. For encrypting and decrypting the information, cryptography is useful. Encryption means convert the simple text into cipher text. The decryption means translate the cipher text into plain text. The encryption is completed at the sender side and decryption is completed at the receiver side. Cryptography is divided into asymmetric cryptography and symmetric cryptography. The symmetric key means same key is used at sender and receiver for encryption and decryption. The asymmetric key means dissimilar key is used at sender and receiver for encryption and decryption. Hash is a function of cryptography that constructs the hash value. The hash value is an arbitrary-length code that provides the reliability as well as confirmation. The hash value is a one-way function. Hash functions play a chief role in cryptographic application. SHA (Secure Hash Algorithm) is a legendary message compression standard used in computer cryptography, it can condense a long message into a short message. The SHA-1 Verilog source code is separated into three modules, namely Initial, Top and Round module. The Verilog code is assembled on Virtex5 FPGA using Xilinx ISE software tool. A comparison between desired SHA-1 hash function implementation in the company of supplementary works shows that it achieves a high output and clock frequency.

## II. SHA-1 HASHING ALGORITHM

The hashing function i.e. Secure Hash algorithm-1 is used to create the hashing value. It produces the hash value of 160 bits that is 20 bytes. It has the 80 number of rounds. The consumer which has the hash value can vary the information. The hashing algorithm provides accuracy and consistency. If any user modifies the data then the hash value will be distorted. SHA-1 is a complex algorithm that includes multiple 32-bit, 5-way additions, complex

logical functions, data shifting and a great amount of return. Normally implementations of the SHA-1 algorithm have essential large die areas and so made moderately exclusive portable devices. A proposed method has been useful to be relatively inexpensive one. The architecture is offered for SHA-1 hash function. The accomplishment is carried out by Verilog HDL on Xilinx FPGA device. The synthesis grades are compared and presented with other SHA-1 implementations.

Here, the hardware terms of system performance, operating frequency and covered area are compared. The hash algorithms, also called as message digest algorithms which generate a single fixed length bit vector for an arbitrary length message. The bit vector is called the hash of the message and it is noted as H. This hash value should be the same each time the same input is hashed. A hash function used in cryptography is one-way and collision resistant. The purpose of a hash function is to generate a fingerprint of a file, message or may be other block of data. Hash function must have the following requirements:

I.1. Weak collision resistance: For any given block x, it is infeasible to find y with

$$H(x) = H(y).$$

I.2. Strong collision resistance:- For several given block x, it is infeasible by computation to find x, y with

$$H(x) = H(y).$$

I.3. One-way property:- For any known value h, it is computationally infeasible to find x with  $H(x) = h$ . SHA (Secure Hash Algorithm) is intended by National Security Agency of the U.S.A. It is a message compression standard used to co-operate DSS (Digital Signature Standard) machinery. SHA is designed for DSS, it will be helpful in plenty of protocols and secure algorithms.

The basic version of SHA is called SHA or SHA -0. SHA-1 is the better version of SHA -0.

SHA-1

SHA1 is one of the most popular hash functions. The message block size for SHA-1 is 512 bits and message digest size is 160 bits. Calculation of message assimilates for one block message is finished in 80 rounds. The common properties of SHA-1 are summarize in Table.

SHA1	
Message Size	$<2^{64}$
Block Size	512 bits
Word Size	32 bits
Trans.Rounds	80
Message. Digest	160 bits
Security	80 bits
# of chaining variables	5

SHA-1 calculation is completed in 80 rounds and 5 hash variables each of 32 bits are used. The word size of all the calculations is 32 bits. The padded message is processed by 512 bit blocks. This 512 bit block is composed of 16 message words. These 16 message words are expanded by means of functions and in each of the total 80 rounds a new message word is used.

III. ONE WAY HASH FUNCTIONS

Hash function H satisfies the following requirements:

- I. H can be applied to block of data of any length.
- II. H generate a fixed-length output.
- III. Given H and x , it is simple to supercomputer message digest H(x).
- IV. Given H and H(x), it is computationally infeasible to find x.
- V. Given H and H(x), it is computationally infeasible to find x and x' such that H(x) = H(x').

