ON IMPLEMENTATION OF BLOCKCHAIN ARCHITECTURE USING SHA CONCEPT IN FPGA

[1]Asma Arjumand, [2]G. ARUNA
[1]PG Scholar, M.Tech, Dept of DSCE, SWCET, HYD, T.S, INDIA
[2]Asso. Professor, Dept of ECE, SWCET, HYD, T.S, INDIA.

ABSTRACT

This paper concentrates on the Design of Parameterizable Implementation of SHA-256 calculation in FPGA giving Blockchain Concepts. SHA-256 is the key rule used in Blockchain engineering to grant security and protection into a system. This one-way hash function produces exceptional yield for a given info guaranteeing information realness and nonrepudiation. Blockchain innovation is picking up notoriety in the Internet world because of its property of decentralization. Through this implementation, principle objective is to bring this new innovation into VLSI space for making sure about equipment computerized system designs and SOC's (System on Chip). The proposed strategy empowers any piece length input message to get converted to fixed length message digest known as Hash. The design for the proposed engineering was recreated in Modelsim and combined in Xilinx Vivado Design Suite utilizing Artix 7 FPGA.

INTRODUCTION

The actually expanding and determined exertion for shielding validness and respectability of Internet information has led to the improvement of the most cutting-edge innovation of this time known as Blockchain. This new term was brought into the cryptographic world by a class of individuals known by the name "Satoshi Nakamoto" [1]. In this innovation, PCs known as companions in an organization conveys to one another utilizing consensus calculation to concur consistently upon the most recent transaction or action to be embraced. Blockchain as the name indicates, are chain of squares that speaks to open and decentralized advanced information base among transaction substances or persons which are unquestionable and unchanging.

It is an open record where all the transactions are straightforward to everyone inside the organization. Hence it evades the need of a believed outsider to check the transactions done among the exchanging parties subsequently sparing time and cost. Various transactions done for a particular period shapes a solitary square in blockchain. Chosen peers that are associated with creation of new square are called as Miners. The quantity of transactions per block relies upon the size of a square for that particular application. It changes from a size of 1 MB to 8 MB and beyond. Hash functions and Digital Signatures are the two essential crypto-natives behind Blockchain innovation [2], [3]. In this new innovation, information inside the squares are made sure about cryptographically utilizing complex calculation called SHA-256 (Secure Hash Algorithm). Squares are fastened utilizing hash to confer changelessness. Information is marked carefully utilizing Digital Signatures to guarantee information genuineness.

Numerous investigates have been under advancement for investigating broad application of Blockchain innovation in different fields. The paper [4] features the key qualities of Blockchain, trailed by its applications and difficulties. An improved version of consensus calculation was examined in [5] where in scourge conventions were embraced to perform unicast communication among the friends instead of conventional transmission system to accomplish ideal speed in Information trade. Another consensus calculation called Distributed Byzantine Fault Tolerance calculation (DBFT) was talked about in [6] through which the impact of vindictive clients if present inside the associating gathering of friends could be limited by 2-stage consensus arrangement. The second phase of friends are chosen in irregular to check the contract or transaction. The transaction is performed if both the consensus cycle corresponds with one another. Different explorations on this innovation are edified in [7], [8], [9]. In the event that these circuits are having flaws because of assembling variations, they are analysed through other testing VLSI structures as enrolled in [10], [11], [12], [13]. SHA-256 implementation in [14] uses 7-3-2 blower technique to lessen equipment overhead and speed up. [15] concentrates on the comparison of the above calculation with new hash function called blake-256. Be that as it may, all connected deals with SHA-256 as in [16], [17], [18] have designed the calculation in order to register the hash for a solitary message block.

EXISTING SYSTEM

In the current system, SHA-256 calculation was isolated into two phases: pre-preparing and hash computation. Pre-handling includes cushioning a message and parsing the cushioned message into m-blocks. Initialization esteems are set to be utilized in the hash computation. Hash computation creates a message plan from the cushioned message. The yield hash esteem created by hash computation is utilized to decide the message digest. Hash computation includes message plan, functions, constants and word operations that are produced iteratively so as to get a hash esteem.

