VIKAS GROUP OF INSTITUTIONS

(Sponsored by Mother Theresa Educational Society)
Approved by AICTE, New Delhi, PSI, New Delhi & Affiliated to JNTUK, Kakinada
ISO 9001 : 2015 Certified
Nunna – 521 212, Vijayawada Rural, NTR District, A.P. India.
Email: principal.9t@gmail.com

3.3.1 Number of research papers published per teacher in the Journals as notified on UGC CARE list during

AY 2021

S.No	Title of paper	Name of the author/s	Department of the teacher	Calendar Year of publication
1	Design a High speed RIFFA based Homomorphic Encryption /Decryption	G.SEKHAR REDDY	Dept of ECE	2021
2	Design high Speed Efficient IVLSI Architecture of E.R hybrid Adder	G SEKHAR REDDY	Dept of ECE	2021
3	Implementation of 32 and 32 dadda multipier	G.SEKHAR REDDY	Dept of ECE	2021
4	Design and implementation of high efficient delay utillisation of (n/2) parallel multiplier	G.SEKHAR REDDY	Dept of ECE	2021
5	Design and Implementation of triple bit error correcting bch decoder with high efiiciency for emerging memories	G.SEKHAR REDDY	Dept of ECE	2021
6	Design of efficient 32-bit vedic multiplier	G.SEKHAR REDDY	Dept of ECE	2021

[illegible]

PRINCIPAL/DIRECTOR
VIKAS GROUP OF INSTITUTIONS
NUNNA - 521 212
Vijayawada Rural, NTR Dist., A.P.

DESIGN A HIGH SPEED RIFFA BASED HOMOMORPHIC ENCRYPTION/DECRYPTION

[1]K.ROJA, [2]G SEKHAR REDDY
[1]M.Tech Scholar, Dept of ECE, Vikas Group of Institutions, Nunna, Vijayawada, Andhra Pradesh, India
[2]Assistant Professor, Dept of ECE, Vikas Group of Institutions, Nunna, Vijayawada, Andhra Pradesh, India

ABSTRACT: In this project, design and implementation of high speed and high secure BFC based Homomorphic encryption is done. This system will provide better security and resource efficiency compared to existing standards. RIFFA based homomorphic encryption technique guarantee both privacy and integrity. The main intent is to increase the speed of operation. Initially, input bits and key is expanded serial by using PCIe. Next, bits are substituted using S-Box. After substitution of bytes, the bits will be reusable using RIFFA. After reusable procedure shifting operation is performed. Now these bits are encrypted. Similarly, decryption process is reverse to this operation.. Hence RIFFA based homomorphic encryption and decryption is implemented and it gives better security compared to exist one.

KEY WORDS: Homomorphic encryption, Large Integer Multiplication, Operand Reduction, VLSI Architecture, S-Box, Peripheral Component Interconnect express (PCIe), Reusable Integration Framework for FPGA Accelerators (RIFFA).

I. INTRODUCTION

Fully Homomorphic Encryption is for the most part utilized in the database of the board frameworks (DMBS). One of the present issues related with the utilization of databases is the test of verifying and securely putting away the legitimate treatment of classified information in the remote database. Privacy of touchy data can be guaranteed using cryptography.
It may, be the utilization of industrious encryption calculations to store the data in remote databases can fundamentally decrease the presentation of the framework without interpreting. To take care of the issue, in MIT examines exhibited Crypto system.

Utilizing additively homomorphic crypto framework enables the server to execute SUM, AVG, and Count Questions over encoded information; the other SQL inquiries utilize the distinctive encryption calculations with the vital usefulness. The adjustment of completely homomorphic cryptosystem will keep the capacity to perform run of the mill database tasks on encoded information without decoding the information in a confided condition. In any case, such a cryptosystem must fulfill certain prerequisites for practical qualities and computational unpredictability, which is significant.

Fully Homomorphic Encryption (FHE) is a huge achievement in cryptographic research in recent years. A FHE plan can be utilized to elective perform calculations on figure content without trading off the substance of relating the plain text [1]. Therefore, a practical FHE plan will open the way to various new security advances and protection related to the applications, for example, security safeguarding pursuit and cloud-based processing. For the most part, FHE can be ordered into three classifications: cross section based, number based, and learning with mistakes.

One of the fundamental difficulties in the improvement of FHE applications is to moderate the amazingly high-computational intricacy and asset necessities [2-4]. For instance, programming usage of FHE in superior PCs still expend the critical calculation time, especially to achieve the vast whole number duplication which more

[illegible]

PRINCIPAL/DIRECTOR
VIKAS GROUP OF INSTITUTIONS
NUNNA - 521 212
Vijayawada Rural, NTR Dist., A.P.

often than not includes more than countless bits. For cross section based FHE, bit increase the required for the little setting with a grid measurement. To quicken the FHE tasks, different effective plans have been proposed to handle the extensive whole number duplication.

The objective of this paper is to revive the encryption natives in entire number based FHE using FPGA advancement. This particular FHE count is picked because of the less unpredictable theory, humbler key size and equivalent execution. Also, the introduction of a grouped FHE plots over the entire numbers ensures further capability upgrades. Augmentation is a key segment in these FHE plans the features in the encryption, unscrambling and evaluation steps. Broad entire number FFT duplication has furthermore been used in the late of referenced gear and GPU use of other FHE plans. Future work will look into the impact of the gear multiplier on substitute walks inside the FHE plot. Specifically, presenting the primary gear execution of encryption rough required for FHE over the numbers.

II. EXISTED SYSTEM

The below figure (1) shows the architecture of existed system. In this system mainly, two NTT units, a controller unit, an AGU, and several memory units are used. ROM main intent is to store the twiddle factors. There are mainly two single ports of SRAM in NTT block. Here firstly two inputs are computed at same time by using the two NTT data there are NTT1 and NTT2. For the purpose of multiplication the NTT is used as inverse NTT and because of R input data is processed.

Addition and subtraction operations are performed in the Mul Mod unit. The result of this unit is processed to the buffer unit. Now the values are saved in ROM. Here point wise multiplication process is performed in the NTT block and bits are computed depends on the current status of operation.

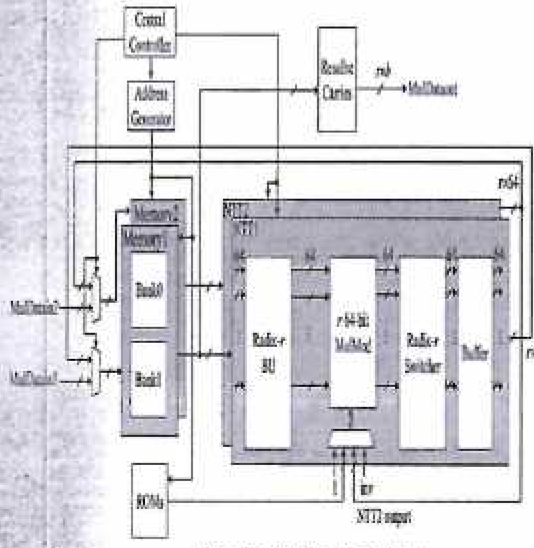


Fig. 1: Existed system

To relocate the data radix r is used and this will saves the memory temporarily. Basically there are four pipelined stages in the MulMod unit. To get conflict free address in the system buffer is used. But this system does not give effective results in terms of delay and time. Hence to overcome this, a new system is introduced which is discussed in below section.

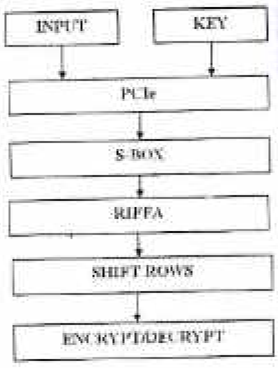
III. PROPOSED SYSTEM

The below figure (2) shows the proposed system. This system will provide better security and resource efficiency compared to existing standards. RIFFA based Fully homomorphic encryption technique guarantee both privacy and integrity. The main intent is to increase the speed of operation. Initially, input bits and key is expanded serial by using PCIe. Next, bits are substituted using S-Box. After substitution of bytes, the bits will be reusable using RIFFA. After reusable procedure shifting operation is performed. Now these bits are encrypted. Similarly, decryption process is reverse to this operation. The description of each block is given in detail manner.

[illegible]

PRINCIPAL/DIRECTOR
VIKAS GROUP OF INSTITUTIONS
NUNNA - 521 212
Vijayawada Rural, NTR Dist., A.P.

ISSN NO: 0022-1945



- Peripheral Component Interconnect express-PCIe
- Reusable Integration Framework For FPGA Accelerators-RIFFA

Fig. 2: PROPOSED SYSTEM

A. SUBSTITUTE BYTES TRANSFORMATION (S-BOX)

The modified structure starts with changes in the Sub bytes step. The function of this step is to substitute data present in the S-box memory unit within the state by diverse data present in other memory unit. The dispersion of data in memory units creates the confusion. The main purpose of this Shannon's contents for scientific restraint arrangement is to stimulate security. The basic purpose of substitution of bytes is to secure information.

B. ENCRYPTION

Encryption algorithm is a combination of complex mathematical functions which are used to encrypt the confidential information. Encryption key is a secret values that the sender utilizes as one of the inputs to the encryption algorithm in conjunction with plain text to generate a cipher text..

C. REUSABLE INTEGRATION FRAMEWORK FOR FPGA ACCELERATORS (RIFFA)

Our goal is to expand the use of FPGAs as an acceleration platform by releasing, as open source, a no cost framework that easily integrates software on traditional CPUs with FPGA based IP cores, over PCIe, with minimal custom configuration. RIFFA requires no specialized hardware or fee licensed IP cores.

D. PERIPHERAL COMPONENT INTERCONNECT express (PCIe)

Peripheral Component Interconnect Express (PCIe or PCI-E) is a serial expansion bus standard for connecting a computer to one or more peripheral devices. PCIe provides lower latency and higher data transfer rates than parallel busses such as PCI and PCI-X.

E. DECRYPTION

Decryption is taking encoded or encrypted text or other data and converting it back into text you or the computer can read and understand. This term could be used to describe a method of unencrypting the data manually or unencrypting the data using the proper codes or keys

IV. RESULTS

The below figure (3) & (4) shows the RTL schematic and technology schematic of RIFFA based homomorphic encryption and decryption system.

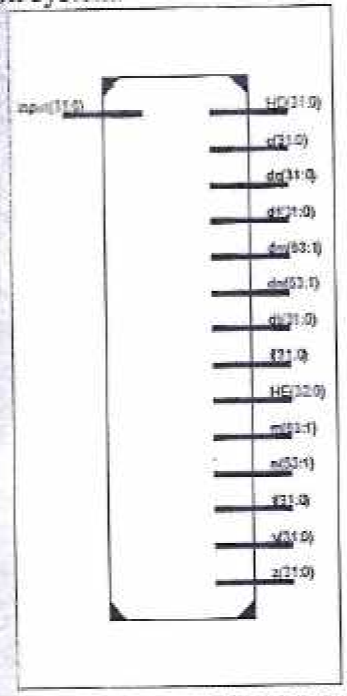


Fig. 3: RTL SCHEMATIC OF PROPOSED SYSTEM

[illegible]

PRINCIPAL/DIRECTOR
VIKAS GROUP OF INSTITUTIONS
NUNNA - 521 212
Vijayawada Rural, NTR Dist., A.P.

DESIGN A HIGH SPEED EFFICIENT VLSI ARCHITECTURE OF E.R HYBRID ADDER

[1]VALLURU PRAGATHI, [2]G SEKHAR REDDY
[1]M.Tech Scholar, Dept of ECE, Vikas Group of Institutions, Vijayawada, A.P, India
[2]Assistant Professor, Dept of ECE, Vikas Group of Institutions, Vijayawada, A.P, India

ABSTRACT: In this paper the Design a high speed efficient VLSI architecture of E.R Hybrid adder is implemented. Basically, adders are most commonly used in the applications of digital signal processing's, Microprocessors etc. Hybrid adder performed in three stages, they are propagator and generator stage, Internal carry generation stage and final sum stage. In this propagator and generator signals are generated by using propagation stage. Internal Carry generation stage will generate the carry. If any errors are occurred then error detection stage will detect the errors and error recovery stage will recover those output values. Final sum stage will take the output of hybrid adder and performs the operation and gives sum as output. At last from results it can observe that the propose system gives effective results.

KEY WORDS: Hybrid Adder, Internal Carry generation Stage, Final Sum Stage, propagator, Generator, VLSI.

I. INTRODUCTION

Fastest technologies are developed in present days. In present days, reduction of device size, fast operation and low power consumption are required. The designing of low power VLSI system has more demand in mobile communication. Due to the device designed by designer with high speed, low power consumption and small silicon area, the device is available with low power.

ALU (Arithmetic logic unit) and FU (Floating point unit) are the main parts in computations [1]. Logical computations are addition, subtraction, multiplication, division and logical operations are AND, OR, INV and comparison which are processed by Arithmetic logic unit (ALU). Data path has an important role in digital signal processors and microprocessors because of some characteristics such as power consumption, speed of operation and die-area. Data path contains complex operations are subtraction, addition, division and multiplication [2]. The main important factor is data path performance which is affected by efficient hardware units of complex computations. In the data path addition is the important executed operation, addition operation contains binary adder to add given numbers. In complex computations such as decimal operations, multiplication and division, adders has important task [3]. To get data path efficiently, the implementation of binary adder should be efficient.

In central processing unit (CPU) crucial element is ALU (Arithmetic logic unit). An adder has important function in ALU and an adder performs not only addition but also performs multiplication, subtraction and decrement/increment. In ALU and general processors to get better performance, efficient adder is needed. From 1950s, for hardware implementation of VLSI arithmetic circuits, research started on efficient adder implementation. In control systems and digital signal processing main operation is the addition.

The properties of system or processor like accuracy and speed depends upon the performance of adder. To execute the addition of numbers, adder is used which is a digital circuit. Different processors and computers contains ALU in which adder is used. To reduce different parameters, different designs have been implemented based on parallel and serial structures. Four elementary operations are performed in binary addition.

The adders can be represented in many forms like BCD (binary coded decimal)

[illegible]

and excess-3 code; binary numbers are used in adders to perform the operation. Negative numbers are represented by ones complement or two's complement, for this adder is modified as adder-subtractor. More logic is required to represent signed numbers including basic adder [4-5].

To execute the addition operation, computers contain Arithmetic Logic unit (ALU) in which adders are mostly used. Graphics processing unit (GPU) and central processing unit uses the adders to decrease the redundancy for the graphics applications. In the adders first type is half adder it include two inputs and it provides two outputs such as carry and sum. Next one is full adders which include two inputs with carry input and it provides two outputs such as carry and sum. For single bit, both half adder and full adder is utilized. The full adder is coupled in parallel form to perform the multi bit addition operation.

