An Efficient VLSI Design of AES Cryptography in Memory Implementation

Bijjam.Swathi, Manchalla.O.V.P.Kumar, G.Marlin Sheeba, M.Kiran, Y.Sudarsana Reddy

Abstract*: This paper depicts a novel sub bytes strategy for executing the executing the advanced encryption standard (AES) algorithm that offers a considerably enhanced cryptographic strength. Our strategy depends on composite field math randomization, which involves a low cost of execution while not adjusting the algorithm, does not decrease the recurrence of the work and maintains an ideal similarity to the distributed standard. In this document, we suggest a fast and knowledgeable execution of AES in memory (AIM) to scramble the whole part of the memory only when needed. We use NVM's intrinsic logic working ability to implement the AES algorithm instead of adding extra processing parts to the cost-sensitive memory. The proposed design is implemented using Modelsim 6.4 C and Xilinx tool Verilog HDL and simulated. The proposed framework actualized in FPGA Vertex or Spartan-3.The proposed AES system has been made into an IP and effectively connected in encryption application.

***Keywords*: AES Algorithm, Verilog HDL, FPGA, MEMORY UNIT.**

I. INTRODUCTION

Cryptography is the process of constructing and using a cryptosystem or cipher to prevent all but the intended recipients from reading or using the encrypted information or request. A steganography is a strategy for encoding a signal. The recipient can see the scrambled message with the keys and right computation. Steganography is used mainly to transport touchy information across PC systems cross-sectionally. The encryption procedure takes an unmistakable archive of content and applies a key and a numerical calculation to it, transforming it into crypto-content. In crypto text, unless the user has the key that can undo the encryption, the file is unreadable. In 997, a division of the U.S. government, the National Institute of Standards and Technology(NIST), started a project to Develop a substitute for the Data Encryption Standard(DES). It was widely considered that owing to advances in PC preparation power, DES was not verified. NIST's aim was to characterize a DES trade that could be used by US government offices for non-military data security apps.

It was understood, of course, that industrial and other nongovernmental consumers would benefit from NIST's research and that the work should generally be accepted as a standard of trade .To take an interest in the process of dialog and determination, the NIST welcomed cryptography and information security officials from around the globe. For research, five calculations for encryption were adopted. The Belgian cryptographers Joan Daeman and Vincent Rijmen proposed encryption calculation was chosen through an agreement procedure. Before determination Daeman and Rijmen utilized the name Rijndael (got from their names) for the calculation. The name Advanced Encryption Standard (AES) was provided after choice of the encryption calculation which is used in the same way today. The NIST officially embraced the AES encryption calculation in 2000 and distributed it as a government standard under the FIPS197 assignment. The full FIPS-197 standard is available on the NIST site. True to form, numerous suppliers of programming and equipment for encryption have joined AES encryption into their items.

Using an encryption key and a number of encryption rounds, the AES encryption algorithm is a block cipher. Calculating AES encryption is a square figure with a key and few rounds. A square figure is an encryption calculation that takes a shot at a solitary square of information at any given moment. The square is 128 bits or 16 bytes long due to standard AES encryption. The word-rounds algorithm relates to how the encryption algorithm mixes the information to re-encrypt it ten to fourteen times based on the important duration. This is described in the AES encryption article in Wikipedia. For a computer program or a computer, the AES algorithm itself is not a source code. It is a mathematical description of a method that obscures information. Various people, including the first creators, have produced AES encryption source code executions.

As part of the encryption operation, AES encryption uses a lonely key. The key can be either 256 bits (32 bytes), 192 bits (24 bytes) or 128 bits (16 bytes). The term encryption 128-piece refers to the use of a 128-piece encryption key. Using a similar code, encoding and encryption was achieved with AES.