The initial step of the SHA-256 hash function is preprocessing; the information message is cushioned. The way toward cushioning the message begins subsequent to getting the message input, and a solitary 1-bit is included toward the finish of the message. At that point, it is trailed by n 0-digit until the length of the message is congruent to 448 modulo 512. The last 64-digit is held for computing the length of the message. Consequently, the general message input is 512-cycle.

PROBLEM IDENTIFICATION

1. To figure SHA-256, the size of hash esteem should be 256, which while registering makes message square of size 512, and in this cycle it considers more number of rounds ie., 64.
2. This presents more measure of deferral and furthermore intricacy in computation, as the entire cycle is to be processed consecutively.
3. In terms of security and future use there is no cycle which permits us to share the scrambled information for additional transactions.

PROBLEM DEFINITION

The paper [4] features the key qualities of Blockchain, trailed by its applications and difficulties. An improved version of consensus calculation was talked about in [5] where in pestilence conventions were embraced to perform unicast communication among the friends as opposed to conventional transmission component to accomplish ideal speed in Information trade. Another consensus calculation called Distributed Byzantine Fault Tolerance calculation (DBFT) was examined in [6] through which the impact of malignant clients if present inside the collaborating gathering of companions could be limited by 2-stage consensus arrangement. The second phase of friends are chosen in arbitrary to confirm the contract or transaction. The transaction is performed if both the consensus cycle matches with one another. SHA-256 implementation in [14] uses 7-3-2 compressor strategy to lessen equipment overhead and speed up. [15] concentrates on the comparison of the above calculation with new hash function called blake-256.

PROPOSED METHODOLOGY

PROPOSED TECHNIQUE

Blockchain can be clarified in a basic manner as a chain of squares where information is put away systematically and each square is interconnected to the past square through its hash esteem, shaping a boundless record of information. Its interior structure has two components: block header and transactions. Each square is ordered with an extraordinary number known as Block ID made cryptographically. SHA-256 hash calculation is the fundamental cryptographic component that shapes the structure square of this new innovation. Connection between the squares is set up by putting away the hash of the past square in the header of the prompt straightaway.

The proposed engineering is designed so that it could produce the hash for any N-bit message contribution subsequent to isolating the message into squares of 512-bits each and plays out the compression function and register allocation iteratively. This is accomplished by defining the message contribution to acknowledge any-piece esteem and producing a reconfigurable message module that outcomes in fixed piece message blocks after suitable cushioning.

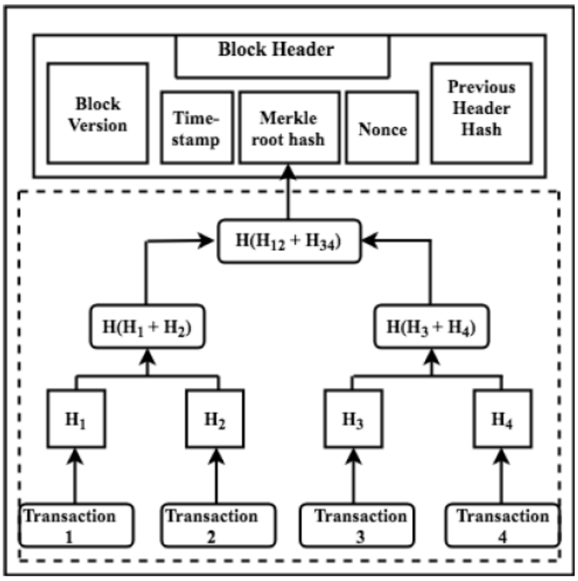


Fig.1 Structure of Blockchain

Its inside structure has two components: block header and transactions. Each square is recorded with a remarkable number known as Block ID made cryptographically. SHA-256 hash calculation is the primary cryptographic component that shapes the structure square of this new innovation. Connection between the squares is set up by putting away the hash of the past square in the header of the quick next. This basic structure gives it the name called Blockchain.