II. RELATED WORK

Binary addition performs the addition process based on the logic gates. Here single or two bit binary numbers are used. In binary addition process sum and carry are the outputs. Bit serial adder is the name of serial adder. Binary addition operation is performed in serial adder. Carry out and sum is the two outputs which are also single bit. Addition is executed by adding each bit from least significant bit to most significant bit and each bit has one clock cycle.

Full adder and one flip-flop are used in serial binary adder. Carry out signal for every clock cycle is given into flip flop. Thus the flip-flop generates the output as carry-in signal for next clock cycle. After completion of all bits of input operand, all bits of sum come from sum output.

1. The serial adder contains two binary digits along with carry bit from previous addition. Pulse on clock input triggers the every addition.
2. By using carry reset pin R, carry bit from addition at previous clock pulse is set to zero.
3. The output can be complement of sum in some serial adders, it is optional.

Serial adder is a digital circuit which is a sequential circuit which contains full adder and flip-flop. At every clock cycle, previous bit addition result is taken and flip-flop stores the carry from full adder. Sum result is calculated and carry is given to flip-flop for next calculation. In this way, input data is given to full adder in serially, reading of results in serially, this is synchronized by clock.

BCD (binary coded decimal) adder is a digital circuit in which two BCD numbers are added in parallel and carry out and sum bits are generated. The result of sum will not be in BCD form when addition of two BCD digits is done.

The BCD result is correct in first example and BCD result is not correct in second example. BCD digits are represented from 0 to 9, to represent BCD numbers, four bits are required. But by using four bits, 16 values are represented. In the BCD digits extra six values are ignored because BCD digits are represented from 0 to 9. After addition, the result will not in BCD form when the result is greater than 9. It contains corrections to be done to obtain correct BCD results.

Half adder is used in digital electronics for the purpose of addition of two binary numbers. Full adder is used for the addition of 3-bit input sequence. If input sequence contain more number of bits, half adder and full adder does not satisfy the addition operation. These drawbacks are overcome by Ripple carry adder. For the addition of N-bit numbers, this type of adder circuit is used in digital operations.

[illegible]

PRINCIPAL/DIRECTOR
VIKAS GROUP OF INSTITUTIONS
NUNNA - 521 212
Vijayawada Rural, NTR Dist., A.P.

By using multiple full adders, logical circuit can be created for the purpose of addition of N-bit numbers. In each full adder, one of the sources of info is Cin which is taken care of from Cout of past adder. This kind of adder is known as Ripple-Carry Adder. Since each convey bit waves to next full adder. May be half adder is put in the principal full adder since first full adder contains Cin=0.

The Ripple carry adder circuit chart is straight forward, it creates quick plan time. In any case, the activity of wave convey adder is moderate on the grounds that each full adder ought to be hanging tight for convey bit which is originated from past full adder. By utilizing full adder circuit, door postponement can be determined. Three degrees of rationale are required in full adder. N full adders are required in n-bit Ripple convey adder.

III. PROPOSED SYSTEM

The below figure (1) shows the block diagram of proposed system. Hybrid adder performed in three stages, they are propagator and generator stage, carry generation stage and final propagation stage. In this propagator and generator signals are generated by using propagation stage. Internal Carry generation stage will generate the carry. Final sum stage will take the output of hybrid adder and performs the operation and gives sum as output.

Three stages are presented in the ER based Hybrid adder is explained in detail manner.

1. Propagator and Generator Stage:

In this stage, propagate signals and generate signals are manipulated to pair of each inputs A and B. Propagate signal and generate signal are represented as

Pi = Ai XOR Bi

Gi = Ai AND Bi

2. Internal Carry Generation Stage:

In carry generation stage the calculation is performed based on the bits and carries obtained. The entire operation is performed in the form of parallel. Generate and propagate signals are obtained from the intermediate signals.

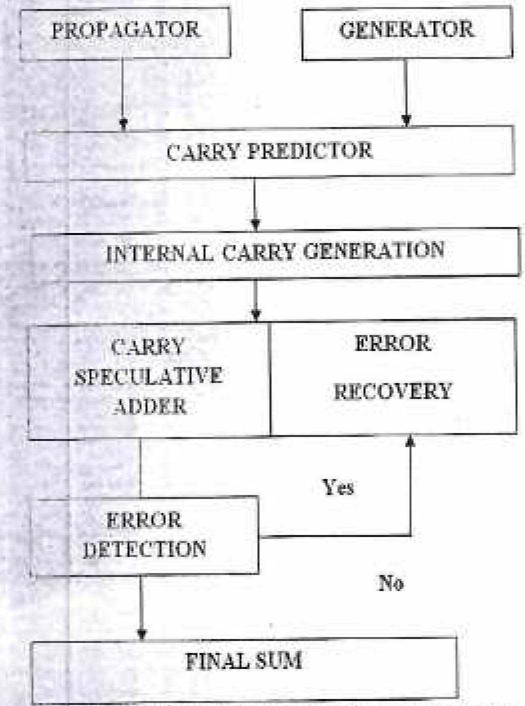


Fig. 1: BLOCK DIAGRAM OF PROPOSED SYSTEM

The below equations shows the propagate and generate signals:

Pi:j = Pi:k AND Pk-1:j

Gi:j = Gi:k OR (Pi:k AND Gk-1:j)

Black/gray cells implement the given two equations, which will be usually used in the following discussion on prefix trees.

3. Final Sum Stage:

In post processing stage the calculation is performed based on the input bits. From post processing stage sum and carry is generated. The below equations shows the sum and carry equations:

Ci = (Pi AND Cin) OR Gi

Si = Pi XOR Ci-1

In applications of high speed circuits, very useful adder is PPA. PPA is designed based on the power and area.

[illegible]

PRINCIPAL/DIRECTOR
VIKAS GROUP OF INSTITUTIONS
NUNNA - 521 212
Vijayawada Rural, NTR Dist., A.P.

Structure delay= $\log_2 n$

Number of computation nodes= $[(n)(\log_2 n)-n+1]$

IV. RESULTS

The below figure (2) shows the RTL schematic of proposed system. here a and b are the inputs and cout is the output.

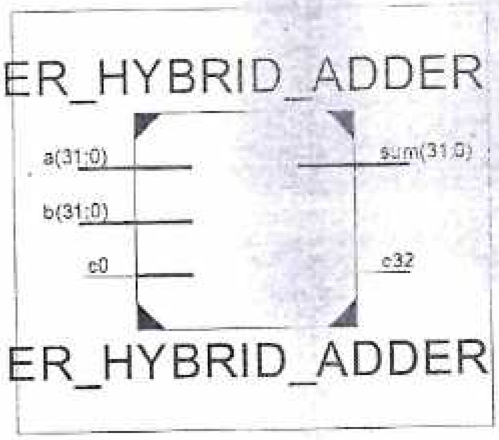


Fig. 2: RTL SCHEMATIC OF PROPOSED SYSTEM

The below figure (3) Technology schematic of proposed system. RTL schematic is the combination of Look up tables, truth tables, K-MAP and equation.

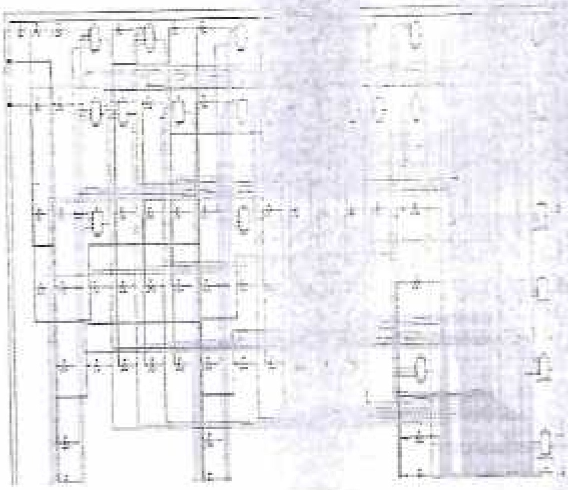


Fig. 3: TECHNOLOGY SCHEMATIC OF PROPOSED SYSTEM

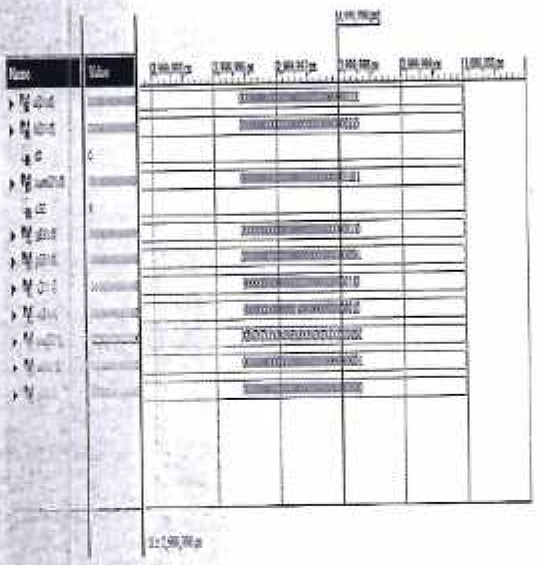
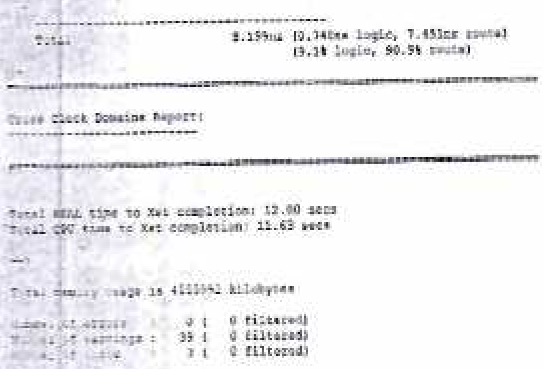


Fig. 4: OUTPUT WAVEFORM OF PROPOSED SYSTEM



```
   ----------------------------------------
   Total                    8.199ns (0.746ns logic, 7.453ns route)
                                    (9.1% logic, 90.9% route)

=========================================================================

Cross Clock Domains Report:
--------------------------

=========================================================================


Total REAL time to Xst completion: 12.00 secs
Total CPU time to Xst completion: 11.63 secs

-->

Total memory usage is [illegible] kilobytes

Number of errors   :    0 (   0 filtered)
Number of warnings :   39 (   0 filtered)
Number of infos    :    3 (   0 filtered)
```

Fig. 5: SYNTHESIS REPORT OF PROPOSED SYSTEM

V. CONCLUSION

Hence in this paper Design a high speed efficient VLSI architecture of E.R Hybrid adder was implemented. The internal carry generation stage plays important role in entire operation. Generate and propagate signals are generated to increase the speed of operation. This design is simulated in Xilinx software. Hence from simulation results, it can observe that Hybrid adder gives effective result.

[illegible]

PRINCIPAL/DIRECTOR
VIKAS GROUP OF INSTITUTIONS
NUNNA - 521 212
Vijayawada Rural, NTR Dist., A.P.

VI. REFERENCES

[1] Vishwa Shah, Urvisha Fata, Jagruti Makwana, " Design and Performance Analysis of 32 Bit VLSI Hybrid adder", Proceedings of the Third International Conference on Trends in Electronics and Informatics (ICOEI 2019) IEEE Xplore Part Number: CFP19J32-ART; ISBN: 978-1-5386-9439-8.

[2] R. Zimmermann, "Binary adder architectures for cell-based VLSI and their synthesis," Ph.D. thesis, Swiss Federal Institute of Technology, (ETH) Zurich, Zurich, Switzerland, 2017, Hartung-Gorre Verlag.

[3] R. P. Brent and H. T. Kung, "A regular layout for parallel adders," *IEEE Trans. Comput.*, vol. C-31, no. 3, pp. 260–264, Mar. 2014.

[4] P. M. Kogge and H. S. Stone, "A parallel algorithm for the efficient solution of a general class of recurrence equations," *IEEE Trans.Comput.*, vol. C-22, no. 8, pp. 786–793, Aug. 2007.

[5] J. Sklansky, "Conditional-sum addition logic," *IRE Trans. Electron. Comput.*, vol. EC-9, pp. 226–231, Jun. 2006.

[6] T. Han and D. A. Carlson, "Fast area-efficient VLSI adders," in *Proc. IEEE 8th Symp. Comput. Arith. (ARITH)*, May 18–21, 2004, pp. 49–56.

[7] R. E. Ladner and M. J. Fischer, "Parallel prefix computation," *J. ACM*, vol. 27, no. 4, pp. 831–838, Oct. 2003

[8] Bernd Becker, "Efficient testing of optimal time adders," Proc. IEEE, Vol. 37, pp. 1113-1120, Sept. 2002.

[9] I. Flores, "The Logic of Computer Arithmetic," Chaps. 4, 5 and 7, Englewood Cliffs, N.J, Prentice Hall, 2002.

[10] O.L. Macsorley, "High-Speed Arithmetic in Binary Computers," Proc. IRE, Vol. 49, pp. 67-91, Jan. 2001.

[11] J. Slansky, "Conditiona1-Sum Additional Logic," Proc. IRE Trans. Electronic Computers, Vol. 9, pp. 226-231, June 2000.

[12] J.B. Gosling, "Conditional-Sum Early Completion Logic," IRE Trans. Electronic Computers, Vol. 59, 2000, pp. 226-231.

[13] N.M. Martin and S.P. Hunagel, "Conditional-Sum Early Completion Adder Logic," IEEE Trans. Vol. c-29, pp. 753-756, Aug. 2000.

[14] Z. Navabi and F.H. Hill, "User manual for AHPL Simulator," Tucson, Arizona, University of Arizona.

[15] F.J. Hill, and G.R. Peterson, "Digital Systems," Chap. 14, N.Y, John Wiley & Sons, Inc, 2000.

[1]VALLURU PRAGATHI Completed B.Tech from Vikas college of engineering and technology, Vijayawada, Andhra Pradesh, India and pursuing M.Tech from Vikas group of institutions, Andhra Pradesh, India. Her M.tech specialization is VLSI Design.

[2]G SEKHAR REDDY completed B.Tech at Sana Engineering College-JNTUH, Andhra Pradesh, India. M.tech from Gitam University, Andhra Pradesh, India. Pursuing Ph.D Kalinga University Raipur, India. He has 10 years of teaching experience and working as Assistant professor at Vikas group of institutions, Vijayawada, A.P, India. His area of interest is Low Power VLSI.

[illegible]

PRINCIPAL/DIRECTOR
VIKAS GROUP OF INSTITUTIONS
NUNNA - 521 212
Vijayawada Rural, NTR Dist., A.P.

