International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 12, December - 2013

FPGA Realization of Secured Hash Algorithm with Parallel Architecture

M. Rambabu[1], N. Srikanth[2], J. E. N. Abhilash[3]
Swarnandra College of Engineering and Technology

Abstract

Now-a-days security is the main problem to transmit the information from one place to another. Hackers tries to get the information which is to be secured. So these attacks will be a problems for us and also a challenge. So there is need to provide the security for the information between different systems or networks. Hence for this Secure Hash Algorithm is introduced. Secure Hash Algorithm is the most widely used Hash function in the world. This is one of the cryptographic algorithm which is mainly used for security based applications. This algorithm provides the best security i.e., a message digest of size n produces a collision with a work factor of approximately 2^n/2. The algorithm takes the arbitrary length message as plain text and produces a fixed length message digest (Hash code) as cipher text. The algorithm has several functional blocks like compression function, round calculation etc. The round calculations in the SHA constitute several processing steps. These steps will be processed in a sequential manner. Because of sequential processing the time taken to compute round calculation will be more. So in order to reduce the computation time required for round calculation, instead of using sequential process here parallel process mechanism is applied. This will reduce the computation time for round calculation and hence increased the speed of operation the algorithm. Finally we compare the computation time for this algorithm with the existing one. Here the algorithm concept is implementing on sporton3 FPGA kit by using Verilog HDL.

1. Introduction

Cryptography is the art of protecting information by transforming it (encrypting it) into an unreadable format, called cipher text. Only those who possess a secret key can decipher (or decrypt) the message into plain text. Where cryptanalysis is a study used to break Encrypted messages, also called code breaking, anyways modern cryptography techniques are virtually unbreakable. Cryptology is the study of secure communications, which contains both cryptology and cryptanalysis. Modern cryptography is heavily based on mathematical theory and computer science practice. Encryption was used to ensure secrecy in communications, such as those of military leaders, and diplomats. In recent decades, the field has expanded beyond confidentiality concerns to include techniques for message integrity checking, sender/receiver identity authentication, digital signatures, interactive proofs and secure computation, among others.

Cryptography Techniques:

The basic Techniques of cryptography are

A) Symmetric key

B) Asymmetric (public) key systems

C) Cryptographic hash functions.

The strength of a crypto system is directly related to the length of the key. This assumes that there is no inherent weakness in the algorithm and that the keys are chosen in a way that fully utilizes the key space (the number of possible keys).

A) Symmetric key:

Symmetric key algorithm uses the same key to encrypt and decrypt data. Some common symmetric key algorithms are the Data Encryption Standard (DES), Triple DES, Blowfish and the Advanced Encryption Standard (AES). DES is ineffective because it uses a 64-bit key and has been broken. the main advantage of symmetric key cryptography is speed. The main problems with this system are key distribution and scalability. Keys need to be distributed securely, and each secure channel needs a separate key. Symmetric key systems provide confidentiality but do not provide authenticity of the message, and the sender can deny having sent the message.

B) Asymmetric key:

Asymmetric (public) key algorithm uses a pair of mathematically related keys. Each key can be used to encrypt or decrypt. However, a key can only decrypt a message that has been encrypted by the related key. The key pair is called the public/private key pair. Some common public key systems are Rivest-Shamir-Adelman (RSA), Diffe-Hellman and Digital Signature Standard (DSS). Asymmetric key systems solve the key distribution and scalability problems associated with symmetric systems. Asymmetric key systems provide a greater range of security services than symmetric

systems. They provide for confidentiality, authenticity and non repudiation. The main problem with these systems is speed. It takes significantly more computer resources to encrypt and decrypt with asymmetric systems than symmetric ones.