Revised Manuscript Received on November 15, 2019

Bijjam.Swathi, Mtech student, department of ECE, Gokaraju Rangaraju Institute of Engineering and Technology. Hyderabad, TS, India. Email: swathireddybijjem@gmail.com

Manchalla.O.V.P.Kumar, PH.D scholar, Department of ECE, Sathyabama Institute of science and technology, Chennai, TN, India. Email: pavanomkar@gmail.com

Dr. G.Marlin Sheeba, Assoc. Professor, Department of Electronics and Telecommunication Engineerinng,. Sathyabama Institute of science and technology, Chennai, TN, India. Email: merlinsheebu@gmail.com

M.Kiran, Assoc. Professor, Department of ECE, Gokaraju Rangaraju Institute of Engineering and Technology. Hyderabad, TS, India. Email:kiranmannem14@gmail.com

Y.Sudarsana Reddy, Assoc. Professor, Department of ECE, Gokaraju Rangaraju Institute of Engineering and Technology. Hyderabad, TS, India. Email:ysreddy16@gmail.com

This is called a symmetric encryption algorithm. Encryption algorithms are referred to as asymmetric encryption algorithms using two different keys, a private and a public key. An encryption key is a binary data string used in the encryption method. Since a similar encryption key is utilized to decrypt and encrypt information, it is essential to keep the encryption key a mystery and to utilize keys that are difficult to figure. Some keys are created by programming utilized for this particular undertaking. Another technique is to get a key from a pass expression. Great encryption frameworks never utilize a pass expression alone as an encryption key.

Side channel attacks are on AES, not cipher text outputs. In an effort to imagine the key, it tries to correlate different readings of the encryption instrument with time. A professor at MIT, coded an AES algorithm on his desktop, 850Mhz, Pentium III running Free BSD 4.8 and was able to effectively imagine the key in less than 100minutes by measuring time gaps between the CPU and memory. The index of an array is correlated with the moment it takes to get the outcomes back. This is because of the memory device's physical place of the information. Because it won't take so long for the signal to migrate its way out of the chip, it won't take as much time to retrieve information nearer to the output lead as data away from it.

At this moment, he feels he can enhance. He finished about an hour studying the outcomes of his readings after running a few thousand encryptions. There were many repetitions to prevent noise after studying the information; he found that the key was one of several of opportunities. He was able to discover the key by attempting each one. He thinks that this method of assessment can be programmed to reduce the time to just a few minutes. The technique for estimating time delays in memory solicitations are called timing assaults. Power assaults endeavor to quantify control utilization by the CPU. Switching 8 bits requires more ability than switching 1 bit. Some are likewise now estimating emission layer from CPU's and picking up information of its internal activities.

Some of the side channel attacks are 1)Do not use arrays, S-box and RCon process estimates to abstain from timing attacks.2) To construct a principle's and gadgets to work at steady intervals. Study the speecsheets of your device and insist on correct data. For instance, you should know which operations take longer, XOR or shift. 3) Remember that using the same memory is speedy than DRAM. 4) Calculate key elaborate on the fly. Try not to find out the key elaboration and then refer to it from memory 5) use pipeline to balance the wastage of CPU energy 6) wherever possible, use dedicated chips, they are now considerably quicker than CPUs and require highly costly hardware to measure side channel attacks

II. METHODOLOGY

This paper describes a novel AES Cryptography With Memory to implement the advanced encryption standard algorithm, and Key expansion another input of this paper is designing an Encryption and Decryption design using Mixed column, shift rows, Add round key. Finally with the aid of Encryption and Decryption Block will implement with memory. The results of this paper are for protecting the data with high security. The proposed method is rest on AES Encryption & Decryption based on Memory Design. In this paper, on memory cells the in-memory encryption is directly performed on memory. SA's read the data in memory cells, each divided by a few adjacent columns with a mux as shown in figure-3.Since it is necessary to organize the unit data matrix to be encrypted in a certain way to facilitate the encryption process, we spread the 8 b of each element in the data matrix into dissimilar mats and dissimilar columns in the same mat so that they can be used simultaneously. Before encryption begins it is not necessary to change the plain text data block into matrix during this process.