Implementation of 32*32 Dadda Multiplier

[1]**Rajani** Manukonda, PG Scholar, Department of ECE, Vikas Group of Institutions, JNTUK, Andhra Pradesh, Email id: manukondarajani630@gmail.com
[2]**G Sekhar Reddy,** Assistant Prof, Department of ECE, Vikas Group of Institutions, JNTUK, Andhra Pradesh, Email id: gaddamsekharreddy@gmail.com

ABSTRACT: In this research paper, the technique for the design of a fast multiplier is proposed by using two different techniques. A 4:2 compressor is used for power-efficient row compression in the proposed model, whereas for faster final summation, a carry-select adder has been used. Based on the above technique, a 16-bit Dadda-multiplier is proposed and its performance is analysed by comparing it with the standard 16- bit Dadda-multiplier which normally uses carry-look ahead adder. The time delay of carry-select adder is improved when compared to carry propagate adder. The result depicts that the proposed 16-bit Dadda-multiplier is 4.85% power efficient and 63.4%-time delay is improved. The simulation of the proposed multiplier is carried out using Verilog HDL in Xilinx ISE Design Suite 14.7.

Keywords— ALU, Dadda-multiplier, 4:2 compressor, carry select adder, carry propagate adder, Wallace-multipliers, half adder, full adders

I. INTRODUCTION

Multiplier is one of the most important circuits for designing computers and computing devices. The Dadda technique for partial product reduction is based on the idea of 'avoid use of full adder.' But the use of full adder is more regular in other Wallace tree multipliers. This paper presents a modified Dadda technique based on the idea of 'prefer the use of full adder over the use of half adder.' Only the last stage that is 'three to two reductions' is the exception. This idea used in the modified technique makes it more regular and simple. Hence, based on the idea, it is named as 'Full-Dadda.' The fact behind saying this technique an alternative approach is that this technique results in same number of full adders, half adders, same size of final carry propagation adder and same number of compressors (adders) at each stage as required in the Dadda technique. Therefore the proposed multiplier can be used in place of Dadda multiplier in all its applications. This paper presents a comparative performance analysis of the proposed multiplier with the Dadda multiplier. For this comparison each multiplier with different operand sizes is taken. The main disadvantages of the Dadda multiplier are: (i) it is less regular, (ii) more complex and (iii) it reduces less number of bits at early stages of reductions. The proposed 'Full-Dadda' multiplier is more regular, simple and reduces more number of bits at early stages of summand reduction. In section 2, there is brief overview of literature and in section 3; there are general rules and equations for reduction scheme and number of hardware components respectively. The comparison between multipliers is arranged in five sub-sections under section.

[illegible]

PRINCIPAL/DIRECTOR
VIKAS GROUP OF INSTITUTIONS
NUNNA - 521 212
Vijayawada Rural, NTR Dist., A.P.

II. Literature Survey

A. Some Important points

The parallel multipliers are faster and more fancied [1]. There are many ways and schemes available for multiplication [1], such as Array multiplication scheme, Booth multiplication and Vedic multiplication etc. Among these schemes, the tree multiplication scheme is one of the most popular schemes. [1,2,3,4,5,6]. The most popular multiplier among tree multipliers is the Dadda multiplier [2]. A Reduced Area multiplier is also the best optimized multiplier in terms of Area if interconnects are properly managed [3]. The basic steps involved in a tree multiplication scheme, first used by Wallace [6] in 1963 are:

1) Generation of the partial products [3] (or summands [5, 6] or matrix of partial products [1]).

2) Partial products reduction: reduction in column height by using pseudo adders (there are many parallel addition schemes [8] and partial product reduction schemes [9] implemented by using pseudo additions such as conditional sum addition [7] etc.).

3) Use of final CPA (carry propagation adder): After the last stage of column-height reduction (remaining bits in a column are only two), there is a need of final addition of two rows to obtain final product of multiplied operands. The size and type of final CPA used plays an important role in determining overall performance such as delay, area and power consumption [2, 3, 5].

B. *Dadda Multipliers*

Dadda method for partial product reduction uses only necessary reduction determined by the Wallace table shown in table 1 [2, 3, 4]. It uses an approach to increase the number of half adders and to decrease the number of full adders required for overall reduction of partial products. The detail of Dadda technique is given in the references [2, 3, 4, 5, 9]. The Dadda method keeps as these are all right columns having height less than or equal to the necessary bits required at a stage after reduction. This is repeated at every stage.

Table1: Number of reduction stages for Dadda multiplier

Bits in Multiplier(N)	Number of Stages
3	1
4	2
$5 \leq N \leq 6$	3
$7 \leq N \leq 9$	4
$10 \leq N \leq 13$	5
$14 \leq N \leq 19$	6
$20 \leq N \leq 28$	7
$29 \leq N \leq 42$	8
$43 \leq N \leq 63$	9
$63 \leq N \leq 94$	10

[illegible]

III. PROPOSED METHOD

Two of the most well-known column compression multipliers have been presented by Wallace [5] and Dadda [6]. Both architectures are similar with the difference occurring in the procedure of reduction of the partial products and the size of the final adder. In Wallace's scheme, the partial products are reduced as soon as possible. On the other hand, Dadda's method does minimum reduction necessary at each level and requires thesame number of levels as Wallace multiplier. As a result, final adder in Wallace multiplier is slightly smaller in size as compared to the final adder in Dadda multiplier. The Block diagram of proposed energy efficient column compression multipliers (Wallace/Dadda) is shown in Fig

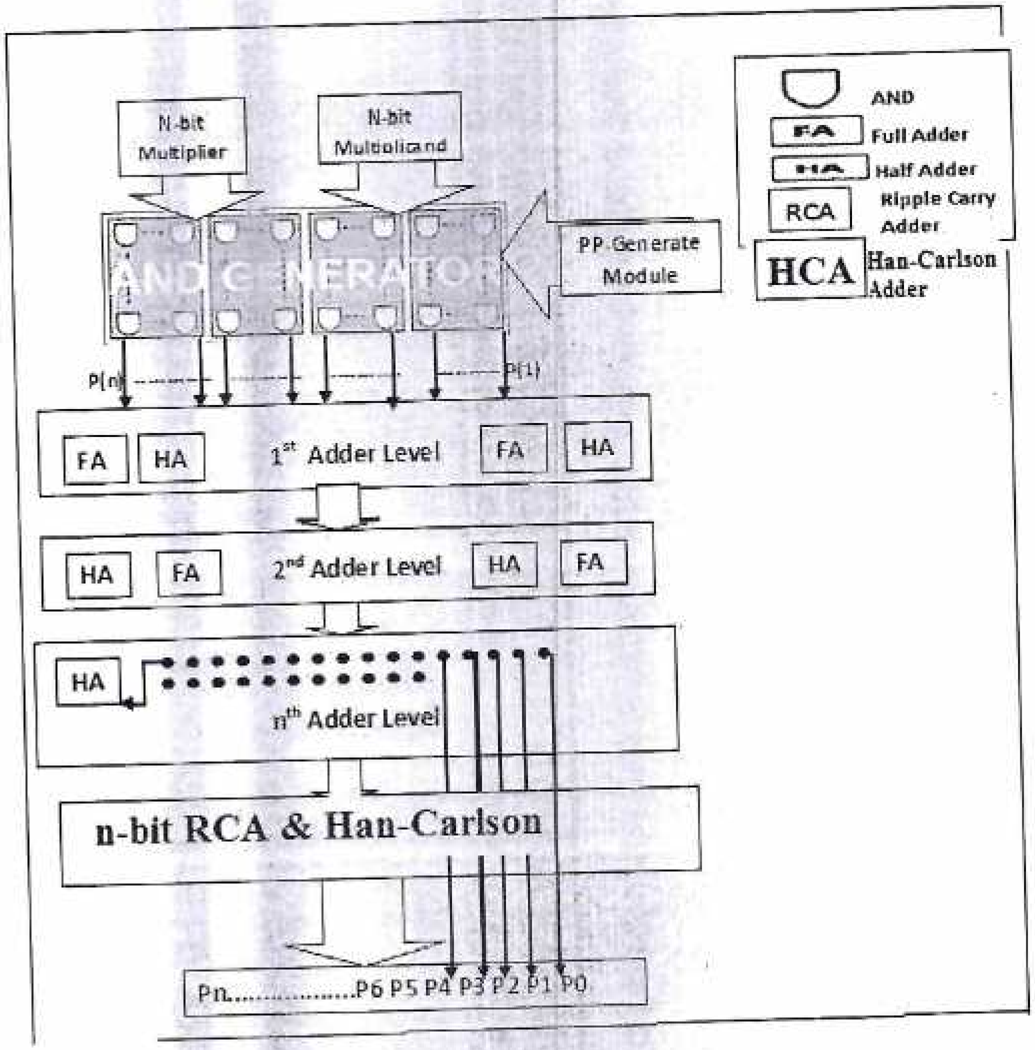


Fig. 1. Block diagram of proposed energy efficient n-bit column compression multiplier

A Wallace/Dadda multiplier is usually composed of threeparts (or modules)

• Partial product generate module

A Half Adder (HA) and Full Adder (FA) tree toreduce the partial products matrix to anaddition of only two Operands.

Han-Carlson and Ripple Carry Adder (RCA) for thefinal computation of the binary result.

In the proposed energy efficient multipliers, the reduction partuses half-adders, full-adders and ripple carry adders; eachpartial product bit is represented by a dot as shown in Fig 3.1.Reduction is performed depending on the number of elementsin that particular column of the group. Three dot products issued to represent a FA, whereas only two dot products is used To represent a HA. If a column has only one element then that ispassed on to the next stage without any reduction. If the lastgroup of a stage contains less than three rows then no reductionis performed on that group. The last part performs the functionto add the remaining two rows using an exact RCA & exactHan-Carlson adder to compute the final binary result in sub thresholdregime. All the standard cells are same in both 8x8column compression Wallace & Dadda multipliers, StaticCMOS logic family is used for AND gate, HA, FA and partialproduct generation module in sub-threshold regime.

ASIC IMPLEMENTATION

The complete ASIC implementation of the proposed design isalso done which follows the cadence design flow. Theproposed design has been developed using Verilog-HDL and synthesized in Encounter RTL compiler using typical libraries of 45 nm technology. The test bench is created for simulation and logic verification by Model-Sim simulator. The CadenceSoC Encounter is used for Placement & Routing (P&R).Parasitic extraction is performed using Encounter Native RC extraction tool. The extracted parasitic RC (SPEF format) is back annotated to Common Timing Engine in Encounter Platform for static timing analysis. Results obtained from both tools are analyzed and compared for semi-custom design verification shown.

[illegible]

PRINCIPAL/DIRECTOR
VIKAS GROUP OF INSTITUTIONS
NUNNA - 521 212
Vijayawada Rural, NTR Dist., A.P.

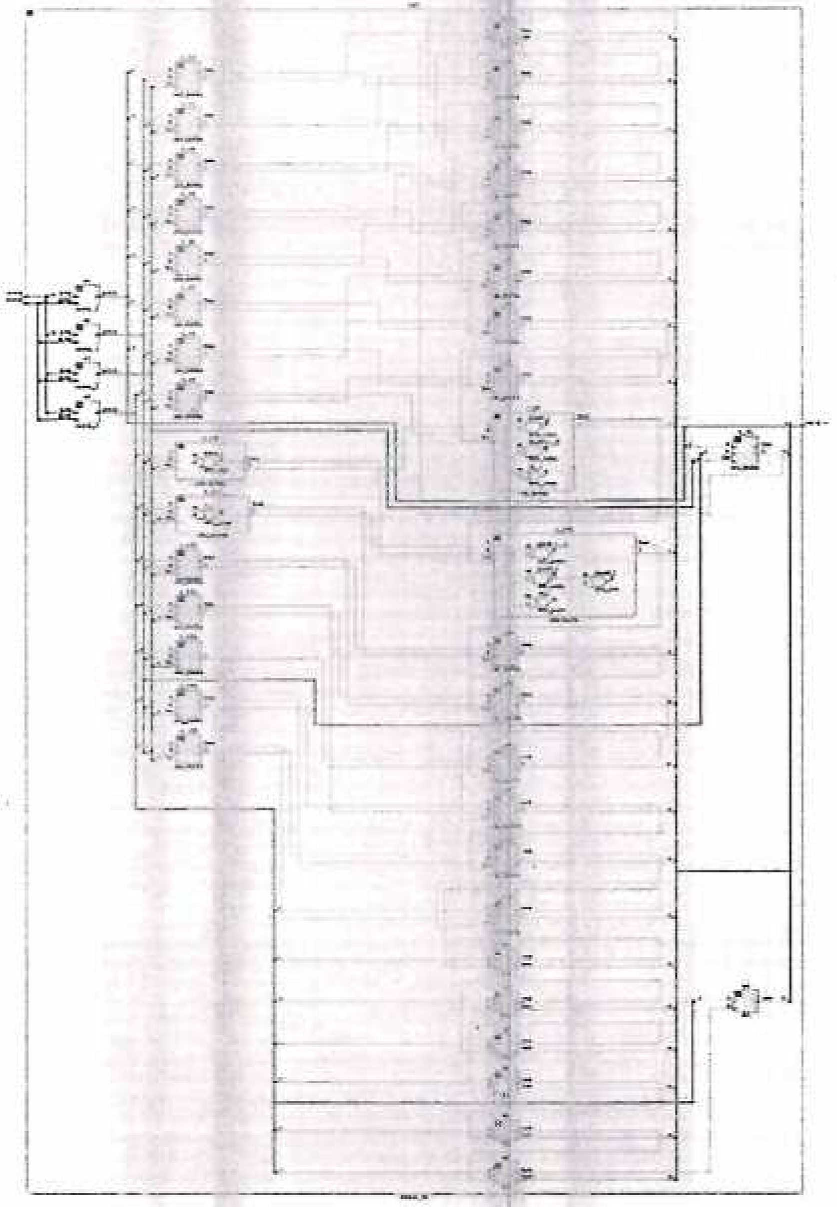


Fig 2 RTL Schematic 32*32 Dadda Multiplier

[illegible]

