International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

FPGA Implementation of MD5 Algorithm for Password Storage

Shadab Ahmad Khan

Department of ECE (M.Tech VLSI TECHNOLOGY), School of Engineering and Technology, Sharda University, Greater Noida, India

Abstract: ***In this paper, the description of a hardware based MD5 core, which is designed and implemented using the hardware description language Verilog, is given. Hash functions are very common in the modern day world as a means of communication integrity and signature authentication. These functions produce a fixed-size fingerprint or hash value for a variable length (very long) message. MD5 represents one efficient algorithm for hashing the data, then, the purpose of implementation and used this algorithm is to give them some privacy in the application. The hash function MD5, Message digest Hash Algorithm, is examined in order to find the common constructs that can be used to implement it using hardware blocks of the FPGA.***

Keywords: Hash functions, authentication, fingerprint, MD5, algorithm, cryptography.

1. Introduction

Due to the rapid developments in the wireless communications area and personal communications systems, providing information security has become a more and more important subject. This security concept becomes a more complicated subject when next-generation system requirements and real-time computation speed are considered. In order to solve these security problems, lots of research and development activities are carried out and cryptography has been a very important part of any communication system in the recent years. Cryptographic algorithms fulfill specific information security requirements such as data integrity, confidentiality and data origin authentication. Hash functions are among the most important cryptographic algorithms and used in the several fields of communication integrity and signature authentication. These functions are sort of operations that take an arbitrary length of input and produce a condensed representation of that input. This condensed representation of an arbitrary long input is usually referred as message digest or hash value. The size of the message digest is fixed depending on the particular hash function being used. The security of a hash function is directly related to this message digest length. Hash functions have some specific properties that make them secure; these properties are pre-image resistance, second pre-image resistance and collision resistance as indicated in the documents of FIPS (Federal Information Processing Standards).

2. Hash Function

A hash function is a sort of operation that takes an input and produces a fixed-size string which is called the hash value. The input string can be of any length depending on the algorithm used. The produced output is a condensed representation of the input message or document and usually called as a message digest, a digital fingerprint or a checksum. The size of the message digest is fixed depending on the particular algorithm being used. This means that for a particular algorithm, all input streams yield an output of same length. Furthermore a very small change in the input results with a completely different hash value. This is known as the avalanche effect. The hashing operation is illustrated below in Figure 1:

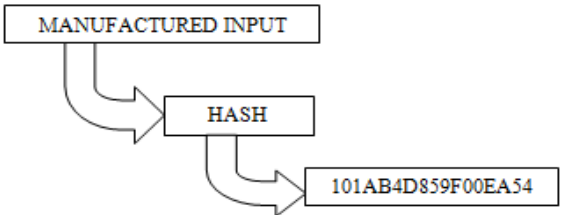


Figure 1: Hashing operation

3. Hashing of Password

The most common use fields of hash functions are verifying data integrity, providing password authentication and generating digital signatures with DSA in applications such as electronic mail, electronic funds transfer, software distribution and data storage which require data integrity assurance and data origin authentication. Password authentication is another field that hash functions are used. For computer systems, it is insecure to store passwords in clear-text. Someone may reach all of the passwords and entire user password database can be compromised. Because of these reasons, a more secure way is to store the hashes of the passwords rather than clear text passwords. Storing the hashes of passwords is shown below in Figure 2:

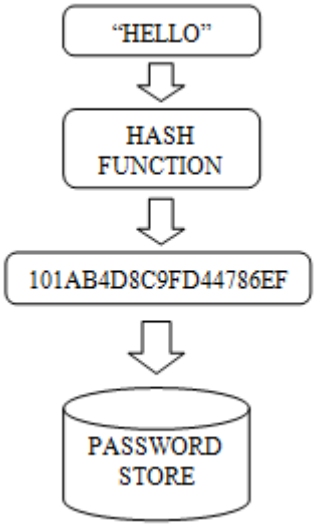


Figure 2: Storing the hash of a password

When a user logs in, the hash value of the submitted password is calculated and compared with the one stored in the password database. If the calculated hash value is identical to the one stored in the database, the user is authenticated, and otherwise the user is not granted. By this way, even if the password database is compromised, user privacy is still protected since it is computationally very difficult to obtain the original passwords from the hash values.