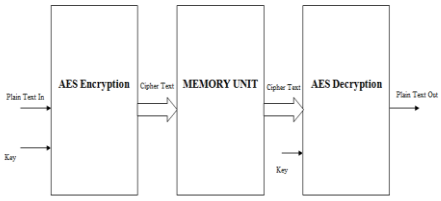


FIG1: PROPOSED BLOCK DIAGRAM

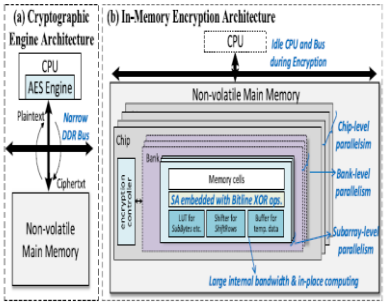


FIG2: MEMORY ENCRYPTION ARCHITECTURE

The figure-3 illustrates distributed data organization for AES encryption. In this algorithm, the processing unit is 1b of data matrix and each matrix is assigned into eight mats of data matrix. Four columns of the data matrix are allocated individually to dissimilar columns of the memory array fixing to four adjacent SAs to encrypt each row of the data matrix in parallel. These four memory array columns have the same local column address. Thus, when one sub array row is initiated and a local column address is selected for each mux, each four adjacent SAs will detect a data matrix row.

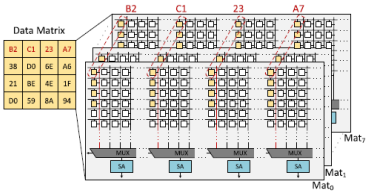


FIG3: DISTRIBUTED DATA ORGANIZATION FOR AES ENCRYPTION

III. IMPLEMENTATION

Implementation is the action that transforms procedure and plans into actions to achieve calculated goals and objectives. Implementation by computer programming and deployment is the realization of a technical specification or algorithm as a program, software component system.

By using Modelsim to implement the flowing blocks booth encoder partial product generator, adder and ecw.

EXISTING SUBSTITUTION BOX (S- BOX):

A non-linear byte replacement is carried out by the sub byte. Each byte from the input state is returned by another byte in the s-box. The S-box in the finite field GF(28) is calculated on the basis of a multiplicative inverse and bit wise affine transition. The s-box composite field is implemented using combination logic circuits instead of pre-stored s-box values.

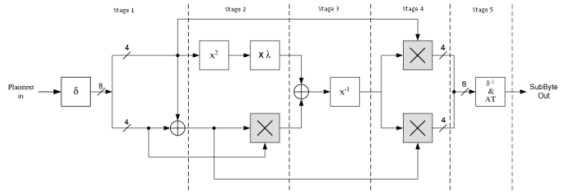


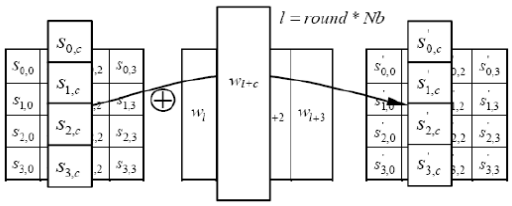
FIGURE: A Non-linear Multiplicative Inverse

Table: S BOX Truth Table:

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5x	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6x	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7x	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8x	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9x	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
ax	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
bx	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
cx	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
dx	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
ex	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
fx	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

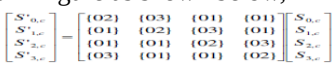
ADDROUNDKEY TRANSFORMATION:

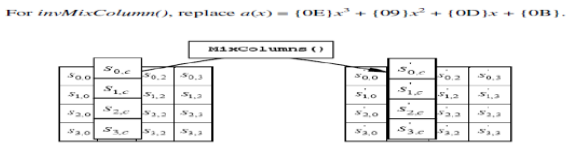
In this process the output of mix column is taken as input to the add round key. In the modification to AddRoundKey, the operation bitwise Exclusive-OR(XOR) adds a round key to the state. The following Figure shows the AddRoundKey, This process is same for both encryption and decryption. Decryption is the inverse process of encryption.



