iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 7 **Issue:** IV **Month of publication:** April 2019

DOI: https://doi.org/10.22214/ijraset.2019.4570

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 7 Issue IV, Apr 2019- Available at www.ijraset.com

Implementation of SHA-3 in FPGA using Round Pipelined Technique

M. Swetha Reddy[1], Mr. G. Ravi Kumar[2]
[1, 2]*Department of Electronics and Communication Engineering, Vidya Jyothi Institute of Technology*

Abstract: ***Secured Hashing Algarithum is used to ensure the integrity and authenticity of data to achieve higher level security. These approaches are considered for FPGA based implementation of SHA-3 hash function. .while the performance of folded structure just matches the state of art. The design is logically optimized for area efficiency by merging Theta, Rho and pi step, Chi step, and Iota step of algorithm into structures which are piplined slice-wise allow to further improve of existing state of art. By solving the intra-round dependencies caused by the Rho step mapping. It is possible to reduce area resources and obtain shorter data path and the fully iterative and round piplined technique based architectures are composed in terms of frequency, throughput per slice improvement and area. In this paper we are implementing SHA3 1024 variant using Xilinx 14.7***
Keywords: theta, rho, pi, chi, iota.

I. INTRODUCTION

The SHA-2 family of cryptography hash functions was designed in 2001 by United States NSA. A remarkable problem with SHA-1 and SHA-2 that both use the same appliance, called Merkle-Damgard, to the process message text. That mean successful attack on SHA-1 becomes a potential risk on SHA-2.Consider SHA-1 for instance. A brute force attack usually takes at least 280 rounds (a round is a single cycle of alteration of the interim hash value) to find a collision in a full-round SHA-1.But the, Xiaoyun Wang and colleagues used a differential path attack to break a full-round SHA-1, and that took only 269 cycles to complete. That same attack was later substantiated by Martin Cochran in August 2008. this attack produced a near-collision later 258.5 cycles. He also estimated a changed attack can manage a full-collision after 261 cycles. the only successful attacks were those against a limited round SHA-2 hash. The most in effect attack was against a 46-round SHA-2 and against a 41-round SHA-2 (256-bit variant). It took 2253.6 cycles to break the 256-bit variant and 2511.5 cycles for the 512-bit variant. The fact remains that, while no successful attacks against a full-round SHA-2 have been announced, there is no doubt that attack appliances are being developed in private.

II. SHA-3

SHA-3 standard, candidate hash functions had to meet four conditions set by NIST. If a candidate failed to meet this conditions, it was prohibited: The candidate hash function had to perform well regardless of implementation. Many candidates actually unable to meet these requirements. The candidate function to be conservative about security. It should be with stand known attacks, while sustaining a large safety factor. It should produce the same four hash sizes as SHA-2 but able to supply longer hash sizes if need be. Any weaknesses were found during they were analyzed and addressed through tweaks or through redesign. The SHA-3 competition contains 51 candidates function entering the first round for evaluations. Out of those, 14 managed to advance to the second round. Round three contain the candidates whittled down to five. And from those five, SHA-3 was declared the winner. SHA-3 is recognized as a new Secure Hash Algorithm-3 i.e. SHA-3 [3] announced by NIST. Gilles Van Assche, Guido Bertoni, Michael Peeters Keccak is generated from sponge function with Keccak [r, c] members. It is categorized by these additional functions i.e bit rate (r) and capacity (c). The 1600-bit state matrix of Keccak composed of 5x5 matrixes of 64-bit words. Initially, the message block should undergo the inversion procedure so that last byte should come first and first byte should become last. Every single compression function of Keccak composed of 24 rounds and each round is sub-divided into five steps i.e. Theta (Θ), Rho (ρ) and Pi (π), Chi (χ), Iota (i) explained below

A. Theta (Θ) Step

Theta function encompasses three equations that involves bitwise cyclic shift operations and simple XOR. Equation (1) involves bitwise XOR operation contain 64-bit lanes for each row where every lane of each and every row is independent for other so parallel operations can be applied on this lanes. We are using conventional 64-bit XOR operator in parallel to perform XOR between the five lanes of each row contain the state of array 'A' and results are stored in in-between registers. The parallel XOR operations make our design fast and more efficient in terms of performance. Second step (2) of theta will be having one bit left circular rotation

©IJRASET: All Rights are Reserved

which is convoyed by simple rewiring or replacing the bit pattern of each row, then XORing with the previous output lanes. The results are stored in an intermediate registers in the form of 5 lanes. These lanes are again XORing with input state matrix A[x, y] to form new 5 x 5 state matrix A'[x, y]. All the operations are done on modulo 5.