4. Hashing of Digital Signature

One of the most popular applications of hash functions is digital signatures. A digital signature is a type of asymmetric cryptography used to simulate the security properties of a signature in digital, rather than in written form. Digital signatures are used to provide authentication of the associated input, usually called a message. Messages can be anything from electronic mail to someone or even a message sent in a more complicated cryptographic protocol. The applications of a digital signature are illustrated below in Figure 3:

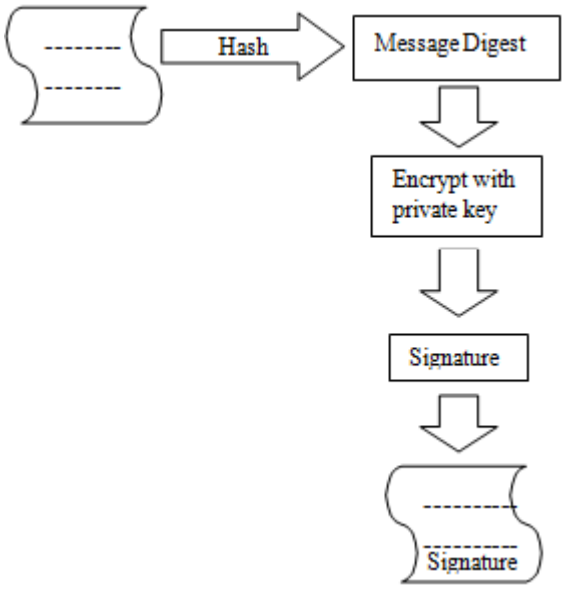


Figure 3: Application of digital signature

5. Known Hash Functions

There is several hash functions developed up to now and among these hash functions MD5, SHA-1, and SHA-256 are most popular. Summary of the standard hash functions is given below in Table 1.

Algorithm	*Output size*	*Block Size*	*Word Size*	*Rounds xSteps*	*Year of the standard*
MD4	128	512	32	16x3	1990
MD5	128	512	32	16x4	1991
RIPEMD	128	512	32	16x3 (x2 parallel)	1992
RIPEMD-128	128	512	32	16x4 (x2 parallel)	1996
RIPEMD-160	160	512	32	16x5 (x2 parallel)	1996
SHA-0	160	512	32	80	1993
SHA-1	160	512	32	80	1995
SHA-256	256	512	32	64	2002
SHA-224	224	512	32	64	2004
SHA-384	384	1024	64	80	2002
SHA-512	512	1024	64	80	2002

6. The MD5 Hash Algorithm

The MD5 (1992) message-digest algorithm was designed as a strengthened extension of the MD4 (1990) message digest algorithm. MD5 is slightly slower than MD4, this is a classical example where security is favoured at the expense of speed. Both algorithms were developed by Ron Rivest who is the "R" in the RSA [Rivest-Shamir-Adleman] public-key encryption algorithm.

7. The MD5 Description

The algorithm accepts an input message of arbitrary length and produces a 128-bit "message digest", "fingerprint" or "hash result". Figure 5 depicts the way the input message is turned into a 128-bit message digest.

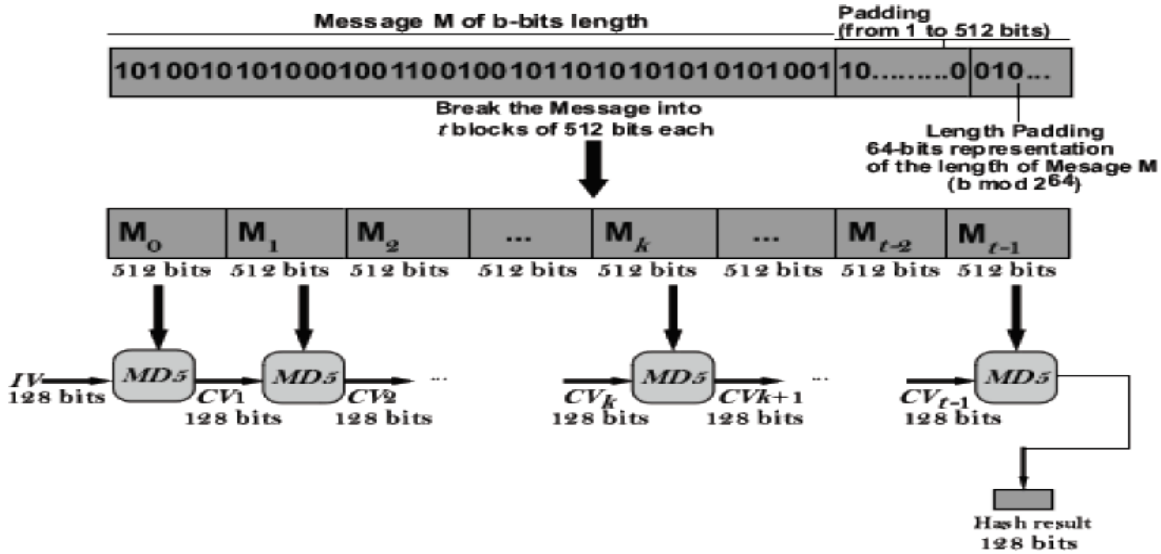


Figure 5: MD5 Algorithm

8. Simulation Results

The proposed MD5 algorithm has been implemented and designed through Verilog (ModelSim) and XILINX ISE ISIM Simulator. The output results have been shown below in Figure 6 . The result for input "Hello world!" and "a" are shown in Figure 6.