MIX COLUMN TRANSFORMATION:

In this process the output of shift rows is taken as input to the mix column operation. The mix column operation is performed by column by column matrix and each column is taken as four term polynomial over GF and multiplied by the equation of a(x) mod(x4+1) where a(x)={03}x3+{01}x2+{01}x+{02}, this modification is expressed in figure as shown below,

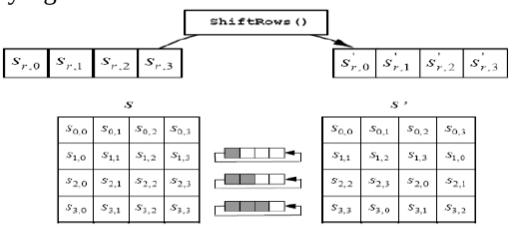




The inverse mix column is the reverse process of the mix column.

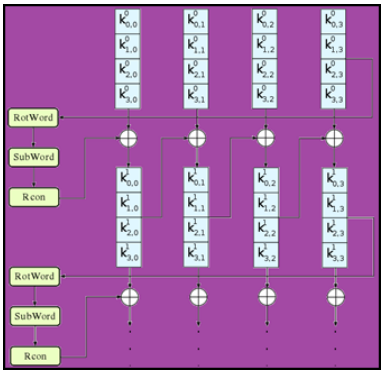
SHIFT ROWS TRANSFORMATION:

In this process the output of sub bytes is taken as input to the shift rows operation. In this process the operation is performed by shifting bytes cyclically. Here, the first row of a matrix remains constant. And the second row the bit is shifted by one bit left and in third row two bits is shifted and in fourth row three bits are shifted left in encryption process. In decryption the process is reverse here the bits are shifted cyclically right.



KEY EXPANSION UNIT:

The main design sends a comparable scheme with further optimizations for s-box and loading information for distinct key sizes into the important registers as in the encryption manner. When not used to spare the dynamic energy usage, the s-box inputs are masked by steady values. The expanded key is determined on the fly and returned to spare land straight to the main registers. Key development module involves two 4-stage shift registers and a main transform module with four s-boxes and a xor.



ROTCON, ROT WORD:

For the primary cycle of a round, these s-boxes are empowered for 128 and 256-b keys and for 192-b keys every six cycles. They remain inert from that stage on. This leads to a 30% reduction in the s-power usage which includes the main growth. RotWord step can be evacuated as it trades the bytes condition in a 32-b signal. The RCon constant can be modeled as a shift register, but we intended it as a ten-value LUT in our architecture to minimize the amount of registers to reduce and minimize power consumption. Key to convert the network of clocks. The RCON XOR with the output of the s-boxes after RotWord is minimized by only the bits required by XORing. The main developments 32-b output is sent directly forward encryption route in the add round key phase to be XORed. The clock gating technique is also applied in the key expansion to save power consumption. During the idle state, the key register and the s-boxes will not create any activates.

IV. FLOW DIAGRAM

This means that for both encryption and decoding it uses a comparable key. Be that as it may, in different ways. AES is not the same as DES.

The Rijindael calculation requires into consideration an assortment of square and key sizes, not just the square and key size 64 and 56 pieces of DES. Depending on which modification is used, the normal name will be altered. Separately to AES-128,AES-192 or AES-256. However, the AES standard states that a square size of 128 bits and a choice of three keys-256,192,128 bits can be acknowledged. The name of the standard is changed separately to AES 256,192,128 bits depending on which adaptation is used. Just as these distinctions AES contrasts from DES in that it's anything but a feistel structure. Review that in a feistel structure, half of the information square is utilized to adjust the other portion of the information square and after that the parts are swapped. During each round, the entire information square is prepared in parallel using substitutions and stages for this situation.