C[X] = A[X,0]

A[X,1] A[X,2] A[X,3] A[X,4] $\oplus$ X ≤ 4 …..(1)

D[X] = C[X -1] $\oplus$ ROT(C[X+1,1]) 0 ≤ X ≤ 4 ….(2)

A[X,Y] = A[X,Y] $\oplus$ D[X] 0 ≤ X , Y ≤ 4 ….(3)

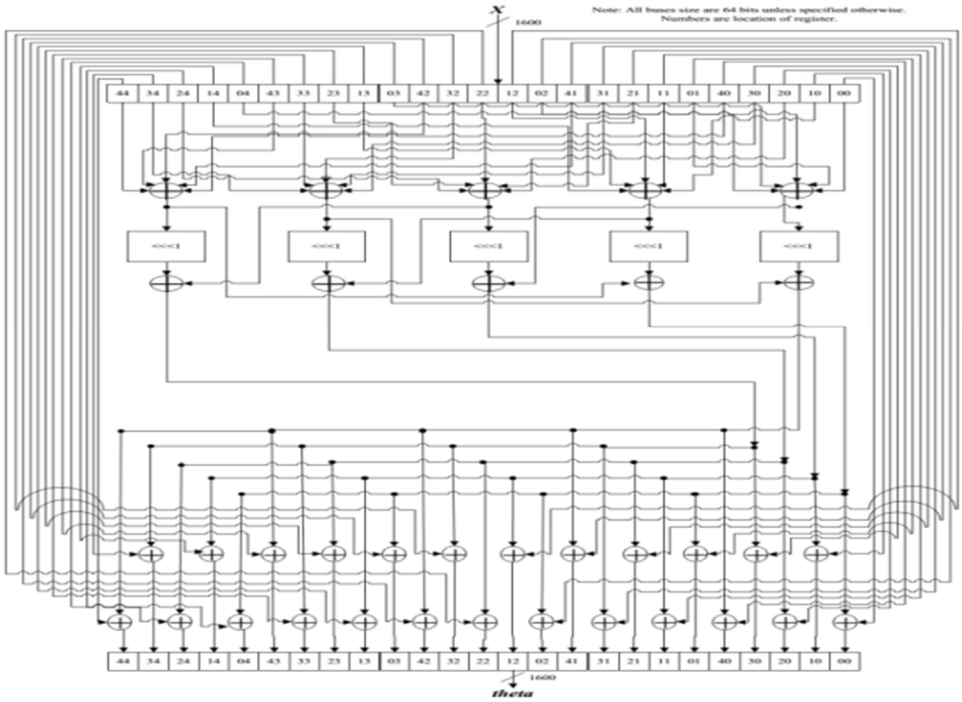


Fig.1 Theta(Θ) step

B. Rho (ρ) and Pi (π) Step

The next two steps are Rho (ρ) and Pi (π) can be expressed in equation (4) compute an auxiliary 5 x 5 array B from the state array 'A'. The operation of Rho and Pi take each of the 25 lanes of the state array 'A', and performing circular rotation and fixing the number of values depending up on the 'x' and 'y' co-ordinates

B[Y,2X+3Y] = (A[X,Y] , r[X,Y] 0≤X,Y≤4 …….(4)

C. Chi (χ) Step

The Chi (χ) step operating on the lanes, words with 64-bits and manipulating the B array obtained in the previous Rho (ρ) and Pi (π) step and interchanges the result in the state array A. We can say that the Chi (χ) step takes the lane at location [x, y] and XOR it with the logical AND of the lane at address location of [x+1,y] and the complement at location [x+2,y]. The equation given below illustrating the function Chi (χ)

A[X,Y] = B[X,Y] $\oplus$ ((NOT B[X+1,Y]) AND B[X+2,Y]) ……(5)

D. Iota (i) Step

The Iota step is the simplest step of SHA-3 algorithm. It will just perform the XOR operation and predefined 64-bit constant RC given in [3] with in the lane located [0,0] of the new state matrix 'A'.