C) Cryptographic hash functions:

Cryptographic hash functions take a message of arbitrary length and compute a fixed signature, often called a message digest, for the message. This can be done for a file, e-mail message or your entire hard-drive image. The main properties of these functions are that it is difficult to find different files that produce the same digest and that the function is one-way. Therefore, it is not computationally feasible to recover a message given its digest. two common examples of hash functions are the Secure Hash Algorithm (SHA), commonly SHA-1, and Message-Digest algorithm 5 (MD5). SHA-1 is used in many common security applications including SSL, TLS, S/MIME and IPSec. MD5 is generally used to create a digital fingerprint for verifying file integrity.

2. SHA Operation

The Secure Hash Algorithm (SHA) was developed by the National Institute of Standards and Technology (NIST) and published as a federal information processing standard (FIPS 180) in 1993; a revised version was issued as FIPS 180-1 in 1995 and is generally referred to as SHA-1.

In 2002, NIST produced a revised version of the standard, FIPS 180-2, that defined three new versions of SHA, with hash value lengths of 256, 384, and 512 bits, known as SHA-256, SHA-384, and SHA-512. These new versions have the same underlying structure and use the same types of modular arithmetic and logical binary operations as SHA-1, hence analyses should be similar. In 2005, NIST announced the intention to phase out approval of SHA-1 and move to a reliance on the other SHA. it is based on the hash function MD4 and its design closely models MD4. SHA-1 produces a hash value of 160 bits. In 2005, a research team described an attack in which two separate messages could be found that deliver the same SHA-1 hash using 2^69 operations, far fewer than the 2^80 operations previously thought needed to find a collision with an SHA-1 hash [WANG05]. This result should hasten the transition to newer, longer versions of SHA.

SHA is an one-way hash function algorithm that turns messages or text into a fixed string of digits, usually for security or data management purposes. The "one way" means that it's nearly impossible to derive the original text from the string. A one-way hash function is used to create digital signatures, which in turn identify and authenticate the sender and message of a digitally distributed message.

A) Hash Algorithm Structure:

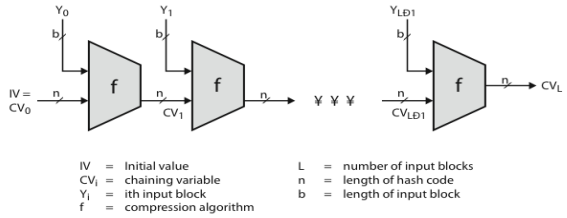

Fig 1: General Structure of Secure Hash Code.

SHA-1:

SHA-1 follows the structure depicted in following Figure.

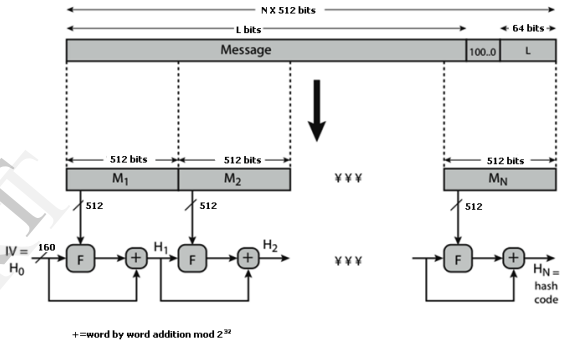

Fig 2: Message digest generation using SHA-1.

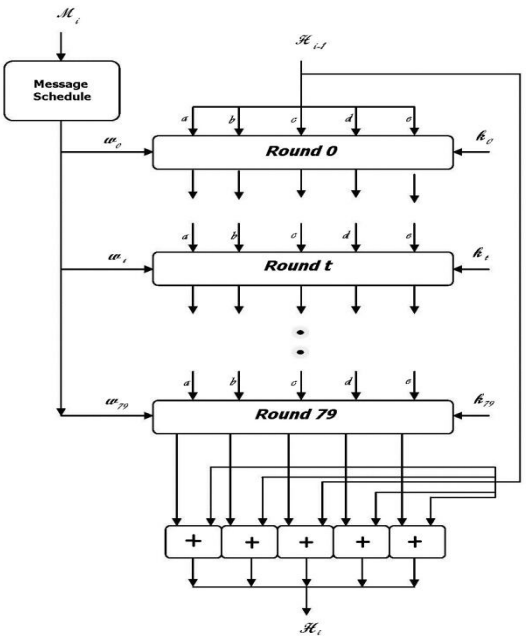

Fig 3: SHA-1processing of a single 512 - bit block.