The Rijndael calculation is a symmetric iterated square figure. The key lengths and square can be the bits of 256,192 & 128. The NIST mentioned that the AES must execute a symmetric square figure with a size of a square is 128 bits. Because of this prerequisite, varieties of Rijndael that can work on bigger square sizes won't be incorporated into the real standard. Rijndael additionally has a variable number of cycles or adjusts: 14, 12, and 10 when the key lengths are 256, 192, and 128 individually. The changes in Rijindael consider the information hindrance to be a rectangular 4 byte vector cluster in four segments. The key is also viewed as a 4 byte vector rectangular cluster the number of segments depends on the key length. Rijndael unscrambling involves the converse of the changes that encryption utilizes, performed backward request. Unscrambling begins with the converse of the last round, trailed by the inverses of the rounds, and completes with the underlying information/key expansion, which is its very own reverse.

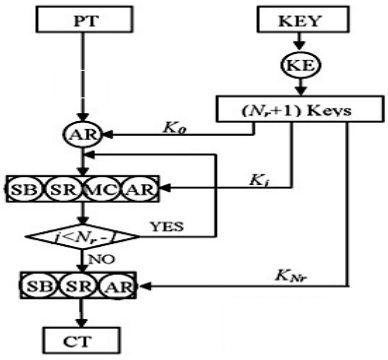
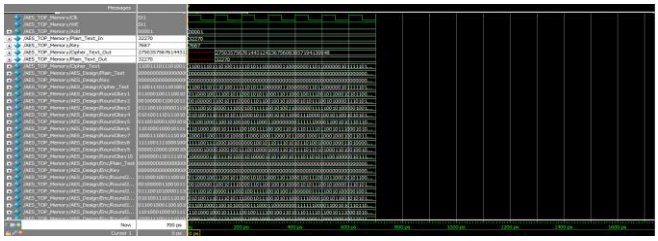


FIG4: FLOW DIAGRAM

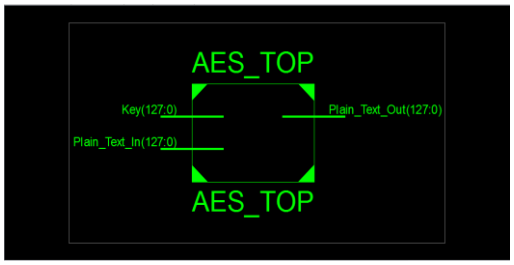
V. RESULTS

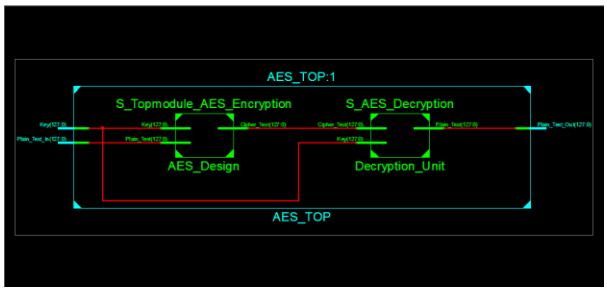
FINAL TOP MODULE:

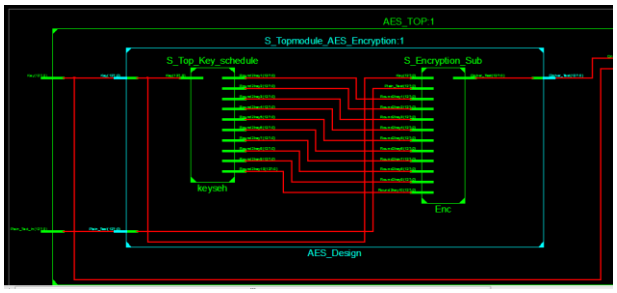
AES DESIGN WITH MEMORY UNIT:

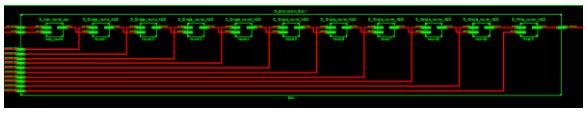


RTL SCHEMATIC:

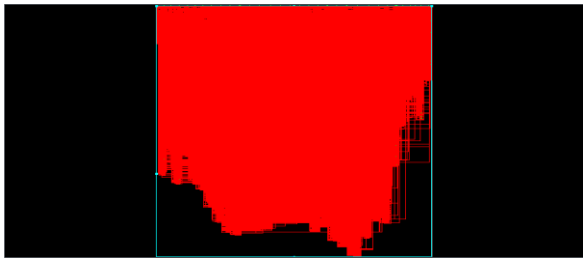








TECHNOLOGY SCHEMATIC:

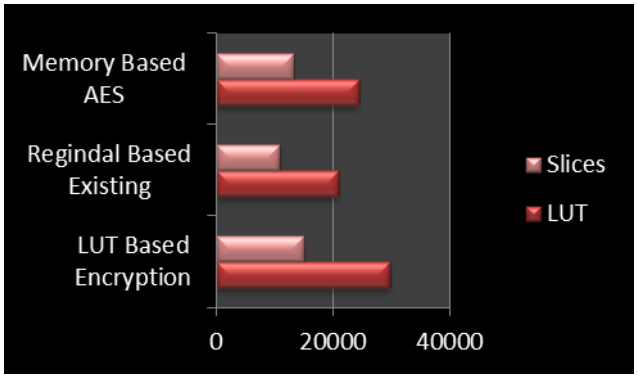


COMPARISON TABLE:

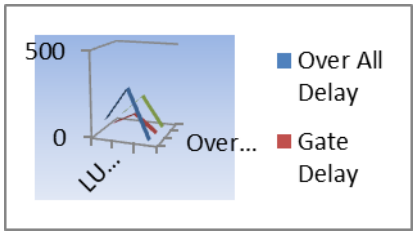
AREA AND DELAY COMMPARISON TABLE:

Method Name	Area in Number of LUT		Delay		
Spartan 3 XC 3S 5000-4FG1156	LUT	Slices	Delay	Gate or Logic Delay	Path or Route Delay
LUT Based Encryption	29744	14962	103.577ns	42.365ns	61.212ns
Existing Rejindal Based Encryption	20852	10807	293.699ns	109.370ns	184.329ns
Memory based Encryption	24,337	13175	7.165ns	6.364ns	0.801ns

AREA GRAPH:



DELAY GRAPH:



VI. CONCLUSION

A novel idea of implementation in memory using the resistive nature of NVM and using existing peripheral memory circuits. With AIM, the memory blocks are simultaneously encrypted with in each memory bank and the entire encryption process with in the main memory can be completed without exposing the results to the memory bus. Rijindael, AES Design implementation with High Security Constrains. Implementation relies on and stays numerical characteristics of the Rijndael algorithm. Another input of this article is that it also uses shift rows to design encryption, mixed columns, add round key, and we will design a decryption part as well. The new design allows efficient area and speed characteristics to be constructed while maintaining a very high level of protection. With the key generation method, we conducted relevant AES implementation. Nearly every calculation inserted in cryptographer was intended to oppose direct, differential and high-order differential attacks in an abnormal state, whereas nothing was done to make them resistant intrinsically safe attacks. However, this work will be implemented in the FPGA. It is possible to design algorithms, so when the next generation of cryptographic.

IV ACKNOWLEDGMENT

The authors would like to thank the authors, Dr. N Swetha, Head of department, Department of Electronics and communication engineering, Gokaraju Rangaraju Institute of Engineering and technology, Hyderabad and Dr. E.Logasanmugam, professor and Director, Sathyabama Institute of Science and technology.