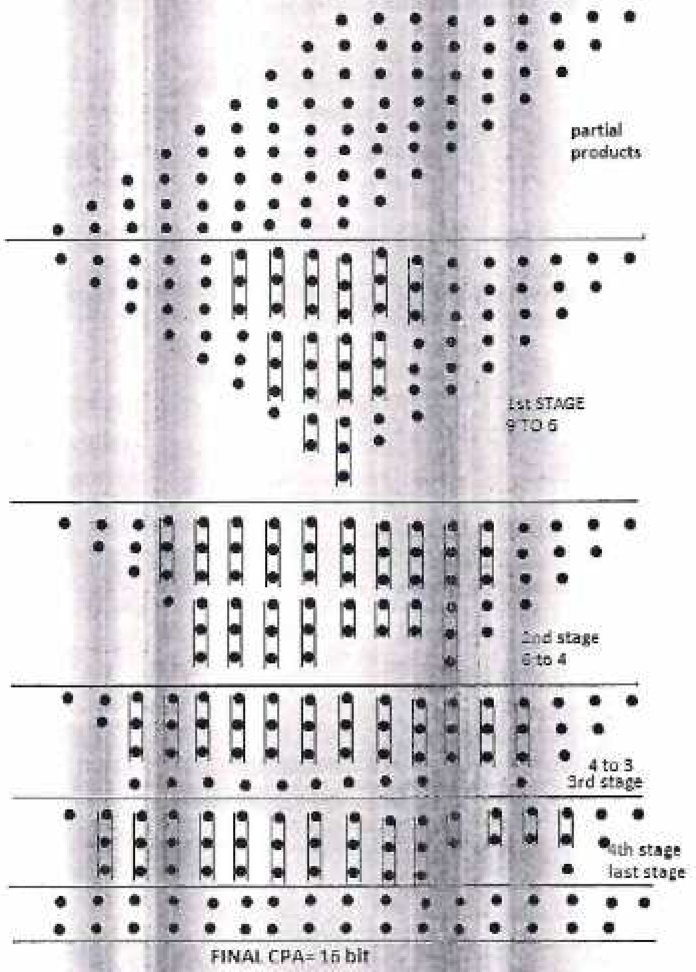


Fig 3: Proposed Multiplier Reduction Scheme

The ASIC Implementation is to be done to differentiate the performance evaluation of the Wallace and Dadda multipliers in sub-threshold and super-threshold regime. The stimulation results were carried out of two 8x8Wallace & Dadda multiplier designs using RCA and two 8x8Wallace & Dadda multiplier design using HCA in sub threshold regime. The designed energy efficient multiplies operates at 0.4V. In fact, in addition to normal transistors, circuits are tested in corner cases with fast and slow transistors and their combinations too. In each stage one of the components FF, SS, FS, and SF is replaced instead of normal transistors in circuit and is pursued in each circuit function. Since the delay of multiplier circuit is proportional to the logarithm of the number of bits in the multiplier and also delay of used standard cells. To measure the critical path delay and to verify the functionality of proposed Wallace/Dadda multipliers, n-no. Of test patterns have been applied to the multiplier. The worst delay has been observed for the inputs from11111111x0000[illegible] for 8x8 Wallace/Dadda multipliers.

[illegible]

PRINCIPAL/DIRECTOR
VIKAS GROUP OF INSTITUTIONS
NUNNA - 521 212
Vijayawada Rural, NTR Dist., A.P.

IV. SIMULATION RESULTS

The proposed design has been developed using Verilog-HDL and synthesized in Xilinx 14.5 tool and the same tool was used for Placement & Routing (P&R).Results obtained from the tool are analyzed and the schematic is shown in the below figure.

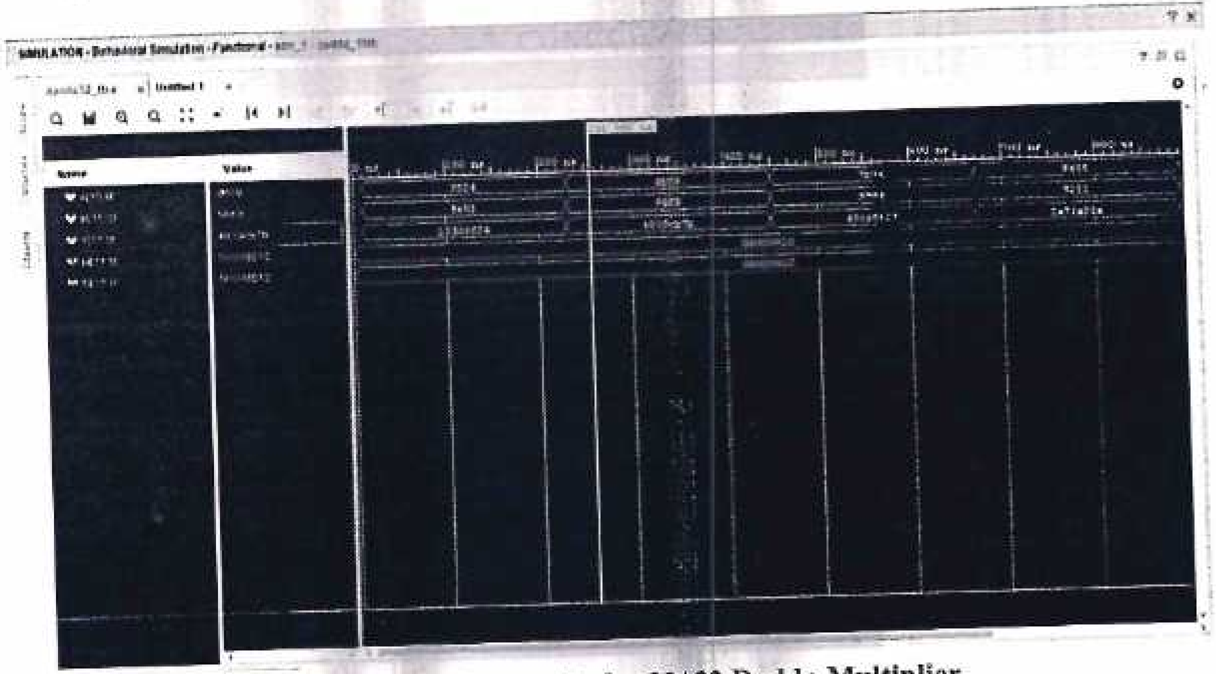


Fig 4 simulation results for 32*32 Dadda Multiplier

V CONCLUSION

In this paper, the 16x16 Dadda multipliers have been implemented at0.4 volt power supply. The main focus of this paper was to optimize overall power delay product. We have also optimized the no. of transistor in the design. The simulated result shows that the proposed Dadda have improved results as compared to existing published results. Thus the design of Dadda multiplier has improved power, performance.

VI. FURTHERSCOPE

As can be seen from the results obtained by Dadda multiplication scheme, this approach is further extended to perform the multiplication of higher bits (i.e., 32 bit × 32 bit, 64 bit × 64 bit and so on). The power consumption and area estimate are further reduced by implementing the final adder with look ahead carry generation logic (look ahead carry adder).

VII. REFERENCE

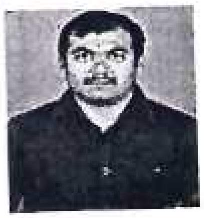
[1]. High-Speed Parallel Multiplication Scheme," IEEE TRANSACTIONS ON COMPUTERS, VOL. C-26, NO. 10, OCTOBER 1977.
[2] R. S. Waters and E. E. Swartzlander, "A reduced complexity wallace multiplier reduction," IEEE Transactions on Computers, vol. 59, no. 8, pp. 1134–1137,

[illegible]

PRINCIPAL/DIRECTOR
VIKAS GROUP OF INSTITUTIONS
NUNNA
Vijayawada Rural, NTR Dist., A.P.

2010.
[3] K. C. Bickerstaff, M. J. Schulte, and E. E. Swartzlander, Jr., "Reduced Area Multipliers," in Proceedings of the 1993 International Conference on Application Specific Amy Processors, pp. 478-489, IEEE, 1993.*The Proposed Full-Dadda Multipliers (IJIRST/ Volume 4 / Issue 11 / 017)*
[4] K'Andrea C. Bickerstaff, Earl E. Swartzlander, Michael J. Schulte, "Analysis of Column Compression Multipliers",15th IEEE Symposium on Computer Arithmetic Proceedings, pp 33-39, 2001.
[5] A. Habibhi and P.A. Wintz "Fast Multipliers," IEEE Transactions on Computers, Vol. C-19, pp. 153-157, 1969
[6] C. S.Wallace, "A suggestion for a fast multiplier," IEEE Transactions on Electronic Computers, vol. EC-13, no. 1, pp. 14–17, 1964.
[7] J. Sklansky, "Conditional-sum addition logic," IRE Transactions on Electronic Computers, vol. EC-9, pp. 226–231, 1960
[8] R. P. Brent and H. T. Kung, "A regular layout for parallel adders," IEEE Transactions on Computers, vol. C-31, no. 3, pp. 260–264,
[9] Wesley Chu, Ali I. Unwala, Pohan Wu, and Earl E. Swartzlander, Jr., "Implementation of a High Speed Multiplier Using Carry Lookahead Adders," 978-1- 4799-2390-8/13, 2013, IEEE. K'Andrea C. Bickerstaff, Earl E. Swartzlander, Michael J. Schulte, "Analysis of Column Compression Multipliers",15th IEEE Symposium on Computer Arithmetic Proceedings, pp 33-39, 2001.
[10] A. Habibhi and P.A. Wintz "Fast Multipliers," IEEE Transactions on Computers, Vol. C-19, pp. 153-157, 1969
[11] C. S.Wallace, "A suggestion for a fast multiplier," IEEE Transactions on Electronic Computers, vol. EC-13, no. 1, pp. 14–17, 1964.
[12] J. Sklansky, "Conditional-sum addition logic," IRE Transactions on Electronic Computers, vol. EC-9, pp. 226–231, 1960
[13] R. P. Brent and H. T. Kung, "A regular layout for parallel adders," IEEE Transactions on Computers, vol. C-31, no. 3, pp. 260–264,
[14] Wesley Chu, Ali I. Unwala, Pohan Wu, and Earl E. Swartzlander, Jr., "Implementation of a High Speed Multiplier Using Carry Lookahead Adders," 978-1- 4799-2390-8/13, 2013, IEEE.

Rajani Manukonda, Pg scholar, Vikas Group of Institutions, JNTUK, Andhra Pradesh, Specialization in VLSI Design



G SEKHAR REDDY, M.TECH IN GITAM UNIVERSITY, ASSISTANT PROF, AREA OF INTEREST :LOW POWER VLSI, Vikas Group of Institutions, JNTUK, Andhra Pradesh

[illegible]

DESIGN AND IMPLEMENTATION OF HIGH EFFICIENT DELAY UTILIZATION OF (n/2) PARALLEL MULTIPLIER

[1]SUNKARA SRIDEVI, [2]G.SEKHAR REDDY
[1]M.Tech Scholar, Dept of ECE, Vikas Group of Institutions, Vijayawada, A.P, India
[2]Assistant Professor, Dept of ECE, Vikas Group of Institutions, Vijayawada, A.P, India

ABSTRACT: In this paper, design and implementation of high efficient delay utilization of (n/2) parallel multiplier is done. Basically, multipliers are key arithmetic circuits in many of these applications including digital signal processing (DSP). This n/2 parallel multiplier uses adder that limits its carry propagation. First the alignment of partial products generator will be done. After that partial products takes this registers and generates the propagate and generate signals. After this n/2 × n/2 multiplication operation is performed. Now this bits perform the addition operation and gives the final product. Hence this paper reduces the delay in effective way.

Key Words: VLSI, Digital signal processing (DSP), Partial products, Multiplexer and Adder.

I. INTRODUCTION

Basically, in communication systems like error correction codes and cryptography, finite field is most widely used. Arithmetic operations are performed using the field elements. Two basis are normally used to implement a system that is normal basis and polynomial basis. Normal basis is used to implement the hardware and perform the low cost squaring operations. In the same way, polynomial basis is used to implement the software and in the same way this also performs the low cost squaring operations [1]. Accuracy could be compromised to a defined extent in most of the present-day applications like image recognition and processing. Multiplier is the basic building block of such applications which involve a lot of mathematical processing. This leads to a win-win balancing between the energy consumed by the circuit and the required accuracy.

The energy consumed by any system is directly proportional to the multiplication accuracy of those systems. If a system requires high accuracy then it consumes more energy and vice versa. Also, there could be section or module of that systems which needs lesser accuracy than other parts of the system. If the accuracy is kept constant across all such modules it greatly increases the amount of energy consumed by the overall system. However, if the accuracy of the multiplier is characterized to change as per the need of that particular module or section of the entire system, this would have a great impact in reducing the amount of energy consumed by the system [2].

This method of configuring and adjusting the accuracy of a multiplier based on the requirement of the system or application is achieved using different adder sub module of the multiplier module to characterize the accuracy based on the approximation technique. There should be reconfigurable multipliers in various program stages or applications [3]. So, in this paper we designed a multiplier which has an accuracy decided on the go based on the requirement of the application.

Montgomery's multiplier is classified into three types, they are bit-serial, bit-parallel, and digit serial architectures. Bit-parallel shape is rapid; however it's far steeply-priced in phrases of vicinity. Bit-serial structure is region efficient, but it's far too sluggish for plenty packages. The digit-serial structure is flexible which may change the space and velocity, consequently, it

[illegible]

PRINCIPAL/DIRECTOR
VIKAS GROUP OF INSTITUTIONS
NUNNA - 521 212
Vijayawada Rural, NTR Dist., A.P.

ISSN NO: 0022-1945

achieves a moderate pace, reasonable price of implementation and hence it is most appropriate for practical use. Montgomery presented a technique for figuring modular multiplication productively. He introduced to move the portrayal of numbers from the Zn to an alternate area, called Montgomery Residual portrayal or Montgomery Domain [6-7].

Here for the purpose of security, the computers and communication system brought with a demand from private sector [4]. The Montgomery multiplication is the calculation that permits effectively for registering. The expense of the particular duplication is equivalent to three whole numbers which increases in addition to the expense of the change in the Montgomery area. Yet, in the event that the large scale task is an exponentiation, at point the change cost is insignificant contrasted with the quantity of augmentations executed in the Montgomery area. In the process of Montgomery multiplication, pre-processing unit and post-processing units are used [5]. The pre-processing unit produces N-Residue operands and in the same way post processing unit will eliminate the constant factor 2n. Hence to form N-Residue operands in the system, modular exponentiation is used.

Here for the purpose of supplanting division activities, shifting tasks are used. After the shifting process the least critical bits will remain zero. Now to eliminate these bits in the modular multiplication, add products are used. After the process of eliminating the bits, the remaining bits are augmented in the multiplicand. Hence from this it can observe that the process of multiplicand is completed. Now the output is obtained after the subtraction of bits. Here if the bits are increased then Montgomery bits also increases. At last the multiplicand bits are controlled without the use of subtraction calculation.

