Cellular Automata based SHA-512 Hashing Algorithm

J. Suganya
ECE, SSN College of Engineering
Chennai, India

K. J. Jegadish Kumar
ECE, SSN College of Engineering
Chennai, India

***Abstract* - Hashing algorithm maps random sized data to fixed size. Hash functions are globally used for secured communications system like digital signature, message integrity. In this paper, hashing algorithm is implemented using cellular automata. SHA 512 from the hashing family is chosen for this implementation. The mixer function is implemented and the performances like delays and Input / Output Bond requirement are focussed in this work.**

Keywords-Cellular Automata, Hashing algorithm, SHA 512

I. INTRODUCTION

Cryptographic hash function has an eminent role in recent cryptographic scenario. The hash value act as compact representative image of an input [1]**.** Mapping up of arbitrary size data to fixed size data is called as hashing function. SHA 512 is one among the hashing function. Even though hash function has many SHA algorithms like SHA-1,SHA-2,…,etc SHA 512 has more complexity with 512 bits. Implementation of SHA 512 follows by breaking up of larger modules into smaller, to ease the implementation task in Verilog description language. SHA 512 consists of 8 words of 64 bits each. Implementation of HCA in tiny chip area and in small architecture will be challenging task. These are the drawbacks that are to be considered. Each algorithm will be differentiated by delays and usage of input output blocks. This will be the major criteria in implementation of SHA 512 using cellular automata. Certain premises of boolean function are used for determining the modern attacks in hash function. These underlying elements are determined by using the CA rules as boolean function [2]**.** Initially SHA 512 will be implemented as it is, then it will be concentrated along with cellular automata.

Hash functions has their applications towards check sums, check digits, finger prints, error-correcting codes, and ciphers. Each concept has its own uses and specifications and the concepts were performed and rectified differently. In order to enhance the quality of cryptographic standard avalanche and collision test is used. Generation of new hash function using cellular automata deals with some of the applications like message integrity, password verification. For the password verification, digital signature is the best example. Rule 30 produces pseudorandom numbers and it is not reversible. These are the reasons behind for choosing rule 30 among the hybrid rules. Strict avalanche criterion and collision test are the two security tests for rule30. If any one bit is changed in the input half of the output should be changed known as SAC [2]**.** But SAC is not applicable in small iteration.

SAC which is desired is applicable only in larger iteration. High level parallelism, high speed and contrary against timing analysis attack are the reasons for embarking Cellular automata [5]**.** Cryptography is a basic requirement in this era of global electronic connectivity to secure data storage and transmission against the threats such as message eavesdropping and electronic fraud. The effectiveness is defined as how long it takes to encode and decode messages without breaking the 'key' using CA rules. CA based encryption provides good immunity due to Boolean functions are satisfied by the cellular automata rules.

II. CELLULAR AUTOMATA

A Cellular Automaton is a distinct time and space dynamic system consisting cells arranged in an array, each of which implements a simple automaton. Arrangement of these cells will be in one or two proportions and can be connected to their neighbours in various combinations. The cells use their state and the neighbour's state to obtain the succeeding state of the transition. Cellular Automata is a computational approach which implements complicated computational blocks into plain modular logic blocks. The parallel CA model is appropiated for the implementation on reconfigurable hardware architectures like FPGA that can provide a significant speedup. The application of CA in cryptography is that the generation of random number has been performed by using one and two dimensional CA. It is proposed for public key cryptosystem. The rule is given, it is simple to find the next state but it is very immense to calculate the previous state. However, the designer can design a rule in such a way as to be able to reverse it easily.

The quality of the cryptographic hash function that has been generated newly is improved by standard avalanche and by collision test [8]. Cellular automat has been initially determined for public key generation. Though one way function is the progression of a definite CA its inverse is considered to be hard to find. The future state can be depicted easily by the given rule, but it appears to be very difficult to calculate previous states. The cellular Automata is utilized in Error correction coding, cryptography, computer processors.

III. SHA 512

SHA was published in the year of 1993 by NIST. Then it was reviewed in the year 1995 in order to remove some of its weakness. And as a result a new hashing algorithm has

been obtained called SHA-2, which uses larger message digest. These message digest were made more resistant to feasible attacks and they were allowed to use the blocks with larger data size. This is applicable to SHA 512.

Fig.1 represents the SHA 512 compression function. SHA 512 consists of 8 words of 64 bits each. The standard hashing algorithm will run individual round for 80 times and the generates 512 bits output. This 512 bits will act as input to the following message block. After processing of 1024 bit message blocks the, last 512 bits of message digest will be the message signature. The looping architecture with 80 iterations would seem to provide the most area efficient solution.

When the first round is about to end , Register1 clinches Hi of the previous round (Hi-1). Using multiplexer each round is repeated 80 times. And the output feedback is returned back to round block. The output that is obtained from each round is stored in registers, to prepare feedback value for the next round.

IV. IMPLEMENTATION OF CA

The following Fig.2 describes about the function of single round. The output ABCEFG has been given respectively to BCDFGH. But for A and E special Boolean function is performed which deals with AND, OR and ROTATE function.