A solitary piece change in the information of the past square will totally change its corresponding hash worth and accordingly causes the ensuing squares hash to be refreshed for the new change. Consequently, a long chain of squares guarantees greater security for all the transactions put away in blocks compelling the information to stay unaltered after some time. Square Header is made out of five fields as enrolled beneath [19], [20]:

1) Header hash

It is the encoded message conceptual figured with block header as info once another square is encased in the chain. This information isn't identified with the information put away inside the square yet it is put away as a different information base that give information about the square. It demonstrates the square position in the chain.

2) Previous Block hash

This component in header is responsible for interlinking of squares. Consequently, the header of i^{th} block contains the hash of I-1th square. This helps each hub in the organization to distinguish the plummeting block from the hash of the past one.

3) Nonce

It is the 32-digit arbitrary worth that is refreshed by the digger until the necessary header hash is gotten on computation. Each hub recomputes for header hash to approve the nonce upon reception of new square.

4) Block Version

This 4-Byte number determines the new version of blockchain.

5) Timestamp

Demonstrates the constant at which the current square is made.

6) Merkle Root Tree Hash

This field contains the historical backdrop of hashes of all transactions remembered for a specific square. It figures the hash, all things considered, and last hash called merkle root is acquired by ascertaining the hash of all go-between hashes.

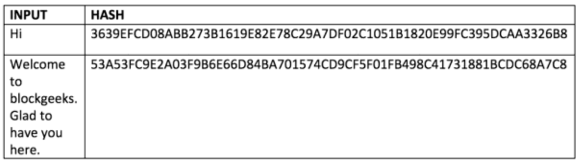
BLOCK CHAIN DESCRIPTION:

A blockchain, initially block chain, is a developing rundown of records, called blocks, that are connected utilizing cryptography. Each square contains a cryptographic hash of the past square, a timestamp, and transaction information (for the most part spoke to as a Merkle tree).

By design, a blockchain is impervious to modification of the information. It is "an open, circulated record that can record transactions between two gatherings productively and in an evident and lasting manner". For use as an appropriated record, a blockchain is regularly overseen by a distributed organization all in all clinging to a convention for between hub communication and approving new squares. Once recorded, the information in some random square can't be adjusted retroactively without alteration of every single ensuing square, which requires consensus of the organization lion's share. Despite the fact that blockchain records are not unalterable, blockchains might be considered secure by design and embody a conveyed processing system with high Byzantine adaptation to internal failure. Decentralized consensus has subsequently been asserted with a blockchain.

The reason why the blockchain has picked up so much admiration is that:

- It isn't claimed by a solitary substance, thus it is decentralized
- The information is cryptographically put away inside
- The blockchain is permanent, so nobody can alter the information that is inside the blockchain
- The blockchain is straightforward so one can follow the information on the off chance that they need to



INPUT	HASH
Hi	3639EFCD08ABB273B1619E82E78C29A7DF02C1051B1820E99FC395DCAA3326B8
Welcome to blockgeeks. Glad to have you here.	53A53FC9E2A03F9B6E66D84BA701574CD9CF5F01FB498C41731881BCDC68A7C8

Table 1: Hash Function Generation of Data

HASH FUNCTIONS

A hash function is any function that can be utilized to plan information of subjective size to fixed-size qualities. The qualities returned by a hash function are called hash esteems, hash codes, digests, or basically hashes. The qualities are utilized to list a fixed-size table called a hash table. Utilization of a hash function to record a hash table is called hashing or disperse capacity tending to.

Hash functions and their related hash tables are utilized in information stockpiling and recovery applications to get to information in a little and almost constant time per recovery, and extra room only fractionally more prominent than the absolute space needed for the information or records themselves. Hashing is a computationally and extra room proficient type of information access which dodges the non-straight access season of requested and unordered records and organized trees, and the frequently exponential stockpiling necessities of direct access of state spaces of enormous or variable-length keys.