II. LUT BASED MULTIPLIER

The below figure (1) shows the block diagram of serial Multiplier. The entire flow chart performs its operation in four stages which are discussed below:

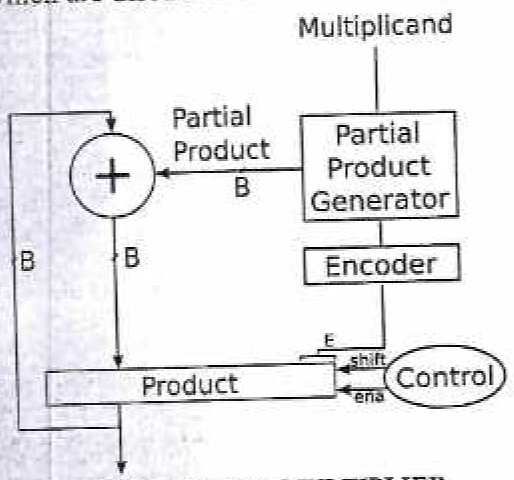


Fig. 1: SERIAL MULTIPLIER

This calculation begins by putting away item esteem in memory where multiplier is 2 and multiplicand is 0, 1, 2, 3, 4, 5, 6, and 7. The Address is the estimation of Multiplicand.

Stage 1: If the Multiplier is 2 and multiplicand is any an incentive from the range 0-7, at that point the item esteem put away at that specific memory area (characterized by multiplicand) is net yield. In any case, if the Multiplier is other than 2 and multiplicand is from 0-7 then re-look in memory to discover if the necessary yield is as of now determined. In the event that indeed, take the yield from that specific location.

Model: how about we guess the multiplier be 2 and multiplicand be 5, at that point 2x5=10 and 10 (10101)2 is as of now put away at memory area 5. In any case, if the multiplier isn't 2 and item esteem is as yet accessible in memory (like 3x4=12 (1100)2

[illegible]
PRINCIPAL/DIRECTOR
VIKAS GROUP OF INSTITUTIONS
NUNNA - 521 212
Vijayawada Rural, NTR Dist., A.P.

previously put away at area 6), at that point essentially get the item esteem at yield put away at that specific area.

Stage 2: If the Multiplier is other than 2 and the worth isn't accessible in memory; search for close by esteem (one less or one more prominent of anticipated item) and flip the last piece. Model: 3x5=15 and this worth isn't accessible in memory, so take the close by esteem 14 (1110)2 and flip the last piece to make 14(1110)2 to 15(1111)2.

Stage 3: If the close by esteem is absent, search for any least factor of the item esteem and annex the bit(s) in the last. For example,4x5=20. Here 10 (1010)2 is minimal factor of 20 (10100)2. So take the yield 10 accessible at memory area 5 and annex zero in the last to get the twofold of 10 that is 20. So also, assume multiplier is 3 and multiplicand is 7. Here the item would be 21(10101)2. Be that as it may, 21 are not accessible in memory. Along these lines, this can be accomplished by taking 10 (1010)2 at the yield and afterward annexing 1 in the last. This will change the 10 (1010)2 in to 21(10101)2.

Stage 4: If above advances doesn't give the necessary yield, attach two bits in the last to get the necessary information. For instance, if there should be an occurrence of 7x5 coming about item esteem is 35 (100011)2 and when we annex (11)2in the last to 8 (1000)2, we can without much of a stretch get the yield 35(100011).

III. PROPOSED SYSTEM

The below figure (2) shows the block diagram of serial multiplier. The entire system is divided into following modules. The following modules are inputs A and B, partial products, n/2 multiplier, multiplexer. adder and final product. This operation is mainly used in the coprocessor in parallel format. This system provides the functionality for coprocessor. First the alignment of partial products generator will be done. After that partial products takes this registers and generates the propagate and generate signals. After this n/2 × n/2 multiplication operation is performed. Now this bits perform the addition operation and gives the final product.

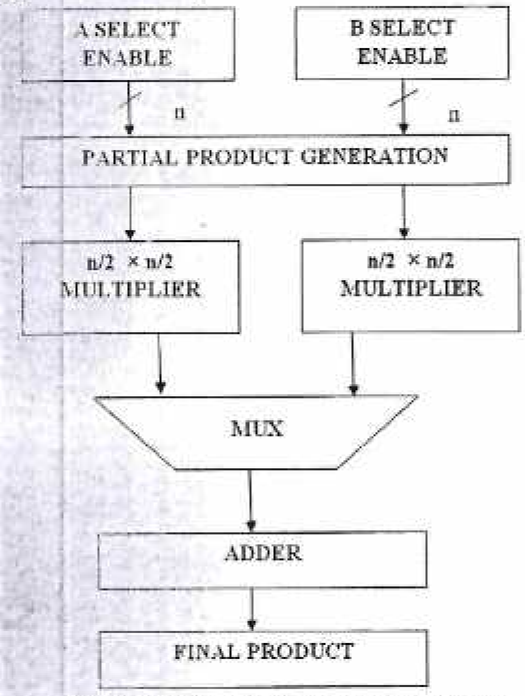


Fig. 2: BLOCK DIAGRAM OF SERIAL MULTIPLIER

Here firstly, the operands are loaded in the multiplier. The arithmetic operations like addition and multiplication operations are performed. The obtained result of this will be saved in the barrel shifter. Here irreducible polynomial function is not used in the system. The main intent of register multiplier is to store the bit representation and give polynomial output a(t). Here parallel load operation is performed in the most significant bit position. In the same way left shift operations are performed in MSB bit.

The B select enable is used b(t) value to store the value in register. The parallel load operation is also applied in the multiplicand.

[illegible]

The obtained value is stored in the register. The right shift operation is performed in the multiplicand register block. crypto core processor is used to transfer the data in multiplicand register.

The result A select enable and B select enable is processed to the partial product generator block. The both A select enable and B select enable values are assigned in the barrel shifter blocks. The obtained values in the propagator and generator block will generate the signals. This block will perform the addition operation.

An adder is a digital circuit that performs addition of numbers. A multiplexer or mux is a combinational circuits that selects several analog or digital input signals and forwards the selected input into a single output line.

Final product is the combination of output. Where all bits gets mixed to get the final output. Hence from results it can observe that the proposed system, will reduce the delay and memory usage in very effective way.

IV. RESULTS

The below figure (3) shows the RTL schematic of proposed system. here a and b are the inputs and cout is the output.

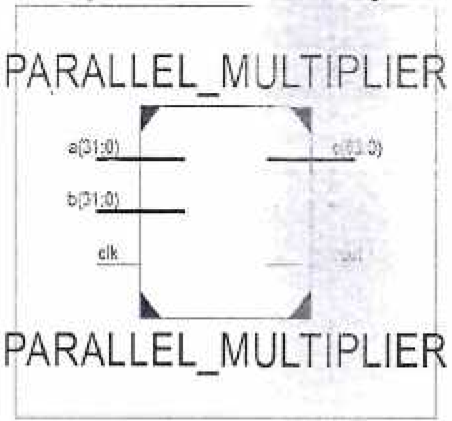


Fig. 3: RTL SCHEMATIC OF PROPOSED SYSTEM

The below figure (4) Technology schematic of proposed system. RTL schematic is the combination of Look up tables, truth tables, K-MAP and equation.

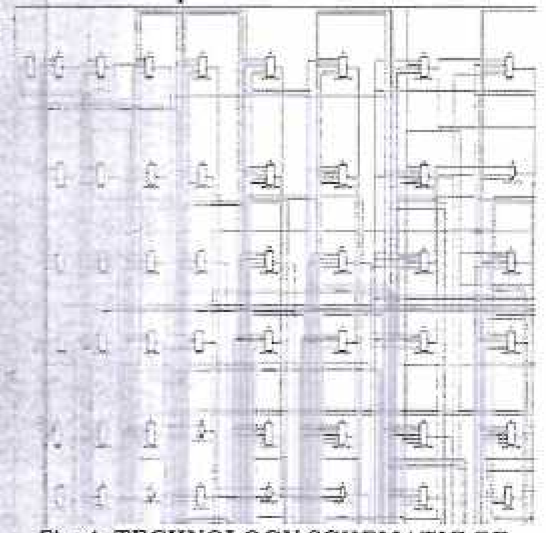


Fig. 4: TECHNOLOGY SCHEMATIC OF PROPOSED SYSTEM

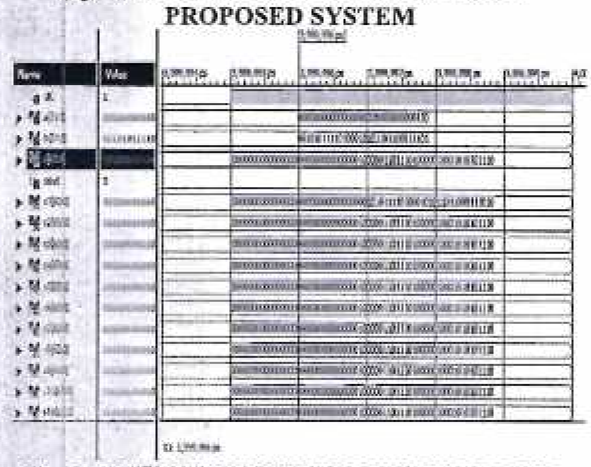
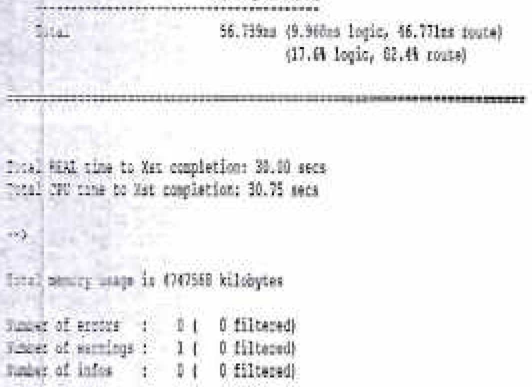


Fig. 5: OUTPUT WAVEFORM OF PROPOSED SYSTEM



```
---------------------------------------------
   Total                   56.739ns (9.968ns logic, 46.771ns route)
                                    (17.6% logic, 82.4% route)

-----------------------------------------------------------------------


Total REAL time to Xst completion: 30.00 secs
Total CPU time to Xst completion: 30.75 secs

-->

Total memory usage is 4747568 kilobytes

Number of errors   :    0 (   0 filtered)
Number of warnings :    1 (   0 filtered)
Number of infos    :    0 (   0 filtered)
```

Fig. 6: SYNTHESIS REPORT OF PROPOSED SYSTEM

[illegible]

PRINCIPAL/DIRECTOR
VIKAS GROUP OF INSTITUTIONS
NUNNA - 521 212
Vijayawada Rural, NTR Dist., A.P.

V. CONCLUSION

Hence in this paper design and implementation of high efficient delay utilization of (n/2) parallel multiplier was done. The (n/2) parallel multiplier performs the multiplication in very fast. The proposed system will reduce the switching activities that are produce in the system. To reduce the delay partial product unit is introduced. Hence the memory of complexity is reduced in (n/2) parallel multiplier. This system is mainly used in the applications of low delay and high speed applications.

VI. REFERENCES

[1] Tayab D Memon, Aneela Pathan, "An Approach to LUT Based Multiplier for Short Word Length DSP Systems", 978-1-5386-5689-1/18/$31.00 ©2018 IEEE.

[2] "Power-delay-area efficient design of vedic multiplier using adaptable manchester carry chain adder", Raghava Katreepalli , Themistoklis Haniotakis, 2017 International Conference on Communication and Signal Processing (ICCSP).

[3] "Low power array multiplier using modified full adder", S. Srikanth , I. Thahira Banu , G. Vishnu Priya , G. Usha, 2016 IEEE International Conference on Engineering and Technology (ICETECH).

[4] "Design of high speed multiplier using modified booth algorithm with hybrid carry look-ahead adder" R Balakumaran , E Prabhu, 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT).

[5] "Design of area and delay efficient Vedic multiplier using Carry Select Adder", G. R. Gokhale , S. R. Gokhale, 2015 International Conference on Information Processing (ICIP).

[6] "Design of ultra low power multipliers using hybrid adders", Thottempudi Pardhu , N.Alekhya Reddy, 2015 International Conference on Communications and Signal Processing (ICCSP).

[7] "Comparative study of performance vedic multiplier on the basis of adders used", Josmin Thomas , R. Pushpangadan , S Jinesh, 2015 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE).

[8] "Design of area efficient and low power multipliers using multiplexer based full adder" S. Murugeswari , S. Kaja Mohideen, Second International Conference on Current Trends In Engineering and Technology - ICCTET 2014.

[9] "A vertical-MOSFET-based digital core circuit for high-speed low-power vector matching", Yitao Ma , Tetsuo Endoh , Tadashi Shibata, 2011 International SoC Design Conference.

[10] H. Hinkelmann, P. Zipf, J. Li, G. Liu, and M. Glesner, "On the design of reconfigurable multipliers for integer and Galois field multiplication," Microprocessors Microsyst., vol. 33, no. 1, pp. 2–12, Feb. 2009.

[1]SUNKARA SRIDEVI Completed B.Tech from Usha Rama College Of Engineering and Technology, Andhra Pradesh, India and pursuing M.Tech from Vikas Group of Institutions, Andhra Pradesh, India. Her M.tech specialization is VLSI Design.

[2]G SEKHAR REDDY completed B.Tech at Sana Engineering College-JNTUH, Andhra Pradesh, India. M.tech from Gitam University, Andhra Pradesh, India. Pursuing Ph.D Kalinga University Raipur, India. He has 10 years of teaching experience and working as Assistant professor at Vikas group of institutions, Vijayawada, A.P, India. His area of interest is Low Power VLSI.

[illegible]

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 12, pp: 336-346 www.ijaem.net ISSN: 2395-5252

Design & Implementation of Triple Bit Error Correcting Bch Decoder with High Efficiency for Emerging Memories

D Naga Bhavani[1], G Sekhar Reddy[2], B Naveen Kumar[3]
[1]P.G Scholar, VLSI Design, ECE, Vikas Group of Institutions,Nunna.
[2]Asst. Professor. ECE., Vikas Group of Institutions,Nunna.
[3]Asst. Professor. ECE., Vikas College of Engineering & Technology,Nunna.

Date of Submission: 20-11-2021 Date of Acceptance: 05-12-2021

ABSTRACT: As the technology scales down emerging memories struggling with reduced reliability. As a solution error-correcting code ECC and its decoder circuits have been applied. To correct two (or) three errors BCH code is widely adopted. We have double- error-correcting and triple error-detecting (DEC-TED) bose-Chaudhuri–Hocquenghem (BCH) code decoder with high decoding efficiency and low power for error correction in emerging memories. Here we are implementing triple bit error correcting (TEC) decoder to increase the decoding efficiency, we propose an adaptive error correction technique for the BCH codethat detects the number of errors in a codeword immediately after syndrome generation and applies a different error correction algorithm depending on the error conditions. With the adaptive error correction technique, the average decoding latency and power consumption are significantly reduced owing to the increased decoding efficiency. To further reduce the power consumption, an invalid-transition-inhibition technique is proposed to remove the invalid transitionscaused by glitches of syndrome vectors in the error- finding block.