A[0,0] = A[0,0] $\oplus$ RC ……(6)

©IJRASET: All Rights are Reserved

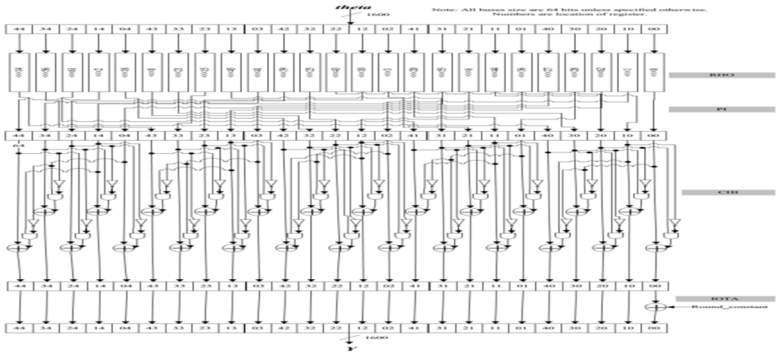


Fig.2 rho (ρ), pi (π), chi (χ) and iota (i) step.

III. STATE-OF-THE-ART

This structures mainly contain folding and the number of pipeline stages. The unfolded structures obtains low throughput, folded structures require less resources and low cost and throughput is lower. It is mainly caused due to rounds in step mapping, As such, additional logic and clock cycles must be used to solve the dependencies. These results in lower T/A efficiency metrics can complicate further optimization. The relevant existing Solutions are either folded lane-wise or slice-wise and the maximum folding factor is respectively 64 and 25, denoting the total number of lanes and slices of the state.

The first folded structure was proposed by Bertonietal.in the SHA-3 Implementation Overview [10], with FF=25 and the state stored in the memory with 215 clock cycles per round. Improvements to this structure is Proposed by Kerckhofetal. [11] where the latency is reduced to 88 clock cycles per round by more efficient control logic which improves the instruction Parallelism[12].

The ρ and θ step-mappings still require additional logic for provision of the necessary input bits. As proposed in [`13], [14], the rescheduling of the round function allows for incorporating the ρ step-mapping into addressing of the state in memory Winderickx *et al.* [15] propose a slice-wise structure with FF=64, so that 1 slice of 25 bits is processed in parallel. The first pipelined unfolded structure with pipeline registers incorporated in the round function. This optimization achieves a very high frequency, but at a cost of high area requirements, resulting in a low T/A efficiency. Additional variations of pipelined structures have also been explored [16], [17],[18] and results suggest that one or two pipeline stages in the round function is the optimal trade-off between increased throughput and area. As is pointed out by Ioannou *et al.* [19], pipelining is not relevant for applications where single larger messages need to be processed.

IV. PROPOSED SOLUTION

Here we are implementing the three a straight-forward implementation, folded structure and piplined structure and comparing the three structures by applying 128bit as hash input and checking the hash output message bit as 1024 bit for the three structures and a straight forward structure the control logic and the IO buffering. The state contains the registers for storing the 1600 bits of the state and the round function contains the 5 step mappings needed to compute SHA-3.and in folded structure we are using Reg A to copy the message and reducing the delay and power, increasing the throughput. In pipeline structure using Reg A and Reg B to copy the message and increasing throughput and reducing the power and delay compared to folded structure.

1) *Unfolded Structures:* The basic structure contains one instance of above-mentioned components, computing one round per clock cycle. A multiplexer controlled by a round counter determines whether the state is updated with the result of the round function or in the from of input block provided by the wrapper during the absorption phase. The separate IO-buffer allows outputting the resulting digest and absorbing a new message in parallel with the processing of a message, as considered in [10]. The resulting structure is identical to the straight-forward ones considered in . The pipelined structure is modified from the basic structure with an internal pipeline register which separates the θ step mapping from the rest of the round function. This is realized by implementing a synchronized input to the ρ step-mapping.
2) *Folded Structure:* Towards a more compact SHA-3 structure, folding of the round computation can be considered. In this case each round is computed over multiple clock cycles, depending on the folding factor (FF). Targeting a throughput of about 1GBs and a moderate folding factor, a FF=4 is proposed. A slice-wise folding structure is considered, resulting in the processing of 16 slices in each iteration. special care must be taken regarding data dependencies in the θ and ρ step mappings, in order to provide the necessary input values for the computation of the slices on each iteration. The ρ step-mapping dependencies can be solved by rescheduling the round computation in a similar manner as Proposed With this, the re-scheduled computation 88