All these three supplies are must for practical application of a hash function to message verification and digital signature. The fourth obligation also known as pre-image confrontation or one method property, states that it is simple to produce a message code given a message but firm (virtually impossible) to create a message known a code. The fifth necessity also known as second pre-image resistance property guarantee that an alternative message hashing to the equal code as a given message cannot be found.

IV. HASHING OPERATION

A hash function is a kind of operation that takes an input and produce a fixed-size filament which is called the hash value. The input string can be of some length depending on the algorithm used. The formed output is condensed representation of the input message or manuscript and usually called as a message digest, a digital fingerprint or a checksum. The size of the message digest is set depending

on the exacting algorithm. This means that for a particular algorithm, input stream give way an output of same length. In addition a very small change in the input results with a totally dissimilar hash value. This is known as the sudden large amount effect. The hashing operation is illustrate below in figure. The security of a hash function is straight related to the message digest length. Pre-image resistance, second pre-image resistance and collision resistance are extremely important uniqueness of any hash function.

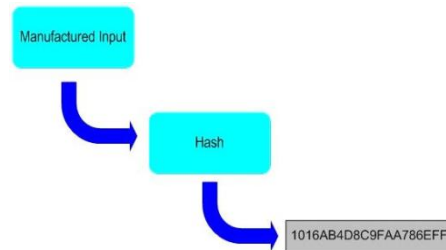


Fig. hash operation

V. HASH COMPUTATION

Each hash calculation process consists of two stages. The first stage is the preprocessing stage. In this stage the message is padded, parsed into n blocks and the chain variables are initialized. In the second stage, hash calculation is completed. In the hash calculation stage, constants, functions and word operations precise to the hash function are use. Hash calculation generate a message agenda from the padded message and uses that schedule, along with functions, constants and word operations to iteratively produce a sequence of hash values. The final hash value generate by the hash computation is used to generate the message digest. This situation is illustrate below in figure.

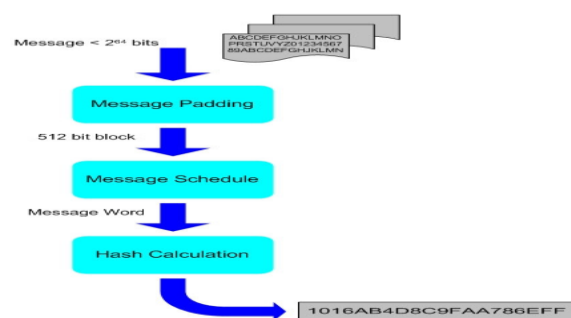


Fig. Computation Flow

VI. DATA HIDING

In cryptography data is hide by means of steganography. It is process of hiding information. The data can be hidden beside image, video and audio. For steganography, LSB replacement technique is use. The steganography is way to hide data in this that only sender and receiver can view the message. The data can be hide behind:

- I. Image steganography
- II. Audio steganography
- III. Video steganography

Image steganography :Least Significant Technique is use for image steganography. If the LSB is changed then that cause the small change to the original value. If the image is of 24 -bit , there is 3 byte of data to present RGB values for every pixel which shows that the 3 bits can be stored in every pixel.

Audio steganography: In case of audio steganography, the data will be hide at the back of audio file to hide the data behind audio is somewhat matches to image steganography. In audio steganography the data is hide behind samples. For samples, the sampling technique is follow.

### VI.Synthesis And Design of SHA-1

The input message of SHA-1 is no longer than 264bits can generate a 160 bit message abstract. The input is process in 512- bit blocks. The algorithm processing involve the following steps:

I.Padding: The aim of message padding is to make the total length of a padded message matching to 448 module 512. The number of padding bits is among 1 and 512. Padding consists of 1 single bit which follow a series of 0-bit.

II.Appending Length: A 64-bit binary account of the creative length of the message is appended to the conclusion of the message.