Keywords: adaptive error correction, bose-Chaudhuri–Hocquenghem (BCH) code, double error correcting (DEC)and triple error detecting & correcting (TED -TEC), emerging memories, error correcting code(ECC),LUT- based decoders, invalid-transition- inhibition technique

I. INTRODUCTION

Emerging memories, such as phase change memory, spin-transfer torque magneto resistive random access memory (STT-MRAM), phase change RAM (PRAM), andresistive random access memory (ReRAM) have been investigated to fill the gaps in terms of performance and density between DRAM and NAND flash memory, referred to as storage class memories (SCMs). They are of interest for their flexible and efficient memory hierarchy, owing to their nonvolatile, high-density, and low-latency characteristics [1]. In addition to SCMs,
some emerging memories, such as STT-MRAM, are also considered promising candidate embedded memories due to their fast read and write latencies, low leakage power, and logic-friendly compatibility [2], [3]. As technology scales down, these emerging memories arealso struggling with reduced reliability, and as a solution, error-correcting code (ECC) and its encoder/decoder circuits have been applied. While NAND flash requires a powerful ECC capable of correcting up to 100 errors, most of the emerging memories can reach the required chip yield using an ECC capable of correcting two or three errors because of new developments in storage physics [2]–[8]. In addition to simply increasing the memory yield, ECC can be used to optimize memory performance regarding density [9], [10] and energy consumption [11], [12]. In this manner, ECC has become an essential part of emerging memories. To correct two or three errors, the Bose–Chaudhuri–Hocquenghem (BCH) code is widely adopted for emerging memories [2]–[8]. However, the standard iterative and sequential decoding processes, which require multiple cycles, are not compatible with emerging memories. This is because the latency of the BCH code decoder should bea few nanoseconds, considering the short read or write access time in emerging memories. To achieve a double-error-correcting (DEC) BCH code decoder withlatency of a few nanoseconds, a fully parallel decoder structure that uses combinatorial logic gates has beenproposed in [13]–[17]. However, it continues to have 50%–80% latency penalty and consumes 6–8 times more power than the single-error-correcting and double-error detecting (SEC-DED) decoder. As non- or single-bit errors are considerably more likely than multi bit (double-bit or triple bit) errors despite the increased raw bit-error rate (RBER) in nanotechnology, it is inefficient to deal with non- or single bit errors with a DEC-TED decoder in terms

[illegible]
PRINCIPAL/DIRECTOR
VIKAS GROUP OF INSTITUTIONS
NUNNA - 521 212
Vijayawada Rural, NTR Dist., A.P.

of latency and power, which leads to reduced decoding efficiency. Moreover, the fully parallel decoders consume large dynamic power owing to the invalid transitions in the error-finding block. Since most emerging memories have been widely researched for use in low-power applications, such as wearable devices and IOT devices, the power of fully parallel BCH decoders should also be reduced to maximize thebenefits of emerging memories.

In this paper, we propose a high-decoding-efficiency and low-power BCH decoder with DEC and triple-error-detecting (DEC-TED) capability for emerging memories. To reduce the average delay andpower consumption, an adaptive error correction technique for the DEC-TED BCH code is proposed. In addition, an invalid transition inhibition technique using flip-flops (FFs) and a specific ECC clock is applied to reduce the power consumption further. The synthesis results using 65-nm technology show that the proposed DEC-TED BCH decoder with 64-bit data words achieves more than 50% average latency reduction and 70%-75% average power saving in comparison to the conventional decoder with an insignificant area overhead.

The remainder of this paper is organized as follows. In Section II, an overview of BCH codes and fully parallel structure is given. In Section III, the problems in conventional fully parallel BCH decoders are discussed. The proposed decoder with highdecoding efficiency and low power is presented in Section IV. Section V presents the synthesis and comparison results of conventional decoders and the proposed decoder. Finally, Section VI concludes this paper.

II. CODENING THEORY

Background coding theory more detailed accounts of error-correcting codes can be found in: Hill, Pless MacWilliams and Sloane, van Lint, and Assumes and Key. See also Peterson for an early article written from the engineers' point of view. Proofs of all the results quoted here can be found in any of these texts;our summary here follows.

Here a message is first given by the source to the encoder that turns the message into a codeword, i.e. a string of letters from some alphabet, chosen according to the code used. The encoded message is then sent through the channel, where it may be subjected to noise and hence altered. When this message arrives at the decoder belonging to the receiver, it is equated with the most likely codeword, i.e. the one (should that exist) that, in a probabilistic sense depending on the channel, was probably sent, and finally this "most likely" codeword is decoded andthe message is passed on to the receiver.

III. BCH CODES AND FULLY PARALLEL BCH DECODERS FOR EMERGING MEMORIES

In general, a primitive binary BCH code is defined over a binary Galois field with degree m, denoted by GF (2^m). The (n, k, d) BCH code over GF(2^m) is represented as follows [18]:

Codeword length: $n = 2^m - 1$ Number of information bits: $k \geq 2^m - mt - 1$

Minimum distance: $d \geq 2t + 1$.

This code is capable of correcting any combination of t or fewer errors in a block of n digits, called a t-error correcting BCH code. Since the number of information bits is not represented as the power of two, a shortened binary BCH code is applied in a memory system by eliminating information bits (p), such as (n−p, k − p, d).

The RBER of the memory cell varies widely depending on design goals such as memory density, read or write latency, and energy consumption. For emerging memories, RBERs of STT-MRAM, ReRAM, and PRAM are distributed with a range of 10−10–10−3 [5]– [18], [19]–[21]. These RBER can be reduced by appropriate device, circuit, and architecture design techniques [5], [6], [8], [20]. When it is lower than 10−5, the target block failure rate (BFR) can be achieved with an ECC capable of correcting two errors[2]–[8]. If TED option is added to DEC, the BFR can be improved further. Thus, DECTED BCH code is adoptedin this paper, and the following decoding processes are described based on the primitive binary DET-TED BCHcode [22], [23].

1) Computing Syndrome: For (n, k, 6) DEC-TED BCH code, the parity-check matrix H is given by

$$H = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \alpha^3 & \alpha^6 & \cdots & \alpha^{3(n-1)} \end{bmatrix} = \begin{bmatrix} 1 \\ H_1 \\ H_3 \end{bmatrix}$$

where α is the primitive element in GF(2m).

PRINCIPAL/DIRECTOR
VIKAS GROUP OF INSTITUTIONS
NUNNA - 521 212
Vijayawada Rural, NTR Dist., A.P.

TABLE I
RELATION SHIP BETWEEN THE NUMBER OF ERRORS AND SYNDROME VECTORS FOR THE BCHDEC-TED CODE

	Syndrome Condition	
	S_0	S_1 and S_3
Non	0	$S_1 = S_3 = 0$
Single-bit	1	$S_1^3 = S_3$
Double-bit	0	$S_1^3 \neq S_3$
Triple-bit	1	$S_1^3 \neq S_3$

To determine whether the received codeword, v, has errors, syndrome vector S is calculated as
$\mathbf{S = v \cdot H^T = v \cdot 1^T, v \cdot H^T 1 , v \cdot H^T 3} = [S0, S1, S3]$

where S0 is a 1-bit vector, and S1 and S3 are m-bit vectors for the code generated in GF(2m). A single-bit error can be corrected using only the S1 vector because H1 can be used as the parity-check matrix for the Hamming code. For a double bit error correction, S1 and S3 vectors are utilized together. It is worth noticing that the syndrome vector can be used to detect the number of errors in the received codeword using the specific relationships among S0, S1, and S3 vectors, as shown in Table I. This specific relationship is applied in the proposed decoder, as will be explained in Section IV

2) Determining the Error Location Polynomial: The next decoding step is to complete the error location polynomial (ELP) based on the calculated syndrome vectors. For a DEC-TED BCH code, the ELP can be represented by
$\sigma(x) = 1 + \sigma 1x + \sigma 2x\ 2$
Notice that each coefficient of the ELP is an m-bit vector if the codeword is constructed on GF(2m). Conventionally, the Berlekamp–Massey (BM) [24] algorithm is widely applied to compute the coefficientsof the ELP.

3) Finding the Error Locations: After computing the coefficients ($\sigma 1$ and $\sigma 2$), the Chien search is performedto find the roots of the ELP by substituting n elementsof GF(2m), $\{\alpha^0, \alpha^1, ..., \alpha^{n-1}\}$, into (3).

4) Correcting Errors: Through step 3, an error vector, e, is obtained, and a corrected codeword, v*, can be represented as v* = v + e. This can be implemented using XOR gates.

B. Fully Parallel BCH Decoders for Emerging Memories The long BCH code is already adopted in NAND flash memories to correct tens of errors in thousands of data bits [25]–[27]. For long BCH codes, conventional iterative BCH decoding algorithms are applied, and the decoder is usually implemented by the linear feedback shift register, which takes 2n + 2t cycles to finish the error correction. However, this decoding algorithm is not compatible with low latency emerging memories, so a fully parallel decoding architecture has been employed to achieve a decoding latency of a few nanoseconds [13]–[17]. The fully parallel decoding architecture is fully parallelized and implemented using a combinatorial logic, which can significantly reduce the decoding latency at the expense of hardware overhead. However, the hardware overhead caused by the fully parallelized

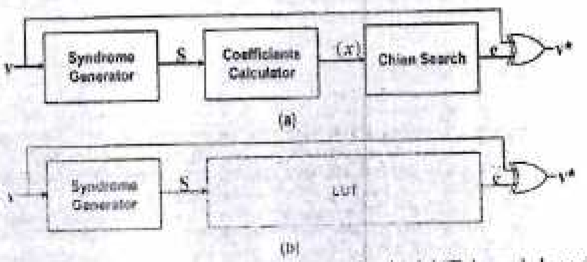


Fig.1.Block diagram a) PA-based decoder b) LUT-based decoder

[illegible]
PRINCIPAL/DIRECTOR
VIKAS GROUP OF INSTITUTIONS
NUNNA - 521 212
Vijayawada Rural, NTR Dist., A.P.

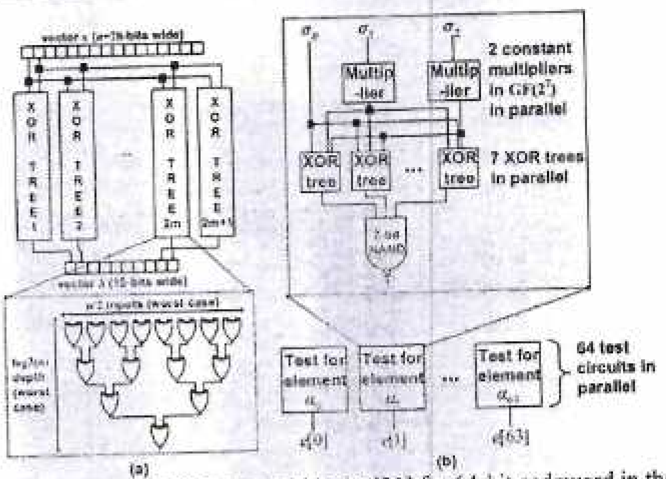


Fig. 2.(a) Syndrome generator and (b) Chien searchblocks [31] for 64-bit codeword in the fully parallelDEC-TED BCH decoder.

Implementation is not significant in emerging memories. This is because a short BCH code [28]–[30]can be used owing to a low required error-correcting capability and its relatively low size of memory array compared to the NAND flash memories [2]–[8].

In the fully parallel structure, the syndrome vector S can be obtained by separate XOR trees with inputs taken from the received code vector, as shown in Fig. 2(a) [31]. According to the decoding algorithm and implementation methods in determining ELP and finding the roots, fully parallel BCH decoders can be divided into two categories, Peterson's algorithm (PA)-and lookup table (LUT)-based decoders. 1) Peterson's Algorithm-Based Decoder: As an alternative to the BMalgorithms, PA was proposed in [32] to eliminate the time-consuming iterations. By applying PA, the ELP of DEC-TED BCH code is given by

$$\sigma(x) = 1 + \sigma_1 x + \sigma_2 x^2 = 1 + S_1 x + \left(S_1^2 + \frac{S_3}{S_1}\right) x^2.$$

However, complex finite-field dividers are required tocompute the coefficients. Thus, a reverse ELP (RELP) was proposed in [16] to alleviate the complexity of coefficient evaluation and computation during the Chien search, and it is expressed as

$$\bar{\sigma}(x) = \bar{\sigma}_0 + \bar{\sigma}_1 x + \bar{\sigma}_2 x^2 = \left(S_1^3 + S_3\right) + S_1^2 x + S_1 x^2.$$

The overall structure of a PA-based decoder is shown in Fig. 1(a). A coefficient calculator determines the bit components of the σ˜0, σ˜1, and σ˜2 vectors in (5), andeach component of the coefficients is obtained by usingthe syndrome vector bit components with only modulo-2 addition and multiplication [14]. In the Chien search block, the computations of σ (α ˜ i) for 0 ≤ i ≤ n
−1 are conducted in parallel using simple logic operations [13].

The test circuit for checking whether σ (α ˜ i) is 0 requires a multiplication by a constant (αi), and it canbe implemented by XOR-trees, as shown in Fig. 2(b) [31].

2) LUT-Based Decoder: In [17], an LUT-based decoder is proposed by replacing the coefficients calculator andChien search blocks in the PA-based decoder with an LUT. The LUT contains all the possible pairs of syndromes and their corresponding error patterns. In this decoder, the error positions can be determined directly from the syndromes after a syndrome vector iscomputed.

3) Comparison of PA-Based and LUT-Based Decoders: In the LUT-based decoder, the error vector can be directly determined immediately after the syndrome vector is computed. Thus, the LUT-based decoder has a shorter decoding latency at the cost of increased areaoverhead in comparison to the PA-based decoder. However, as the number of correctable errors (t) or thenumber of information bits

[illegible]

(k) increases, the table size exponentially increases, resulting in an inefficient realization in both area and delay. In comparison to the LUT based decoder, the increased area of the PA-based decoder with increased t or k is much smaller. Therefore, the PA-based decoder is more appropriate for area-constrained applications. In terms of dynamic power consumption, the PA-based decoder consumes more power than the LUT-based decoder. In the PA-based decoder, the computed syndrome vectors are continuously used in both the coefficient calculator and Chien search blocks until the error vector is determined. Thus, whenever the syndrome vectors are newly computed in response to a newly received codeword, most of the nodes in the blocks following the syndrome generator are toggled, leading to high dynamic power consumption. On the other hand, only one circuit path in the LUT block is activated by the corresponding input syndrome vector due to the inherited LUT characteristic, leading to low dynamic power consumption
Therefore, the LUT-based decoder is favorable in power constrained applications.