The processing consists of the following steps:
• Step 1: Append padding bits
• Step 2: Append length
• Step 3: Initialize hash buffer
• Step 4: Process the message in 512-bit (32- word) blocks, which forms the heart of the algorithm
• Step 5: Output the final state value as the resulting hash

B) SHA-1 Compression Function:

The SHA-1 Compression Function is the heart of the algorithm. In this Step 4, it processes the message in 512-bit (32-word) blocks, using a module that consists of 80 rounds. Each round takes as input the 160-bit buffer value, and updates the contents of the buffer. Each round t makes use of a 32-bit value Wt derived using a message schedule from the current 512-bit block being processed. Each round also makes use of an additive constant Kt, based on the fractional parts of the cube roots of the first eighty prime numbers. The output of the eightieth round is added to the input to the first round to produce the final hash value for this message block, which forms the input to the next iteration of this compression function, as shown on the previous slide.

C) SHA-1 Round Function:

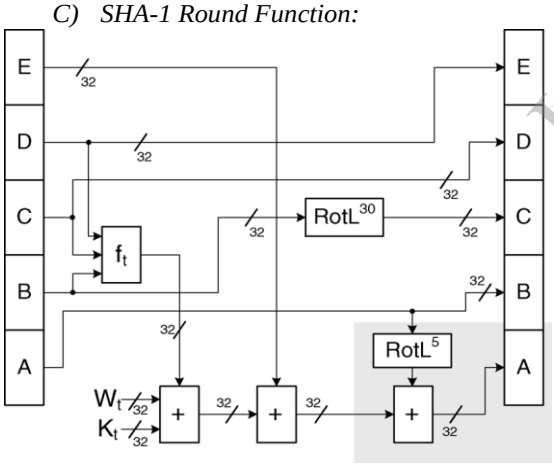

Fig 4: Elementary SHA-1 operations (single round).

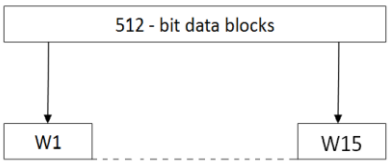

Fig 5:Derivation of Wt for t.[0,15]

The algorithm consists of 80 steps.
Let t denote the index of a step i.e, 0≤t≤79.
First a 32-bit message block Wt is derived for every step t from the 512-bit message block Mj using a message schedule.
For t<16, Wt is simply the ith 32-bit word of Mj
When t≥16,Wt are derived recursively with the following formula
Wt=(Wt-3+Wt-8+Wt-14+Wt-16)n1(s)
n denotes circular shift to the left by s bits &
+ is a logical xor operation
+ = addition modulo 2^32
Kt = a 32-bit additive constant
Wt = a 32-bit word derived from the current 512-bit input block.

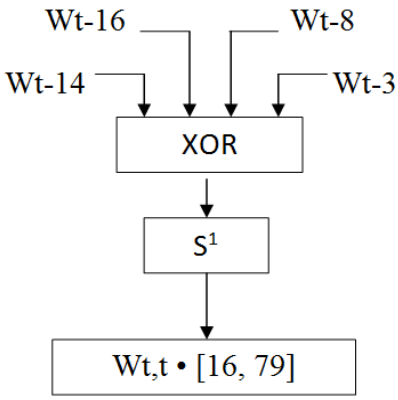

Fig 6: Derivation of Wt for t.[16,79]

The structure of each of the 80 rounds is shown in the above figure. Each 32-bit word shuffled along one place, and in some cases manipulated using a series of simple logical functions (ANDs, NOTs, ORs, XORs, Rotates), in order to provide the avalanche & completeness properties of the hash function. The elements are:

3. High Speed Architecture

SHA is a one way hash function. This type of one way hash functions mainly used in digital signatures, Authentication. Now a days in many applications we are adding passwords to secure our data. Now a days hardware security using a password lock also widely used. In all these cases authentication is required. The password which we are providing while unlocking must be matched with the predefined password. If the password is encrypted and stored as a message digest, The password which we are supplying at the time of

unlocking also be encrypted with the same algorithm and the digest produced must be matched with the previous digest. In order to speed up the authentication here is an architecture for Secured Hash Algorithms. To get the output message digest from the Secure Hash Algorithm, we need to follow some processing steps like splitting the available arbitrary message in 512-bit block, append padding bits, length bits and initialize the Hash Buffer Value etc..,. In the SHA-1 compression function, totally there are 80 round calculations. In each round calculation, there are several operations like shift, OR, AND, exclusive-OR etc. So far we follow sequential process for updating the Hash Buffer Registers & for computing the round calculations in the SHA round calculation process. Now in the High Speed Architecture, instead of using sequential process we use parallel process mechanism by posting the initial buffer values in to the additionally used buffer registers. By employing parallel process mechanism, the computation time required for round calculation will be extensively decreased, and hence the speed of operation of the SHA will be greatly improved. This algorithm may be very useful in the upcoming feature whenever High Speed operation required.SHA is a one way hash function. This type of one way hash functions mainly used in digital signatures, Authentication. Now a days in many applications we are adding passwords to secure our data. Now a days hardware security using a password lock also widely used. In all these cases authentication is required. The password which we are providing while unlocking must be matched with the predefined password. If the password is encrypted and stored as a message digest, The password which we are supplying at the time of unlocking also be encrypted with the same algorithm and the digest produced must be matched with the previous digest. In order to speed up the authentication here is an architecture for Secured Hash Algorithms. To get the output message digest from the Secure Hash Algorithm, we need to follow some processing steps like splitting the available arbitrary message in 512-bit block, append padding bits, length bits and initialize the Hash Buffer Value etc..,. In the SHA-1 compression function, totally there are 80 round calculations. In each round calculation, there are several operations like shift, OR, AND, exclusive-OR etc. So far we follow sequential process for updating the Hash Buffer Registers & for computing the round calculations in the SHA round calculation process. Now in the High Speed Architecture, instead of using sequential process we use parallel process mechanism by posting the initial buffer values in to the additionally used buffer registers. By employing parallel process mechanism, the computation time required for round calculation will be extensively decreased, and hence the speed of operation of the SHA will be greatly improved. This algorithm may be very useful in the upcoming feature whenever High Speed operation required.

4. HDL Implementation

For these modifications are specially designed to counter the above mentioned problems. In the implementation of Modified Secured Hash Algorithm Scheme complete module is described using Verilog HDL. In order to speed up the algorithm execution parallel architectures are used for different round functions. Because of the parallel architecture all the round functions execute simultaneously. There by the Secured Hash Algorithm execute very fast.

5. Simulation Results

Overall architecture is described using VHDL language and is simulated using Xilinx ISE simulation environment. Original plain text that is to be encrypted is supplied to the encryptor block using Datain input port. That input data is encrypted using SHA algorithm by using some secured Keys. After Eighty rounds of execution message digest produced. Simulation results shown in figure 10 are for the messeage APPLE. After encrypting the string APPLE the message digest "254698546" is generated. Foe a very small change in the original message it is observed a great change in the message digest. Instead of APPLE if APPLF is supplies as original text corresponding message digest generated is "15680145".