HASH TABLES

Hash functions are utilized in conjunction with hash tables to store and recover information things or information records. The hash function deciphers the key related with every datum or record into a hash code which is utilized to list the hash table. At the point when a thing is to be added to the table, the hash code may record a vacant space (likewise called a basin), in which case the thing is added to the table there. On the off chance that the hash code files a full space, some sort of collision resolution is required: the new thing might be overlooked (not added to the table), or supplant the old thing, or it very well may be added to the table in some other location by a predefined system. That technique relies upon the structure of the hash table: In tied hashing, each opening is the top of a connected rundown or chain, and things that crash at the space are added to the chain. Chains might be maintained in irregular control and looked straightly, or in chronic request, or as a self-requesting list by recurrence to accelerate access. In open location hashing, the table is tested beginning from the involved space in a predefined way, typically by straight examining, quadratic testing, or twofold hashing until an open opening is found or the whole table is tested (flood). Looking for the thing follows a similar method until the thing is found, an open space is found or the whole table has been looked (thing not in table).

Table 2: Hash value converted for “abc”

HASH CHARACTERISTICS

Hash functions includes a lot of complex numerical operations and computations that convert an arbitrary length input message into a fixed length yield called digest. The accompanying qualities of this function make its yield remarkable [21].

- It is practically difficult to figure out to reconstruct the first message from the hash esteem.
- A single piece change in info can change the greater part of the pieces in hash bringing about a totally unique yield.
- Algorithm can pack any extensive contribution to fixed size yield.
- Computational infeasibility to locate a similar hash for two distinctive information messages.

SHA-256

SHA-256 is made out of two function modules: Message Block timetable and Compression function.

In message plan, a N-digit message gets included with bit 1 followed by zero pieces until the accompanying equation is fulfilled, where k shows the quantity of zero pieces to be included.

$$N + 1 + k = 448 mod 512$$

The worth N is then converted to its 64-digit paired representation and further added to the 448-piece halfway an incentive to get the 512-cycle message block. This shaped square is additionally partitioned into sixteen 32-bit word sub-blocks which goes as contribution to the compression function [22].

1) Calculate Maj(a,b,c), Ch(e,f,g), $\Sigma_0(a)$, $\Sigma_1(e)$, $\sigma_0(a)$, $\sigma_1(e)$.

2) Words are prepared for each round using the below equation: For first 16 rounds,

$Wn = Message_n^i$

where “n” ranges from 0 to 15 and “i” indicates number of message blocks. For the other rounds,

$Wn = \sigma_1(Wn-2) + Wn-7+ \sigma_0(Wn-15) + Wn-16$

where n can have value from 16 to 63.

3) Six registers b, c, d, f, g, h are updated with the previous registers value i.e., a, b, c, e, f, g respectively after each round of operation. While register a = T1 + T2 and register e = d + T1.

4) T_1 and T_2 have the following equations:

$T_1 = h+ \Sigma_1(e) + Ch + W_n + K_n$

, K are a set of 64 constant words.

$T_2 = \Sigma_0(a) + Maj$

After 64 rounds of operation, registers H_1 to H_7 are updated for “i" ranging from 1 to M as follows:

Final 256-bit Hash value is obtained by concatenating 32-bit values H_0^M to H_1^M

Hash digest = H_0^M H_1^M H_2^M H_3^M H_4^M H_5^M H_6^M H_7^M

PROPOSED SHA-256 HARDWARE DESIGN

The proposed engineering is designed so that it could create the hash for any N-cycle message contribution in the wake of isolating the message into squares of 512-bits each and plays out the compression function and register allocation iteratively. Fig. 2 shows the Architecture of SHA-256.