IV. PROBLEMS IN CONVENTIONAL FULLYPARALLEL BCH DECODERS

In general, the decoder for DEC-TED BCH codes has longer latency, higher area complexity, and much higher power consumption than the decoder for SEC-DED codes. It should be noted that the PA-based decoder has about four times smaller area than the LUT-based decoder, but it consumes more power correction of all non-, single-, and double-bit errors with the DET-TED decoder is inefficient in terms of latency and power consumption, and this reduces the decoding efficiency. If the proper decoder between SEC-DED and DEC- TED decoders can be adaptively selected depending on the error conditions, decoding latency and power consumption can be significantly reduced on average Dynamic Power Problem in Fully Parallel BCH Decoder:

Most of the previous studies on a fully parallel archi- tecture for the BCH decoder have focused on circuit optimization methods to reduce the latency while minimizing the complexity of implementation. However, considering that the read or write power of emerging memories is generally at the microwatt level, much higher power consumption in conventional fully parallel decoders undermines the low-power advantage of emerging memories.

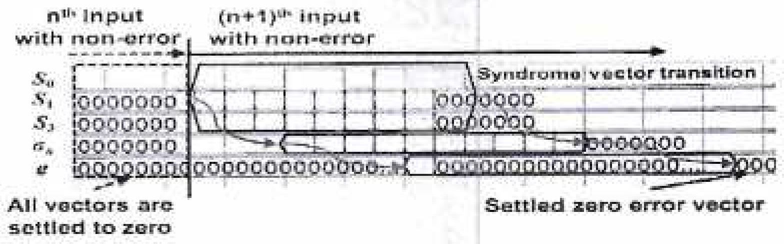


Fig. 4. Invalid transitions in the coefficient calculatorand parallel Chien search blocks due to the transitions on syndrome vectors in the case of consecutive non-error input code words

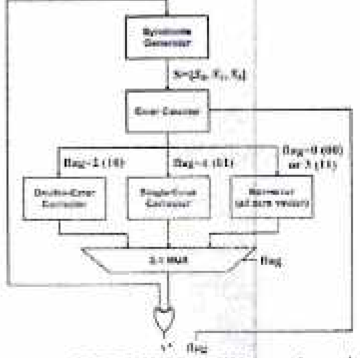


Fig.5. Conceptual block diagram of DEC-TED BCHdecoder with adaptive error correction.

[illegible]
PRINCIPAL/DIRECTOR
VIKAS GROUP OF INSTITUTIONS
NUNNA - 521 212
Vijayawada Rural, NTR Dist., A.P.

The syndrome vector is a key factor in finding errors in both PA- and LUT-based decoders. Whenever a new input is entered into the decoder, the syndrome generator produces invalid glitches before its outputs settle down. The glitches cause undesired transitions at internal nodes in the blocks that follow the syndrome generator, such as the coefficient calculator, the parallel Chien search (LUT), and the error corrector in the PA-based (LUT-based) decoder, and increase dynamic power consumption.

The effect of invalid transition on power is severe, especially when consecutive non-error code words are received. When a non-error codeword is entered into the decoder, most of the key generated vectors, such as syndromes, coefficients of RELP, and error vectors, are settled to 0. If the next non-error codeword is immediately entered, then the syndrome vector toggles several times until it settles down to 0. This causes invalid transitions in other blocks. This invalid transition problem can be prevented if stable syndrome vectors are delivered to subsequent blocks

V. PROPOSED HIGH-DECODING-EFFICIENCY ANDLOW-POWER DEC-TED BCH DECODER

DEC-TED BCH decoder using an adaptive error correction and an invalid transition inhibition technique is proposed to achieve the high decoding efficiency and low-power consumption

FLAG CONDITION FOR THE NUMBER OFERRORS CLASSIFICATION

Number of Errors	2bit flag condition		
	flag [1] =	$(S_1^3 \cdot S_3)$	flag [0] = S_0
Non	0		0
Single-bit	0		1
Double-bit	1		0
Triple-bit	1		1

After syndrome vectors are generated, the number oferrors caused in the received codeword is classified in an error counter block, and a 2-bit flag signal that represents the number of errors is generated. Then, different error correction algorithms are applied depending on the generated flag signal to improve the decoding efficiency, and a proper error vector is addedto the received codeword through the 3:1 MUX.

The 2-bit flag signal can be generated based on the generated syndrome vectors, as shown in Table III. For odd numbers of errors (single- or triple-bit errors), $S0$ is "1," whereas for non-error and double-bit errors, $S0$ is "0." Multiple-bit error (MBE), a logical OR of all the vector bit components.

Based on the generated flag signal, we can choose between the single-error (SE) corrector and the double-error (DE) corrector. In the proposed design, the SE corrector uses Hamming SEC code and the DE corrector uses the DEC BCH code. Since error correction algorithms are not required regarding non-or triple-error cases (flag "00" or "11"), all zero vectors go directly to the MUX without being processed in most delay and power consuming error correction blocks. Thus, the latency and power consumption can be minimized for non- or triple-bit error cases. Since the most common non-error case has minimum latency and power, the average decoding latency and power consumption can be greatly reduced. When a single-biterror occurs (flag 01), the SE corrector, whichcompares each column of the H1 matrix with the S1 vector, carries out single-bit error correction. Thus, when there is a single-bit error in the received codeword, the proposed decoder has similar latency and slightly larger power consumption in comparison to the conventional SEC-DED code decoder. In the case of double bit errors (flag 10), the DE corrector performs error correction, and the latency and power consumption are similar to those of conventional fullyparallel DEC-TED BCH decoders.

Thus, the delay and power consumption of theDEC-TED BCH decoder with the adaptive error correction varies according to the types of errors in the codeword. To realize the adaptive error correction technique, additional error counter, SE corrector, and MUX blocks are added to the conventional fully parallelDEC-TED decoder.

[illegible]

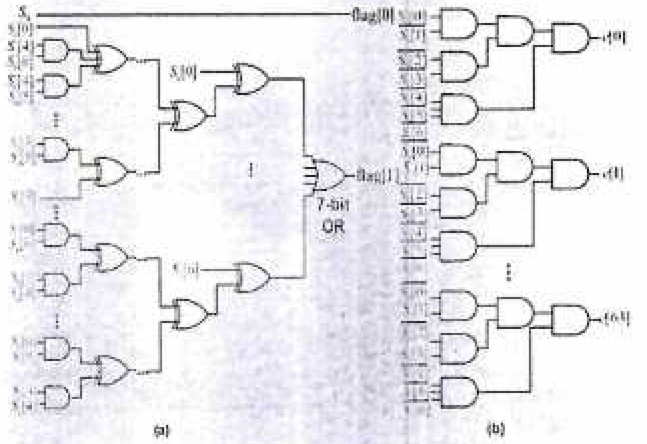


Fig. 6. (a) Error counter and (b) SE corrector blocks in the fully parallel DEC-TED BCH decoder for the 64-bit codeword.

Regarding the PA-based DEC-TED decoder, the coefficient calculator in the conventionaldecodercanbe a part of the error counter because it originally generates the $\sigma 0$ vector. The error counter additionally requires m-bit OR gate for the BCH code in GF (2^m) to evaluate the flag [1] value. Thus, the costs of area, delay, and power in the error counter for the PA-based DEC-TED decoder with adaptive error correction are minor. In addition, the cost of area in an additional SE corrector is insignificant becausethe area of the errorlocation block in the SEC decoder is much smaller than that of the coefficient calculator and parallel Chien search blocks in the conventional DEC-TED decoder. The hardware structures of the error counter and the SE corrector blocks are shown in Fig. 6. For the LUT-based DEC- TED decoder, the area and power costs of the SE corrector are negligible. The pair of syndrome vectors and corresponding error patterns in the LUT for the conventional DEC-TED decoder can be divided into two parts. One is for single-bit error cases, and the other is for double- bit error cases. Thus, each part can be replaced by the SE corrector and the DE corrector in the proposed LUT-based decoder, respectively. Since the number of errors is already classified in the error counter block, the S0vector is no longer necessary in both SE and DE correctors in our proposed LUT- based decoder. Also,especially for the SE corrector, only the m-bits S1 vector is required to determine the corresponding error vector. Thus, the sum of the total area of the SE and DE correctors is smaller than that of theLUT block in the conventional LUT-based decoder. Unlike the PA-based decoder, increased area, delay, and power consumption due to the additional error counter are inevitable. However, it would be insignificant or compensated by the reduced area ofthe LUT block owing to the smaller required size ofthe syndrome vector. For the hardware implementation of the LUT-based decoder, only the DE corrector block differs from the PA- based decoder. The DE corrector for the LUT-based decoderis implemented with AND gates similar to the SE corrector block.

Invalid Transition Inhibition Technique for DEC-TEDDecoder

As mentioned in Section III-B, settled syndrome vectors should be transferred to the SE or DE corrector to prevent invalid transitions. Furthermore, the SE and DE correctors should not operate simultaneously in the proposed decoder to ensure lower power consumption. FFs are used between the syndrome generator and the SE and DE correctors to satisfy these two constraints. A block diagram of the proposed DEC- TED decoder with adaptive error correction and invalid transition inhibition techniques is shown in Fig.

1. Note that positive-edge-triggered FFs are used in this design. FFs connected to the SE corrector (DE corrector) are called SEC-FFs (DEC-FFs) for easy representation.

[illegible]
PRINCIPAL/DIRECTOR
VIKAS GROUP OF INSTITUTIONS
NUNNA - 521 212
Vijayawada Rural, NTR Dist., A.P.

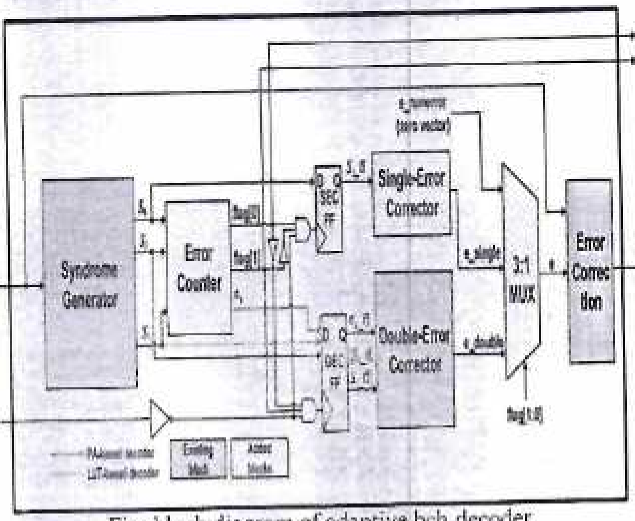


Fig: block diagram of adaptive bch decoder

To make sure that both FFs transfer the settled syndrome vectors, the control signals of both FFs should be activated after the syndrome vector and flag bits become stable. To achieve this, a specific clock forthe decoder (called the ECC clock) is used to generatethe control signal of the FFs, as shown in Fig. 8. For positive-edge-triggered FFs, an inverting ECC clock (FF clock in Fig. 8) is used, and the pulse width of the ECC clock should be larger than the summation of the worst delay of syndrome generator (*T*synd) and that of error counter (*T*EC). In addition, to prevent the simultaneous operation of SEC and DEC-FFs, a clock-gating techniqueis applied to the FF clock signal and flag bits using simple INV and gates. Note that for non- and triple- error bits cases, both FFs do not transfer the vectors tothe following blocks.

The SEC-FFs convey the settled $S1$ vector to the SE corrector only when a single-bit error occurs. DEC-FFs do not transfer the syndrome vectors to the DE corrector; thus, the power consumption is significantlyreduced in the single- bit error case. Similarly, when adouble-bit error occurs, only DEC-FFs transfer the $S1$ and $\sigma 0$ vectors ($S1$ and S3 vectors) to the DE corrector in the PA-based (LUT-based) decoder.

VI. TRIPLE BIT ERROR CORRECTING BCH DECODER

Here we are implementing the triple bit error correcting BCH decoder using an adaptive error correction and an invalid transition inhibition technique is proposed to achieve the high decoding efficiency and low-power consumption. The conceptual block diagram of the proposed adaptive error correction technique. After syndrome vectors are generated, the number of errors caused in the received codeword is classified in an error counter block, and a 2-bit flag signal that represents the number of errors is generated. Then, different error correction algorithms are applied depending on the generated flag signal to improve the decoding efficiency, and a proper error vector is added to the received codeword through the4:1 MUX.

[illegible]

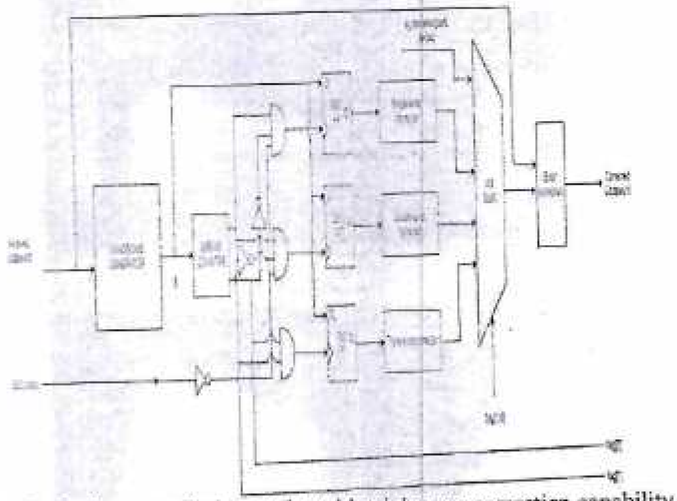
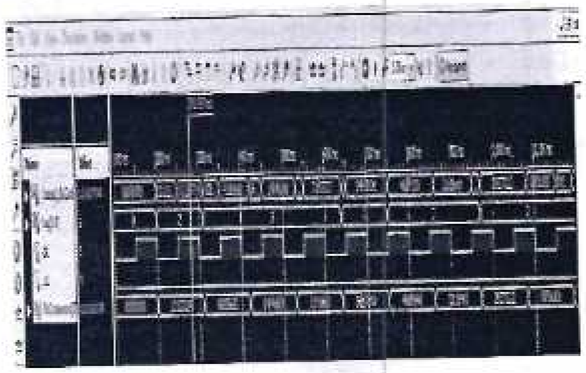
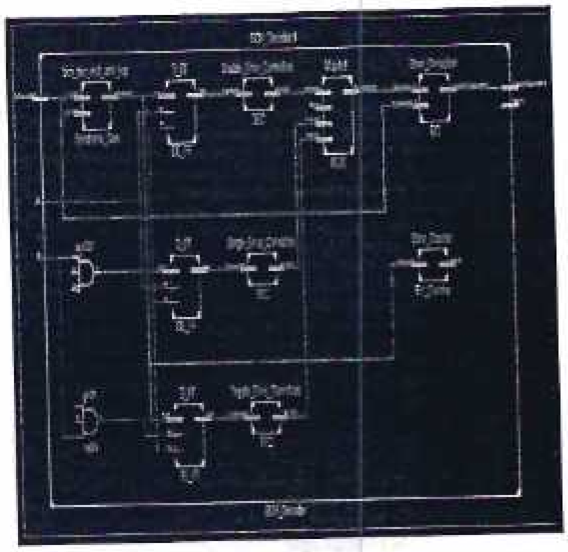


Fig: block diagram of bch decoder with tripleerror correction capability

VII. RESULTS SIMULATION RESULTS:

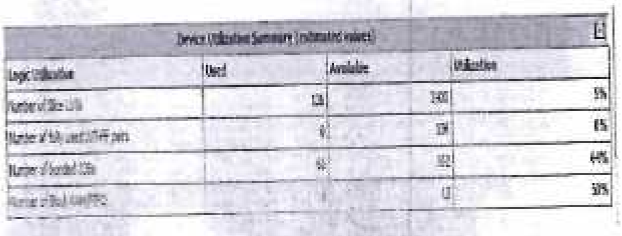


RTL Design:



[illegible]
PRINCIPAL/DIRECTOR
VIKAS GROUP OF INSTITUTIONS
NUNNA - 521 212
Vijayawada Rural, NTR Dist., A.P.