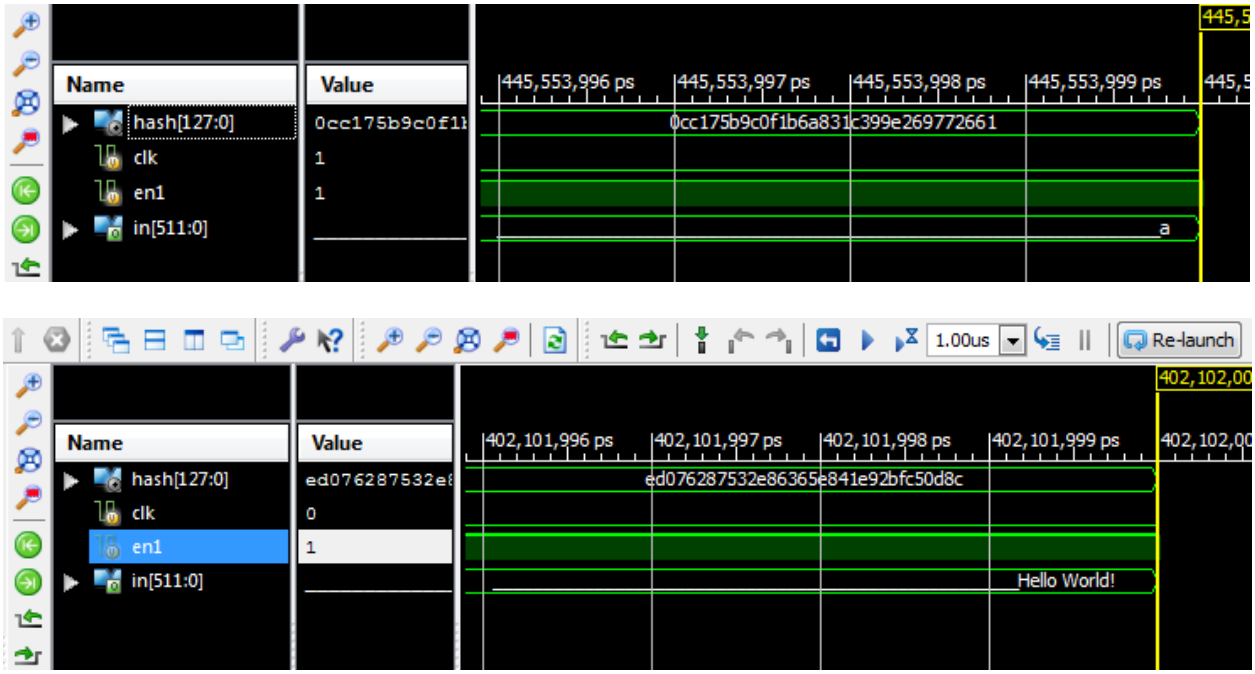
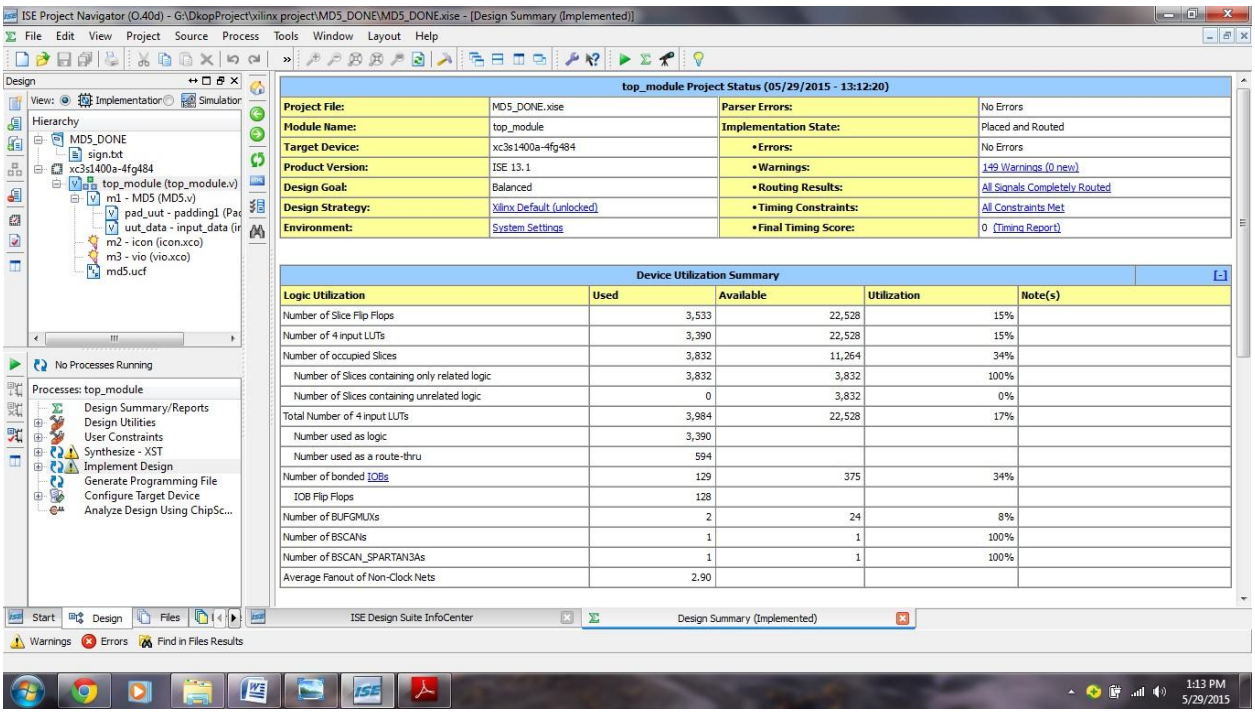