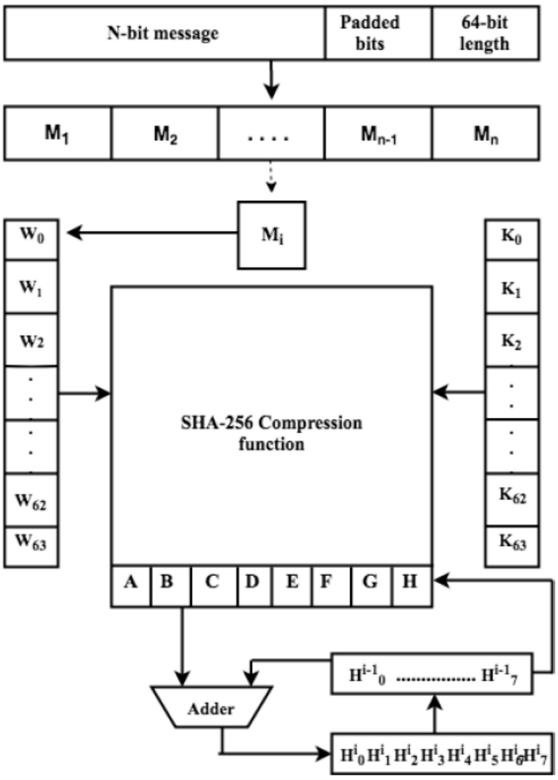


Fig.2: Proposed Sha-256 Hardware Design

This is accomplished by defining the message contribution to acknowledge any-piece esteem and creating a reconfigurable message module that outcomes in fixed piece message blocks after suitable cushioning.

The registers a, b, c, d, e, f, g, h are refreshed for each message block after 64 rounds of compression. Utilizing these qualities, registers H0 to H7 are figured according to equation 6 expressed previously. The new qualities put away in H0 to H7 for (I-1)th message block presently turns into the underlying qualities for registers a to h for the I-th message block and the compression calculation is again performed for another 64-adjusts. In the wake of registering the equivalent for all M squares of messages, last hash digest is gotten.

SIMULATION AND RESULTS

A 24-digit input message "abc" (616263 in hexadecimal) was entered as one of the test vectors and after proper cushioning activity it was changed over to 512-bit 1-block message. The 256-digit message digest acquired after hash calculation were as per the following:

ba7816bf 8f 01cfea414140de5dae2223
b00361a396177a9cb410ff 61f 20015ad

Hash output for 448-bit 2-block message ”61626364 62636465 63646566 64656667 65666768 66676869 6768696a 68696a6b 696a6b6c 6a6b6c6d 6b6c6d6e 6c6d6e6f 6d6e6f70 6e6f7071 80000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 000001c0” were obtained as follows:

248d6a61d20638b8e5c026930c3e6039
a33ce45964ff 2167f 6ecedd419db06c1

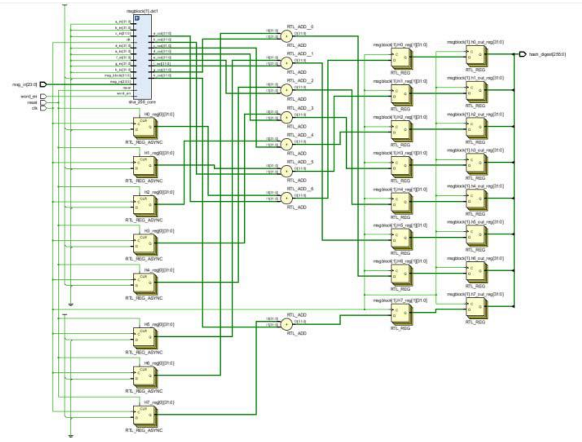


Fig 3: Block Level representation of the Proposed Architecture

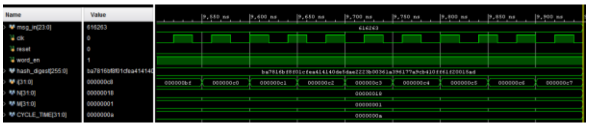
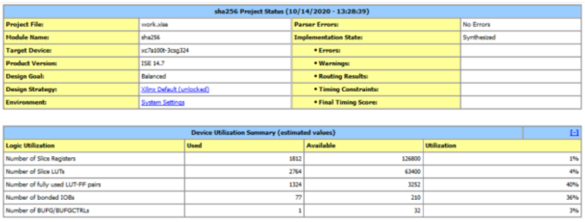


Fig 4: Simulation analysis of Proposed Architecture



sha256 Project Status (10/14/2020 - 13:28:39)			
Project File:	work.xise	**Parser Errors:**	No Errors
Module Name:	sha256	**Implementation State:**	Synthesized
Target Device:	xc7a100t-3csg324	**• Errors:**	
Product Version:	ISE 14.7	**• Warnings:**	
Design Goal:	Balanced	**• Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	
Environment:	System Settings	**• Final Timing Score:**	