©IJRASET: All Rights are Reserved

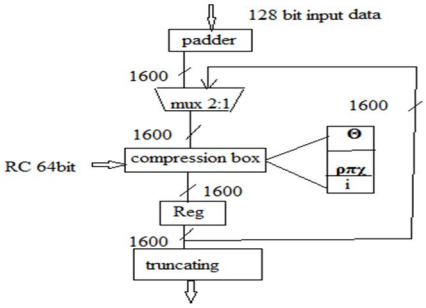


Fig 3.128bit SHA3 folded structure

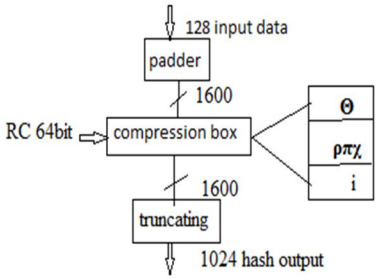


Fig 4. 128bit SHA3 unfolded architecture

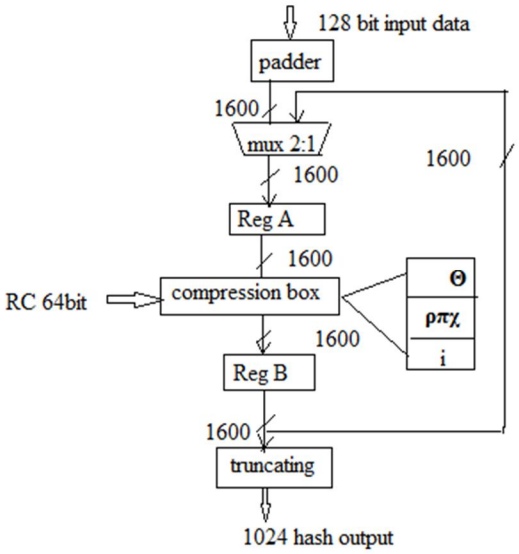


Fig 5. 128bit SHA3 piplined architecture

©IJRASET: All Rights are Reserved

V. RESULTS

A. *Simulation Waveforms*

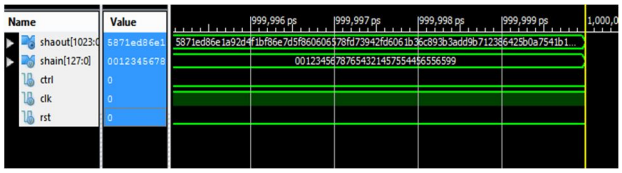


Fig 6.Simulation for straight forward

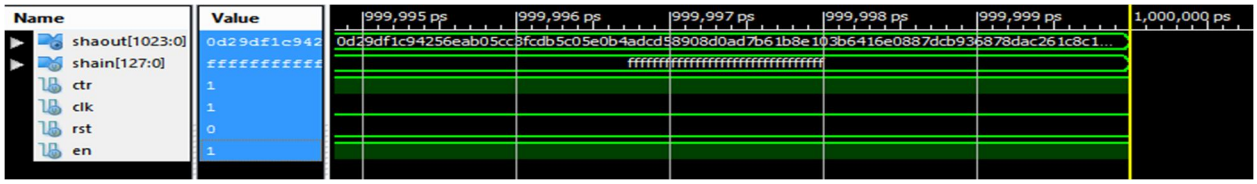


Fig 7. Simulation for folded structure

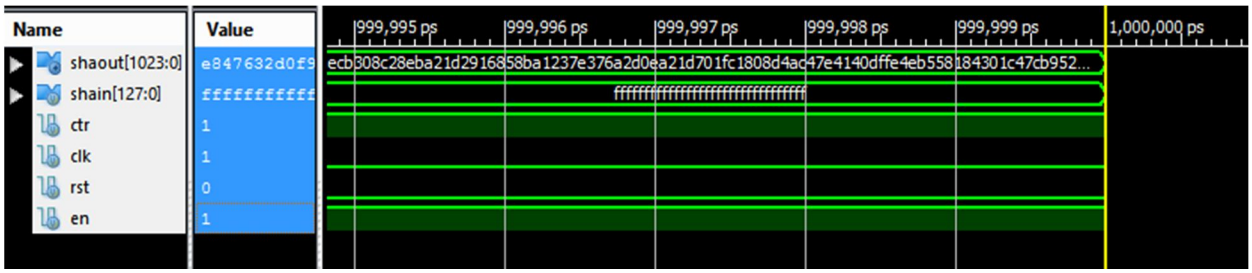


Fig 8. Simulation for pipelined

Table1: Comparison of three straight forward, folded structure and pipe-lined structure

parameters		straight forward	foloded structure	piplined
synthesis report	no of slices	1771/5875	63801/8672	112384/ 2443200
	no of 4 i/p lut	9312/37349	89095/17344	51584/1221600
	bonded iob	24/64	1156/860	1154/850
place and route report	max combi-national delay	9.33ns	6.501ns	2.311ns
	power consumption	0.160mw	0.156mw	0.124mw
	throughput	1.12Gbps	0.68Gbps	1.2Gbps
map report	max fanout of lut	315	1	1
	gate delay	5.471	4.283	0.681