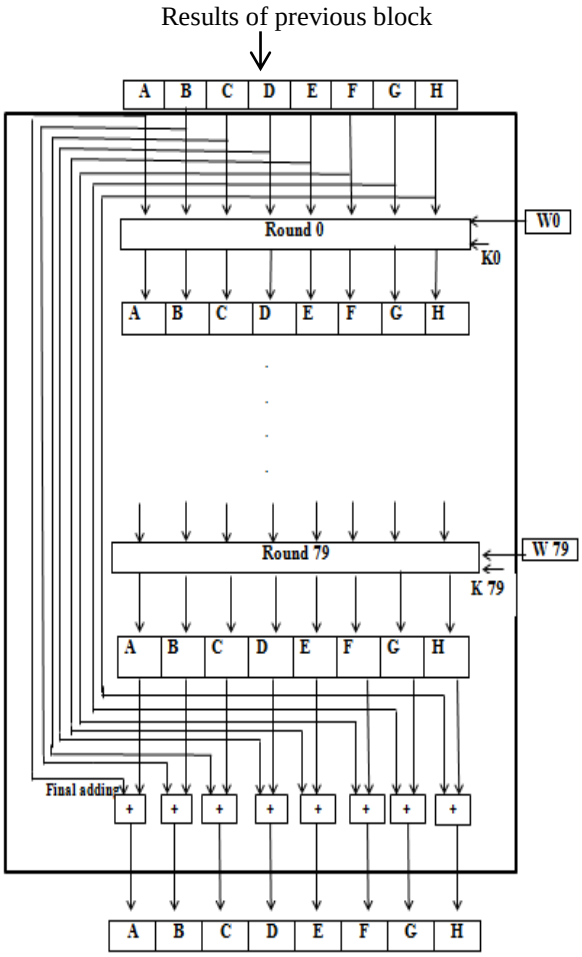

Fig.1 *SHA 512 Compression function*

This paper focussed with the implementation of mixer part in SHA 512. The implementation of conventional part of mixer consumes more IOB's and sliced LUT's. The delay is also more which indicates consumption of more time. The area also seems to be more due to the requirement of IOB's and LUT's. Hence different method has been adopted to overcome this problem. With the help of cellular automata rules these problems were rectified.

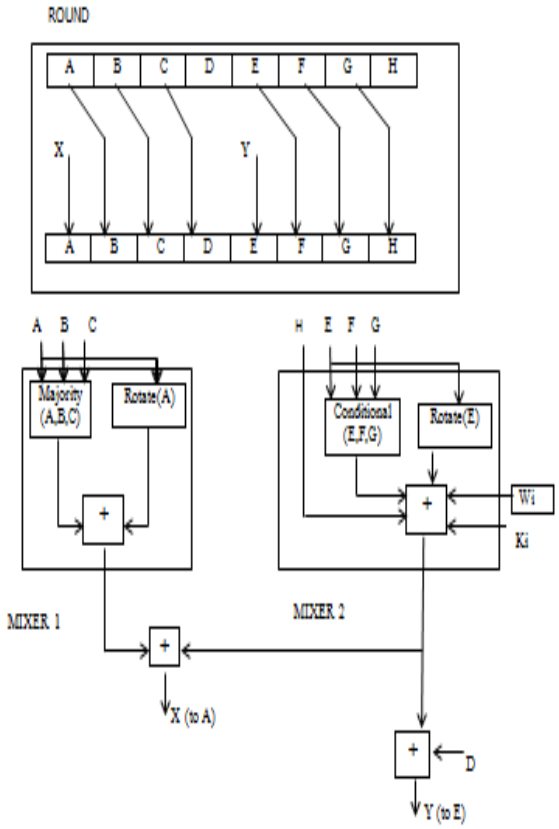

Fig.2 *Structure of Each Round in SHA 512*

By taking xc7v2000t as the targeting device the objective has been achieved. Mixer 1 deals with Majority and Rotate function.

Where,

Majority $(a, b, c) = (a \text{ AND } b) \oplus (b \text{ AND } c) \oplus (a \text{ AND } c)$

Rotate $(a) = (\text{Rot } R_{28}(a)) \oplus (\text{Rot } R_{34}(a)) \oplus (\text{Rot } R_{39}(a))$

Conditional $(a,b,c) = (a \text{ AND } b) \oplus ((\text{NOT } a) \text{ AND } c)$

$+$ = *Addition modulo 2^64*

Rot $R_i(a)$ = *Right rotation of the argument a by i bits*

The output of Rotate, Majority function is modulated and the mixer 1 output is obtained in simple manner. Mixer 2 deals with Conditional, Rotate function and with key words with 64 bits along with modulation. In the new approach the modulo adder in the mixer part is replaced with Cellular automata rules. Different set of rules has been used to monitor the delays and IOB requirements.