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	1812	126800	1%
Number of Slice LUTs	2764	63400	4%
Number of fully used LUT-FF pairs	1324	3252	40%
Number of bonded IOBs	77	210	36%
Number of BUFG/BUFGCTRLs	1	32	3%

Fig 5: Utilization summary of Proposed Architecture

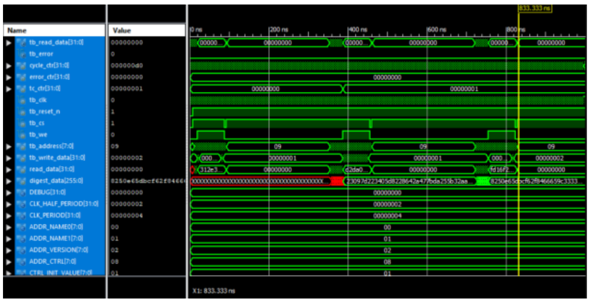


Fig 6: Simulation analysis for Proposed Architecture

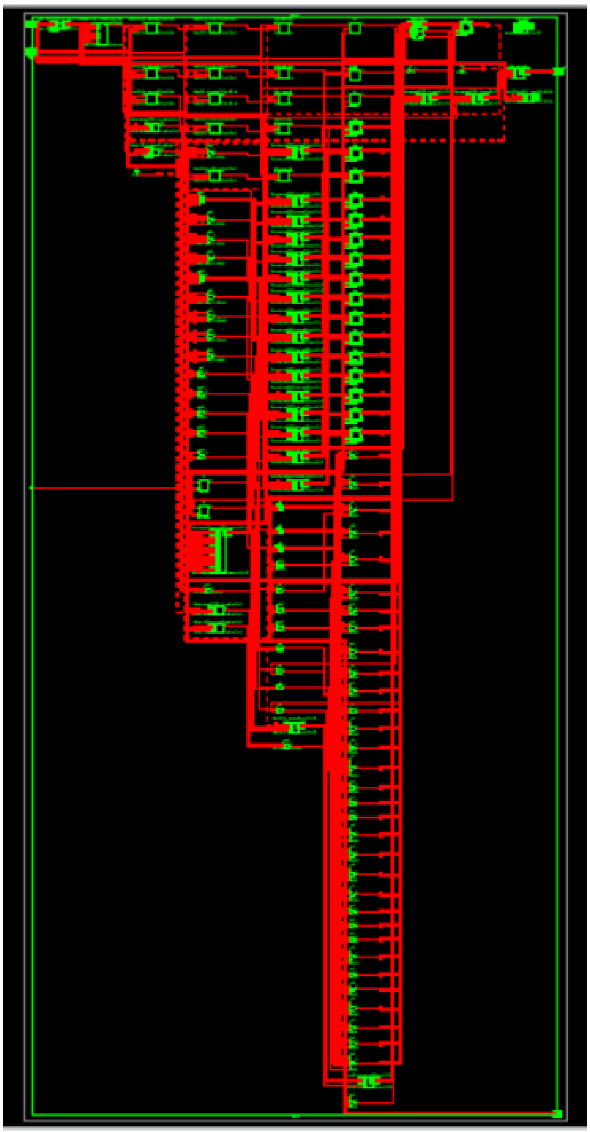


Fig 7: RTL architecture for the Proposed Architecture

CONCLUSION

This paper presents a parameterizable SHA-256 architecture that generates fixed 256-bit message hash for any N- bit length M-block message input. this is often achieved by creating a reconfigurable message module that produce fixed bit blocks of messages. The performance results show that this suggested architecture have higher frequency of operation and better performance as compared to other similar hash function implementations. As a part of future scope, this design might be utilized to implement Efficient Digital Signature architectures with high hardware security. This structure could even be modified to style new consensus algorithm for blockchain as a neighborhood of imparting security in hardware circuits.

REFERENCES

[1] Du Mingxiao, Ma Xiaofeng, Zhang Zhe, Wang Xiangwei, Chen Qijun, “A Review on Consensus Algorithm of Blockchain,” 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 2567–2572, 2017.