VI. CONCLUSION

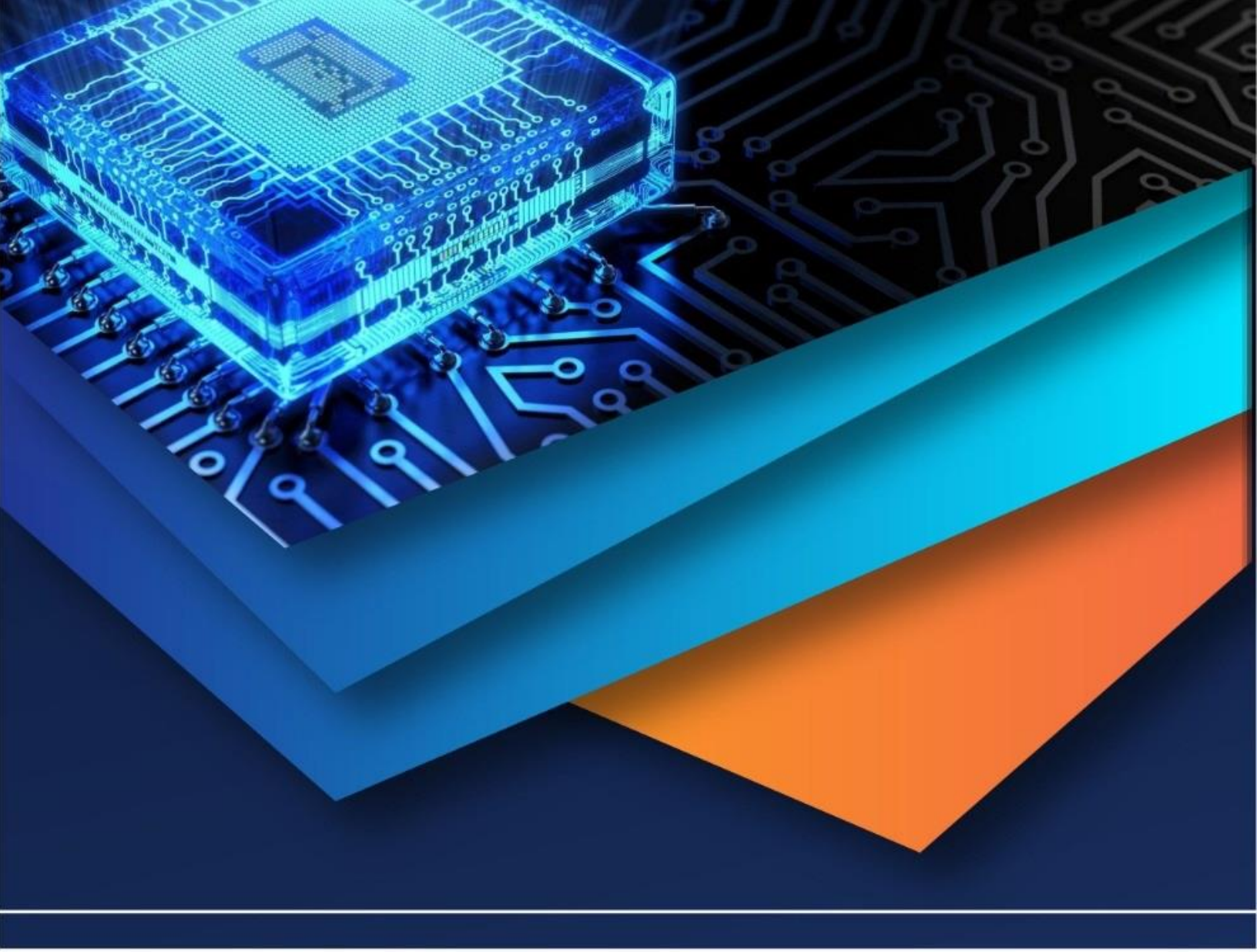
The SHA 3 algorithm provides a good security for the data using the authentication format by generating hash code. fast running speed and can be widely used in digital signatures and 3DES key generation systems and comparing the three structures they are straight forward , folded structure and pipelined architecture by comparing this we get to know that pipelined throughput is higher and delay is reduced then the folded structure and folded structure is having more throughput and less delay then the straight forward and final conclusion is pipelined is having higher throughput and reducing the delay.