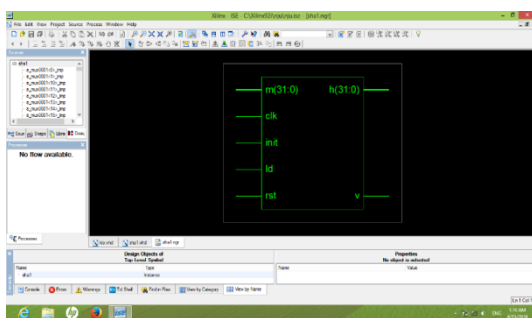
III. Initialize SHA- Buffer: The message digest is computed with the help of last padded message. The computation uses two buffers, in which each one contain of five 32-bit words and a succession of eighty 32-bit words.

IV. Hash Calculation

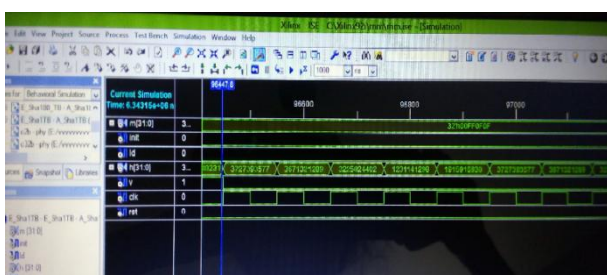
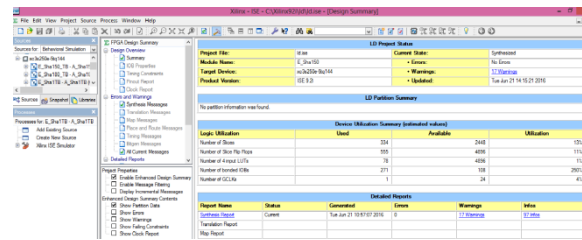
V.Output

### VII. SIMULATION RESULTS

#### I.RTL View of 32 bit SHA-1



#### II. Simulation Result of 32 bit sha1 algorithm

Logic Utilization	Used	Available	Utilization
Number of Slice	324	2460	13%
Number of Slice Flip Flops	268	4080	13%
Number of 4-input LUTs	16	4038	1%
Number of 8-input LUTs	2	18	1%
Number of DCMs	1	24	4%

Fig .Design Summary of 32 bit SHA-1

### VIII. CONCLUSION

Inside data communication, cryptography has their individual importance. Our research work surveys the existing hashing technique parallel to SHA-1 algorithms along with encryption and LSB substitution technique. The propose SHA-1 architecture has capability to achieve a superior working frequency and also higher throughput. Hash algorithm is use as mechanism by other cryptographic algorithms and process to provide information security services.

### REFERENCES

- International Journal of Emerging Technology and Advanced Engineering. ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 2, Issue 12, December 2012.
- Zhou Hua and Liu Qiao, "Hardware Design for SHA-1 Based on FPGA", IEEE International Conference Publications on Electronics, Communications and Control (ICECC), pp.2076-2078, 2011.
- Quynh Dang, "Recommendation for Applications Using Approved Hash Algorithm", NIST special publication 800-107, Computer security Division, National Institute of Standards and Technology, Dept of commerce, USA, pp. 1-21, 2011.
- Zhou Hua and Liu Qiao, "Hardware Design for SHA -1 Base d on FPGA",IEEE International Conference Publications OnElectronics, Communications and Control (ICECC), pp.2076-2078, 2011.
- Cheng Xiao-hui and Deng Jian-zhi, "Design of SHA-1 Algorithm based on FPGA", IEEE Second International Conference on Networks Security, Wireless Communications and Trusted Computing, (NSWCTC), Vol-1, pp. 532-534, 2010.
- A.G.Konheim and Ebooks Corporation., Computer Security and Cryptography. Hoboken:John Wiley & Sons Inc., 2007.
- Guopyin Wang, "An Efficient Implementation of SHA-1 Hash Function", IEEE International Conference on Electro or Information Technology, pp. 575-579, 2006.
- William Stallings, "Cryptography and Network Security, Principles and Practices" Fourth Edition , 2005 .