[2] Liang Liu, Budong Xu, “Research on Information Security Technology Based on Blockchain,” 2018 the 3rd IEEE International Conference on Cloud Computing and Big Data Analysis, pp. 380–384, 2018.

[3] L. M. Bach, B. Mihaljevic, M. Zagar, “Comparative Analysis of Blockchain Consensus Algorithms,” 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1545–1550, 2018.

[4] N. S. Tinu, “A Survey on Blockchain Technology-Taxonomy, Consensus Algorithms and Applications,” International Journal of Computer Sciences and Engineering, vol. 6, pp. 691–696, May 2018.

[5] Pasu Poonpakdee, Jarotwan Koiwanit, Chumpol Yuangyai and Watchara Chatwiriya, “Applying Epidemic Algorithm for Financial Service based on Blockchain Technology,” 2018 International Conference on Engineering, Applied Sciences, and Technology (ICEAST), pp. 1–4, 2018.

[6] Sol Jeon, Inshil Doh, Kijoon Chae, “RMBC: Randomized Mesh Blockchain Using DBFT Consensus Algorithm,” 2018 International Conference on Information Networking (ICOIN), pp. 712–717, 2018.

[7] Shan Jiang, Jiannong Cao, Hanqing Wu, Yanni Yang, Mingyu Ma, Jianfei He, “BlocHIE: a BLOC k chain-based platform for Healthcare Information Exchange,” 2018 IEEE International Conference on Smart Computing, pp. 49–56, 2018.

[8] Masashi Sato, Shin’ichiro Matsuo, “Long-term public blockchain: Resilience against Compromise of Underlying Cryptography,” 2017 26th International Conference on Computer Communication and Networks (ICCCN), pp. 1–8, 2017.

[9] Md Nazmul Islam, Vinay C Patil, Sandip Kundu, “On IC Traceability via Blockchain,” 2018 International Symposium on VLSI Design, Automation and Test (VLSI-DAT), pp. 1–4, 2018.

[10] Ramesh Bhakthavatchalu, Sreeja Krishnan, V. Vineeth and M. Nirmala Devi, “Deterministic seed selection and pattern reduction in Logic BIST,” 18th International Symposium on VLSI Design and Test, pp. 1–2, 2014.

[11] Devika K N and Ramesh Bhakthavatchalu, “Design of efficient programmable test-per-scan logic BIST modules,” 2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS), pp. 1 – 6, 2017.

[12] Ramesh Bhakthavatchalu and Devika K N, “Design of reconfigurable LFSR for VLSI IC testing in ASIC and FPGA,” 2017 International Conference on Communication and Signal Processing (ICCSP), pp. 0928 – 0932, 2017.

[13] Nisha Haridas and M. Nirmala Devi, “Efficient linear feedback shift register design for pseudo exhaustive test generation in BIST,” 2011 3rd International Conference on Electronics Computer Technology, vol. 1, pp. 350 – 354, 2011.

[14] Ling Bai, Shuguo Li, “VLSI Implementation of High-speed SHA-256,” 2009 IEEE 8th International Conference on ASIC, pp. 131–134, 2019.

[15] Fatma Kahri, Belgacem Bouallegue, Mohsen Machhout and Rached Tourki, “An FPGA Implementation and Comparison of the SHA- 256 and Blake-256,” 14th international conference on Sciences and Techniques of Automatic control computer engineering, pp. 152–157, 2013.

[16] Nalini C. Iyer and Sagarika Mandal, “Implementation of Secure Hash Algorithm-1 using FPGA,” International Journal of Information and Computation Technology, vol. 3, no. 8, pp. 757–764, 2013.

AUTHOR’S PROFILE

Student Name:

Ms. ASMA ARJUMAND has completed her B.Tech from Shadan College Of Engineering and Technology, Peerancheru, RR district, JNTUH, Hyderabad is presently pursuing her masters M. tech in Digital Electronics and Computer Architecture (DSCE) from Shadan Women’s College Of Engineering And Technology, Khairtabad, Hyderabad, TS, India.

Ms. G. ARUNA, currently working as an Associate Professor in Shadan Women’s College of Engineering and Technology, Khairtabad, Hyderabad, TS, India.