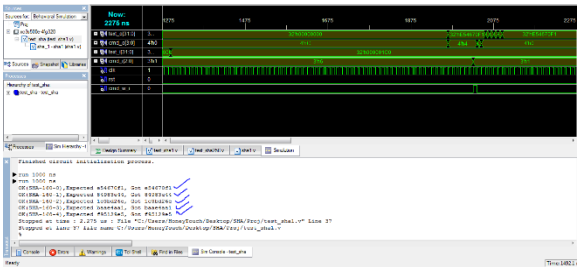

Fig 7: Simulation results showing authenticated.

Complete system is described with Verilog HDL. Xilinx ISE project navigator is used as HDL editor and ISE simulator is used for simulation. Figure 7 shows the ISE simulation results. When a test input is applied for the design the test bench it was checking for a message digest. When the digest is matched with the expected then it is said to be authenticated.

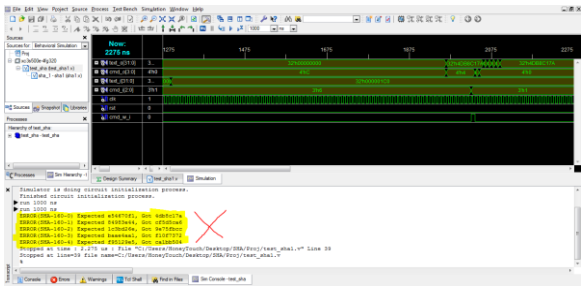

Fig 8: Simulation results showing authentication failed.

Figure 8 shows the Xilinx ISE simulation results when the test input is different from the original input. Only first character is changed by only one bit. But the message digest is completely different from the expected. In Figure 8 the error log is highlighted, showing message digest is completely different from the expected.

6. Conclusion:

In this Secured Hash Algorithm, for a very small change in the original message a great change in the message digest is observed. Simulation results shows even a one bit change in the original message input there was be a great change in the message digest is observed. Because of the high speed architecture the message digest is generated within a very short time. Because of the very fast message digest generation the authentication process also completes in a fast manner. This high speed architecture can be applied to email authentication, Hardware locks and also for digital signatures. By including pipelining technique this encryption process speed further improved.

References

[1] Whitfield Diffie and Martin Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory, vol. IT-22, Nov. 1976, pp: 644–654
[2] FIPS PUB 197: The official Advanced Encryption Standard.
[3] NCUA letter to credit unions, July 2004
[4] RFC 2440 - Open PGP Message Format
[5] SSH at windowsecurity.com by Pawel Golen, July 2004
[6] Bruce Schneier,Applied Cryptography,
2nd edition, Wiley, 1996, ISBN 0-471- 11709-9.
[7] AJ Menezes, PC van Oorschot, and SA Vanstone, Handbook of Applied Cryptography ISBN 0-8493-8523-7.
[8] Walter Tuchman (1997). "A brief history of the data encryption standard". Internet besieged: countering cyberspace scofflaws. ACM Press/Addison-Wesley Publishing Co. New York, NY, USA. pp. 275–280.
[9] National Institute of Standards and Technology, NIST Special Publication 800-67 Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, Version 1.1
[10] American National Standards Institute, ANSI X3.92-1981 American National Standard, Data Encryption Algorithm
[11] "ISO/IEC 18033-3:2010 Information technology — Security techniques — Encryption algorithms — Part 3: Block ciphers". Iso.org. 2010-12-14.
[12] Bruce Schneier, Applied Cryptography, Protocols, Algorithms, and Source Code in C, Second edition, John Wiley and Sons, New York (1996) p. 267
[13] William E. Burr, "Data Encryption Standard", in NIST's anthology "A Century of Excellence in Measurements, Standards, and Technology: A Chronicle of Selected NBS/NIST Publications, 1901–2000.
[14] V. Klima, ―Finding MD5 collisions—A toy for a notebook.‖ Cryptology ePrint Archive, 2005/075, 2005.
[15] National Institute of Standards and Technology (NIST), MD, ―FIPS 180–2, secure hash standard (SHS),‖ 2002.
[16] HELION, Cambridge, U.K., ―Fast SHA-2 (256) hash core for Xilinx FPGA, 2005.