©IJRASET: All Rights are Reserved

REFERENCES

[1] X. Wang, D. Feng, X. Lai, and H. Yu, "Collisions for hash functions md4, md5, haval-128 and ripemd," IACR, August 2004.
[2] National Institute of Standards and Technology (nist), "Cryptographic hash algorithm competition," 2007.
[3] FIPS-202, "Federal information processing standards publication fips-202, secure hash algorithm-3 (sha-3)," 2014.
[4] Xilinx, "Virtex 2.5 V field programmable gate arrays". [5] F.R. Henrquez, A.D. Prez, N.A. Saqib, and C.K. Koc, Cryptographic Algorithms on Reconfigurable Hardware. Signals and Communication Technology, Springer, 2007.
[5] S. Kerckhof, F. Durvaux, N.V. Charvillon, F. Regazzoni, G.M. de Dormale, and F.X. Standaert, "compact fpga implementations of the five sha-3 finalists," Springer Berlin Heidelberg, vol. 7079, pp. 217–233, 2011.
[6] A. Akin, A. Aysu, O.C. Ulusel, E. Savas, "Efficient hardware implementations of high throughput sha-3 candidates keccak, luffa and blue mid night wish for single- and multi-message hashing," ACM, pp. 168–177, 2010.
[7] K. Gaj, E. Homsirikamol, and M. Rogawski, "Comprehensive comparison of hardware performance of fourteen round 2 sha-3 candidates with 512-bit outputs using field programmable gate arrays," 2ndSHA-3 Candidate Conference, pp 23-24, August 2010.
[8] B. Baldwin, N. Hanley, M. Hamilton, L. Lu, A. Byrne, M. ONeill, and W.P. Marnane, "FPGA implementations of the round two sha-3 candidates," The second SHA-3 Candidate Conference, 2010.
[9] Guido Bertoni, Joan Daemen, Michael Peters, Gilles Van Assche and Ronny Van Keer. Keccak implementation overview Report STMicroelectronics Antwerp Belgium, 2012.
[10] St´ephanieKerckhof,Franc¸oisDurvaux,NicolasVeyratCharvillon,FrancescoRegazzoni, Guerric Meurice de Dormale, and Franc¸ois-XavierStandaert. Compact FPGA implementations of the five SHA-3 finalists.In International Conference on Smart Card Research and Advanced Applications, pages 217–233. Springer, 2011.
[11] I. San and N. At. Compact Keccak Hardware Architecture for Data Integrity and Authenticaion on FPGAs. Information Security Journal:A Global Perspective, 21, pages 231–242, August 2012.
[12] BernhardJungk. FPGA-based evaluation of cryptographic algorithms.PhD thesis, JohannWolfgangGoethe-Universit¨at, February 2016.
[13] Bernhard Jungk and Marc St¨ottinger. Among slow dwarfs and fast giants: A systematic design space exploration of KECCAK.In (ReCoSoC),2013 8th International Workshop on, pages 1–8. IEEE, 2013.
[14] J. Winderickx, J. Daemen, and N. Mentens. "Exploring the Use of Shift Register Lookup Tables for Keccak Implementations on Xilinx FPGAs". 26th International Conference on Field-Programmable Logic and Applications, Lausanne, 2016.
[15] F´abio Dacˆencio Pereira, Edward David Moreno Ordonez, Ivan DaunSakai, and A Mariano de Souza. Exploiting parallelism on keccak: Fpgaand gpu comparison. Parallel & Cloud Computing, 2(1):1–6, 2013.
[16] Yusuke Ayuzawa, Naoki Fujieda, and Shuichi Ichikawa. Design tradeoffs in SHA-3 multi-message hashing on FPGAs. In TENCON 2014-2014 IEEE Region 10 Conference, pages 1–5. IEEE, 2014.
[17] George S Athanasiou, George-Paris Makkas, and Georgios Theodoridis. High throughput pipelined FPGA implementation of the new SHA-3 cryptographic hash algorithm. In Communications, Control and Signal Processing (ISCCSP), 2014 6th International Symposium on, pages 538–541. IEEE, 2014.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)