Figure 6: Simulation Results

Synthesis Results

Here using the Xilinx XST tool for MD5 for XC3S1400A following synthesis result has been generated which is shown in following figure:



9. Conclusion

Hash function implementations on hardware seem to be more popular as the developments in the communications area continue tremendously. Implementing hash functions on hardware is preferred since software implementations don't satisfy the speed, throughput and security requirements of the complex communication systems in use today. Hash function implementations are used in several fields of information security such as providing password

authentication, verifying data integrity and generating digital signatures for both data origin authentication and verifying the content of the document. The hash processor proposed in this study can be used in these applications easily. The usage of the processor is flexible, since it has a serial communication interface that makes the communication with the external world possible.

References

[1] Anh Tuan Hoang, Katsuhiro Yamazaki and Shigeru Oyanagi , ,Multi-stage Pipelining MD5 Implementations on FPGA with Data Forwarding, 16th International Symposium on Field-Programmable Custom Computing Machines 2008.
[2] Changxin Li, Hongwei W, Shifeng Chen1, Xiaochao Li2 , Donghui Guo, ,Efficient Implementation for MD5-RC4 Encryption Using GPU with CUDA, 3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication,. ASID 2009.
[3] Chiu-Wah Ng, Tung-Sang Ng and Kun-Wah Yip ,A UNIFIED ARCHITECTURE OF MD5 AND RIPEMD-160 HASH ALGORITHMS, ISCAS 2004
[4] Dongjing He and Zhi Xue Multi-parallel Architecture for MD5 Implementations on FPGA with Gigabit-level Throughput, International Symposium on Intelligence Information Processing and Trusted Computing, 2010.
[5] Feng Wang, Canqun Yang, Qiang Wu, Zhicai Shi, Constant Memory Optimizations in MD5 Crypt Cracking Algorithm on GPU-Accelerated Supercomputer Using CUDA The 7th International Conference on Computer Science & Education (ICCSE 2012). Melbourne, Australia 2012.
[6] H. Mirvaziri, Kasmiran Jumari, Mahamod Ismail, Z. Mohd Hanapi, Anew Hash Function Based on Combination of Existing Digest Algorithms , The 5th Student Conference on Research and Development – SCOReD 2007, 11-12 December 2007, Malaysia
[7] J. Touch, Report on MD5 Performance, RFC 1810, June 1995.
[8] Kimmo J, rvinen, Matti Tommiska and Jorma Skytt ,Hardware Implementation Analysis of the MD5 Hash Algorithm , Proceedings of the 38th Hawaii International Conference on System Sciences – 2005
[9] Kostas Theoharoulis, Ioannis Papaefstathiou, Charalampos Manifavas, Implementing Rainbow Tables in High-end FPGAs for Superfast Password Cracking, International Conference on Field Programmable Logic and Applications 2010.
[10] Md. Didarul Alam Chawdhury, and A.H.M. Ashfak Habib, Security Enhancement of MD5 Hashed Passwords by using the Unused Bits of TCP Header, Proceedings of 11th International Conference on Computer and Information Technology (ICCIT 2008) 25-27 December, 2008, Khulna, Bangladesh.