V. RESULTS & DISCUSSION

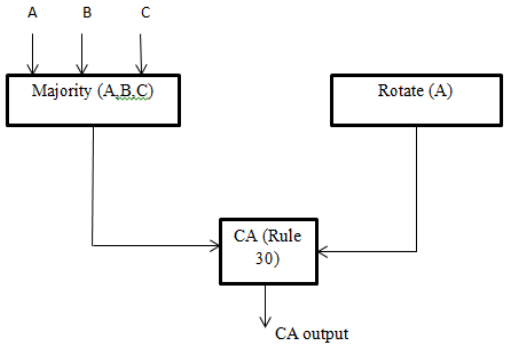

Fig.3 *Mixer 1 using CA rule 30*

Table.1 Comparison table of mixer1

Device – xc7v2000t

Parameters	Conventional Modulo Addition	CA Rule 30
Delay	4.542 (ns)	1.007 (ns)
No.of slice LUT's	319	128
No.of bonded IOB's	576	384

By replacing the modulo added with Cellular Automata rules in mixer 1 the delay, IOB requirements have been reduced to considerable range. The Fig.3 represents the implementation of CA in mixer 1 of SHA 512.

Mixer part of SHA 512 has been compared with rules like Rule 30, Rule 126, Rule 134, Rule 166, and Rule 90. Since rule 30 offers more cryptographic properties like non linearity, balanced and it also provides reduction in delay, IOB requirement, sliced LUT's. Rule 30 has been chosen for this new approach. The comparable readings were tabulated in Table 1.

Fig.4 represents the replacement of modulo adder with Cellular Automata rule 30. The same procedure is repeated for mixer 2 and the readings were tabulated.

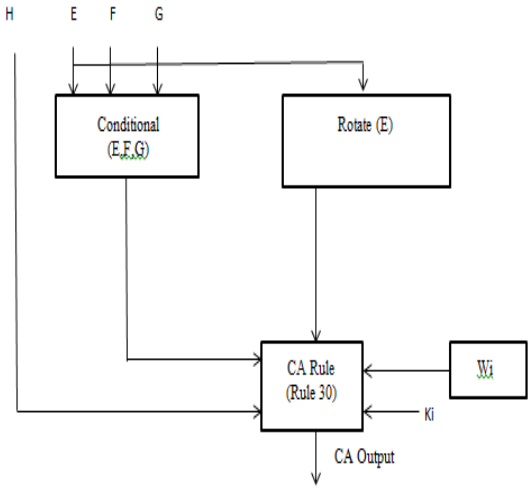

Fig.4 *Mixer 2 using CA rule 30*

Table.2 Comparison table of Mixer 2

Device – xc7v2000t

Parameters	Conventional Modulo Addition	CA Rule 30
Delay	8.328 (ns)	5.074 (ns)
No.of slice LUT's	829	638
No.of bonded IOB's	1152	1024

When comparing the cellular automata rule with conventional type of SHA 512 the delay has been reduced and the requirement of IOB's also reduced for mixer 2. The corresponding readings were tabulated in Table.2

IV. CONCLUSION

In this work, the mixer part of SHA 512 has been implemented using Cellular Automata. Thus by choosing the best among the CA rules mixer is implemented. Based on the algorithm, initially mixer part of SHA 512 is implemented using CA Rule and synthesized in device xc7v2000t using Xilinx 14.7. Thus by comparing the results in terms of delay, slice LUT's, bonded IOB's are all reduced to certain extent. Further, the efficient implementation of SHA 512 with and without Cellular Automata will be the future scope.

V. REFERENCES

[1] Jun-Cheol Jeon(2016),'One-Way Hash Function Based On Cellular Automata', Volume: 15, Pages: 336 – 349.
[2] Norziana Jamil , Ramlan Mahmood , Muhammad Reza Z'aba (2012) 'A new cryptographic hash function based on cellular automata rule 30,134 & omega flip network' International Conference on Information and Computer Networks, vol. 27.
[3] Enrico Zimuel (2013), 'Anew Cryptographic Hash Function Based On The Cellular Automata Rule 30', IEEE Journals & Magazines,Volume: 15.
[4] Gitika Maity & Jaydeb Bhaumik (2015) 'New HCA rules sets for cryptographic design' Computer communication control & information technology (C3IT), 2015 Third international conference.
[5] Nalini C. Iyer and Sagarika Mandal (2013) 'Implementation of Secure Hash Algorithm-1 using FPGA' International Journal of Information and Computation Technology. ISSN 0974-2239 Volume 3, Number 8 pp. 757-764.
[6] Michail, H., Kakarountas, A., Milidonis, A. and Goutis, C., (2009) 'A top-down design methodology for ultrahigh-performance hashing cores'IEEE Transactions on Dependable and Secure Computing, Vol.3, p.255.
[7] Hong-qiang Li, Chang –yun Miao (2006),' Implementation of Hash function 512' First informational conference on Innovative Computing, Information and control, Volume 2, Pages: 38-42.
[8] Nalini C. Iyer and Sagarika Mandal (2013), 'Implementation of Secure Hash Algorithm-1 using FPGA' International Journal of Information and Computation Technology. ISSN 0974-2239 Volume 3, pp. 757-764.
[9] N. Skalovas, C. Efstathiou (2005), 'On The FPGA Implementation Of HAVAL Hash Function' EUROCON 2005-The international Conference on "computer as a tool, Volume: 1, Pages: 709 – 712.