REFERENCES

1. M. Masoumi," Differential power analysis: A serious threat for FPGA security", int.j.internet technology and secured transactions,vol.4,no.1,2012.
2. Yuhan, Xuechengzou, Liu Zhenglin and Yi-chengchen," The research of dpa attacks against AES implementation",j.china univ.posts Telecommun.vol.15, no.4.pp.101-106,Dec2008.
3. J.Zhou and M.Yung.Eds."AES against first and second-order differential power analysis applied Cryptography and Network Security",vol.6123,springer-verlag,pp.168-185.Berlin,Germany. 2010.
4. D. Halperin, T. Kohno, T. S. Heydt-Benjamin, K. Fu, and W. H. Maisel, "Security and privacy for implantable medical devices," IEEE Pervasive Comput., vol. 7, no. 1, pp. 30–39, Jan./Mar. 2008.
5. M. Mozaffari-Kermani and A. Reyhani-Masoleh, "A lightweight high performance fault detection scheme for the Advanced Encryption Standard using composite fields," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 1, pp. 85–91, Jan. 2011.
6. X.Zhang and K.K.Parhi, "High-speed VLSI architectures for the AES algoritham", IEEE trans.VLSIsyst., vol.12,no.9,pp.957-967.
7. D.Gu,J.Li,S.Li, ZMa,Z.Guo, and J.Liu," Differential fault analysis on lightweight block ciphers with statical cryptanalysis techniques", FDTC,september 2012.
8. "Design of optimal fast adder" by Pavan Kumar, M.O.V.; Kiran, M. in International Conference on Advanced Computing and Communication Systems *(ICACCS), 2013*
9. Jamal, K., Chari, K. M., & Srihari, P. (2019). Test Pattern Generation using Thermometer Code Counter in TPC Technique for BIST Implementation. *Microprocessors and Microsystems*.
10. K. Jamal, P. Srihari, K. Manjunatha Chari, B. Sabitha "Low Power Test Pattern Generation Using Test-Per-Scan Technique for BIST Implementation "ARPN Journal of Engineering and Applied Sciences.
11. K.Jamal, Dr.P.Srihari, G Kanakasri "Test Vector Generation using Genetic Algorithm for Fault Tolerant Systems*"International Journal of* Control Theory*and Applications* (IJCTA), 9(12), 2016.
12. K.Jamal, Dr.P.Srihari "Low Power TPC using BSLFSR" I*nternational Journal of Engineering and Technology* (IJET), Vol 8 No 2 Apr-May 2016. Page no.759.
13. K.Jamal, Dr.P.Srihari "Analysis of Test Sequence Generators for Built-In Self-Test Implementation" 2nd International Conference on Advanced Computing and Communication Systems

AUTHORS PROFILE

Bijjam.Swathi Completed Bachelor's degree from JNTU, Hyderabad. Presently doing M.Tech in VLSI Stream in Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, TS, India

Manchalla.O.V.P.Kumar Ph.D scholar in school of Electrical and electronics, Sathyabama Institute of science and technology. Received his Bachelor's degree from ANU, Guntur and Master's degree from Sathyabama University in the year 2007. Presently working as Assistant Professor in Department of Electronics and communication Engineering, GRIET, Hyderabad. He has published and presented 7 papers in International / National journals and conferences.

G.MerlinSheeba received her B.E. (Electronics and Communication) degree in the year 2003 in National Engineering College, Kovilpatti and M.E. degree in the year 2005 from Karunya Institute of Science and Technology, Coimbatore. She has obtained her Ph.D degree from Sathyabama University, Chennai in the year 2017. She is presently working as Associate Professor in the Department of Electronics and Telecommunication Engineering in Sathyabama Institute of Science and Technology, Chennai. She has published and presented 28 papers in International / National journals and conferences. Her research interests include Wireless Mesh Networks, Wireless Body Area Network, Green communication, Evolutionary algorithms and Millimeter wave Antennas