Device Summary:



Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	128	2400	5%
Number of fully used LUT-FF pairs	9	139	6%
Number of bonded IOBs	66	102	64%
Number of BUFG/BUFGCTRLs	[illegible]	12	50%

VIII. CONCLUSION:

In this brief, we proposed an adaptive triple bit error correcting (TEC) BCH decoder with high decoding efficiency for emerging memories. In this an adaptiveerror correction technique that chooses a different decoding algorithm depending on the error conditionsin a code word, to improve the decoding efficiency regarding delay and power consumption. The area overhead due to the added blocks compensated by increased latency constraint of a decoder. Totally this is highly useful for high performance emergingmemories.

REFERENCES:

[1]. 1.P. Amato, S. Bellini, M. Ferrari, C. Laurent, M. Sfozin, and A. Tomasoni," Fast decoding ECC for future memories", "IEEE J.Sel. Areas Commun, vol.34,no.9,pp.2486-2497, Sep.2016.

[2]. S.H. King, "Embedded STT-MRAM for energy-efficient and cost-effective mobile systems", in IEEE symp. VLSI Circuits Dig. Tech.Papers, Jun.2014,pp.36-37.

[3]. H.Noguchi,K. Ikegami,N Shimomura,T. Tetsufumi, J. Ito, and S.Fujitha,"Highly reliable and low –power nonvolatile cache memory with advanced perpendicular STT-MRAM for high-performance CPU," in IEEE Symp. VLSI Circuits Dig. Tech. papers,jun.2014,pp.1-2.

[4]. P. Amato, C. Laurent, M. Sforzin, S. Bellini, M. Ferrari, and A. Tomasoni," Ultra fast,two-bit ECC for emerging memories," in Proc.6th IEEE Int. Memory Workshop Workshop(IMW),May 2014, pp.79-82.

[5]. Y. Emre, C. Yang, K. Sutaria, Y. Cao, and C. chakrabarti, "Enhancing the reliability of STT-RAM through circuit and system level techniques," in Proc. IEEE Workshop signal Process. Syst., Oct. 2012, pp.125-130.

[6]. D. Niu, Y.Xiao, and Y. Xie, "Low power memristor- based ReRAM design with error correcting code", in Proc.17 th Asia South Pacific Design Autom. Conf., jan./Feb. 2012,pp. 79-84.

[7]. M.Mao, Y.Cao,S. Yu, and C. Chakrabarthi," Optimizing latency, energy, and reliable of 1T1R ReRAM through cross-layer techniques", IEEE J. Emerg. Sel. Topics Circuits Syst., vol. 6,no. 3, pp. 352-363,Sep.2016.

[8]. C. Yang, M. Mao, Y. Cao, and C. chakrabarti,"Cost- effective design solutions for enhancing pram reliability and performance", IEEE Trans. Multi-Scale Comput. Syst., vol. 3, no. 1-11, jan./Mar. 2017.

[9] B. Del Bel, J. Kim, C. H. Kim, and S. S Sapatnekar,"

[10]. Improving STT-MRAM desity through multibiterror correction", in Proc. IEEE/AC

[11]. conf.Design,Autom.Test Eur.(DATE),Mar.2014,pp. 1-6.

[12]. Z.Pajouhi, X. Fong, and K. Roy,"Device/circuit/architecture co-design of reliable STT- MRAM", in proc. IEEE/ACM Conf.Design, Autom. Test Eur.(DATE),Mar.2015,pp. 1437-1442.

[13]. Y. Alkabani, Z. Koopmans, H. Xu,A.K.Jones, and R.Melhem,,"Write pulse scaling for energy efficient STT-RAM,"in Proc.IEEE/ACM

[14]. X.Wang,M.Mao,E.Eken, W.wen, H. Li, and Y.Chen,"Sliding basket:An adaptive ECC Schemefor runtime write failure suppression of STT-RAM cache,"in Proc. IEEE/ACM Conf. Design, Autom. Test Eur.(DATE),Mar. 2016, pp. 762-767.

[15]. X.Wang,D. Wu, C. Hu,L.Pan, and R.Zhou,"Embeddedhigh-speed BCH decoder for generation NOR flash memories",in Proc. IEEE custom Integer. Circuits Conf.(CICC),Sep. 2009, pp. 195-198.

[16]. W.Xueqiang, P.Liyang,W.Dong , H.Chaohong, and Z. Runde, "A High-speed two-cell BCH decoder for error correcting in MLC nor flash memories," IEEE Trans.

[17]. Circuits Syst. II, Exp. Briefs, vol. 15,no. 11, pp, 865- 869,Nov.2009.

[18]. Y. Yoo and L-C. Park,:"A search-less DEC BCH decoder for low complexity fault-

[illegible]

PRINCIPAL/DIRECTOR
VIKAS GROUP OF INSTITUTIONS
NUNNA - 521 212
Vijayawada Rural, NTR Dist., A.P.

tolerant systems," inProc. IEEE Workshop Signal Process,Syst.(SiPS),Oct.2014, pp. 1-6.

[19]. C,-C. Chu, Y.-M. Lin,C.-H. Yang and H.- C. Chang,"A fully parallel BCH codes with double error correcting capability for NOR flash applications,"in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), Mar.2012, pp. 1605-1608.

[20]. R. Naseer and J. Draper,"Parallel double error correcting code design to mitigate multi-bit upsets in SRAMs," in Proc.Eur,Solid-State Circuits Conf.(ESSCIRC), Sep.2008, pp. 222-225.

[21]. S.Lin and D. J. Constello, "BCH codes," in Error Control Coding:Fundametals and Applications,2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall,2004,pp. 222-225.

[22]. D. H. Yoon, J. Chang, R. S. Schreiber, and N.P. Jouppi,"Practical nonvolatile multilevel-cell phase change memory,"in Proc. Int. Conf.High Perform, Comput, Netw., Storage Anal.,Nov.2013,pp. 1-12.

[23]. B. L. Ji et al.,"In-line-test of variability and bit-error- rate of HfO_x –based resistive memory,"in Proc.IEEE Int, Memory Workshop(IMW),May 2015,pp. 1-4.

PRINCIPAL/DIRECTOR
VIKAS GROUP OF INSTITUTIONS
NUNNA - 521 212
Vijayawada Rural, NTR Dist., A.P.

Juni Khyat
(UGC Care Group I Listed Journal)

ISSN: 2278-4632
Vol-11 Issue-01 2021

Design_of_Efficient_32-bit_Vedic_Multiplier

Binni Gollapalli, PG Scholar, Department of ECE, Vikas Group of Institutions, JNTUK, Andhra Pradesh, Email id: binnigollapalli23@gmail.com

G Sekhar Reddy, Assistant Prof, Department of ECE, Vikas Group of Institutions, JNTUK, Andhra Pradesh, Email id: gaddamsekharreddy@gmail.com

Abstract: Multipliers are vital components of any processor or computing machine. More often than not, performance of microcontrollers and Digital signal processors are evaluatedon the basis of number of multiplications performed in unit time. Hence better multiplier architectures are bound to increase the efficiency of the system. Vedic multiplier is one such promising solution. Its simple architecture coupled with increased speed forms an unparalleled combination for serving any complex multiplication computations. Tagged with these highlights, implementing this with reversible logic further reduces power dissipation Power dissipation is another important constraint in an embedded system which cannot be neglected.

In this paper we bring out a Vedic multiplier known as "Urdhva Tiryakbhayam" meaning vertical and crosswise, implemented using reversible logic, which is the first of its kind. This multiplier may find applications in Fast Fourier Transforms (FFTs), and other applications of DSP like imaging, software defined radios, wireless communications.For arithmetic multiplication, various Vedic multiplication techniques like Urdhva tiryakbhyam, Nikhilam and Anurupye has been thoroughly discussed. It has been found that Urdhva tiryakbhyam Sutra is most efficient Sutra (Algorithm), giving minimum delay formultiplication of all types of numbers, either small or large.Further, the Verilog HDL coding of Urdhva tiryakbhyam Sutra for 8x8 bits multiplication and their FPGA implementation by Xilinx Synthesis Tool on Spartan 3E kit have been done.

Keywords- Urdhva Tiryakbhayam,Fast Fourier Transforms (FFTs),Vedic multiplier..

1.Introduction: A Multiplier is an electronic circuit used in digital circuits, such as a computer, to multiply two binary numbers. It is built using binary adders. A variety of computer arithmetic techniques can be used to implement a digital multiplier. In digital circuits multiplying to binary numbers is done using repeated addition using full adders and half adders. A multiplier is used in many applications such as image processing, signal processing, microprocessors, and microcontrollers, etc.

ISSN: 2278-4632
Vol-11 Issue-01 2021

There are many types of multipliers such as array multiplier, Wallace tree multiplier, Vedic multiplier, booth multiplier, among all these multipliers Vedic multiplier has less delay when compared to other multipliers [2].Vedic Multiplier is an efficient multiplier with less delay. The structure/block diagram of the Vedic multiplier is similar to that of the array multiplier with some changes in the most significant bit. Vedic Multiplier is based on 16 Vedic sutras in which Urdhva triyakbhyam is general. multiplication formula. The Urdhva triyakbhyam sutra is also known as vertical and crosswire. This sutra was traditionally used in ancient India for multiplication in less time. As the number of multiplication bits increases the timing delay also increases. The Vedic multiplier is used in various applications such as digital signal processing (DSP), communication systems. It is also used in image Processing as fast Fourier transforms, convolution, and in an ALU of a microprocessor.

2.Vedic Multiplier: Vedic mathematics is part of four Vedas (books of wisdom). It is part of Sthapatya-Veda (book on civil engineering and architecture), this is an upa-veda (supplement) of Atharvana Veda. It covers explanation of several modern mathematical terms including quadratic equations, factorization ,arithmetic, geometry (plane, co- ordinate), trigonometry, and even calculus. The proposed Vedic multiplier is based on the Vedic multiplication formulae (Sutras). For the multiplication of two numbers in the decimal number system the proposed Sutras have been traditionally used. To make the proposed algorithm compatible with the digital hardware in this work, the same ideas are applied to the binary number system.

Urdhva Tiryakbhyam Sutra is applicable to all cases of multiplication. This sutra performs the multiplication using the principle Vertically and crosswise multiplication. The generation of all partial products can be done with the concurrent addition of these partial products. The figure explains parallelism in generation of partial products and their summation.

The algorithm can be generalized for n x n bit number. This multiplier is independent of the clock frequency of the processor as the partial products and their sums are calculated in parallel. The same amount of time will require by this multiplierto calculate the product and thus it is independent of the clock frequency.

[illegible]

PRINCIPAL/DIRECTOR
VIKAS GROUP OF INSTITUTIONS
NUNNA - 521 212
Vijayawada Rural, NTR Dist., A.P.

ISSN: 2278-4632
Vol-11 Issue-01 2021

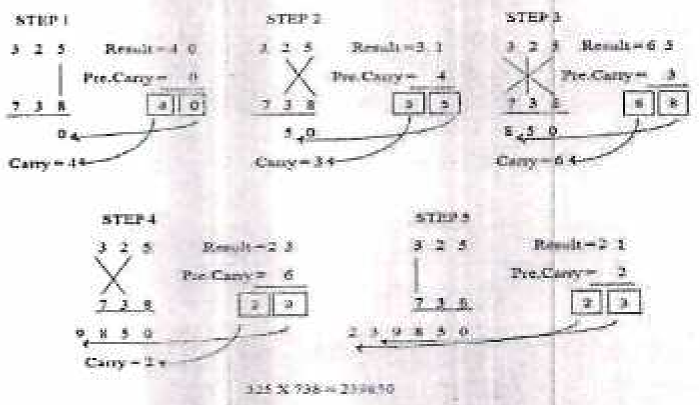


Fig1: Showing Multiplication of two decimal numbers by UrdhvaTiryakbhyam

Algorithm for 4 x 4 bit Vedic multiplier Using Urdhva Tiryakbhyam (Vertically and crosswise) for two Binary numbers

CP = Cross Product (Vertically and Crosswise)

				X3	X2	X1X0	Multiplic andMultiplier
				Y3	Y2	Y1Y0	
H	G	F	E	D	C	BA	
P7	P6	P5	P4	P3	P2	P1P0	Product

PARALLEL COMPUTATION METHODOLOGY

CP X0 = X0 * Y0 = AY0

2. CP X1 X0 = X1 * Y0+X0 * Y1 = BY1 Y0
3. CP X2 X1 X0 = X2 * Y0 +X0 * Y2 +X1 * Y1 = CY2 Y1 Y0
4. CP X3 X2 X1 X0 = X3 * Y0 +X0 * Y3+X2 * Y1 +X1 * Y2 = DY3 Y2 Y1 Y0
5. CP X3 X2 X1 = X3 * Y1+X1 * Y3+X2 * Y2 = EY3 Y2 Y1

[illegible]

ISSN: 2278-4632
Vol-11 Issue-01 2021

6. CP X3 X2 = X3 * Y2+X2 * Y3 = FY3
Y2

7 CP X3 = X3 * Y3 = GY3

3 Results

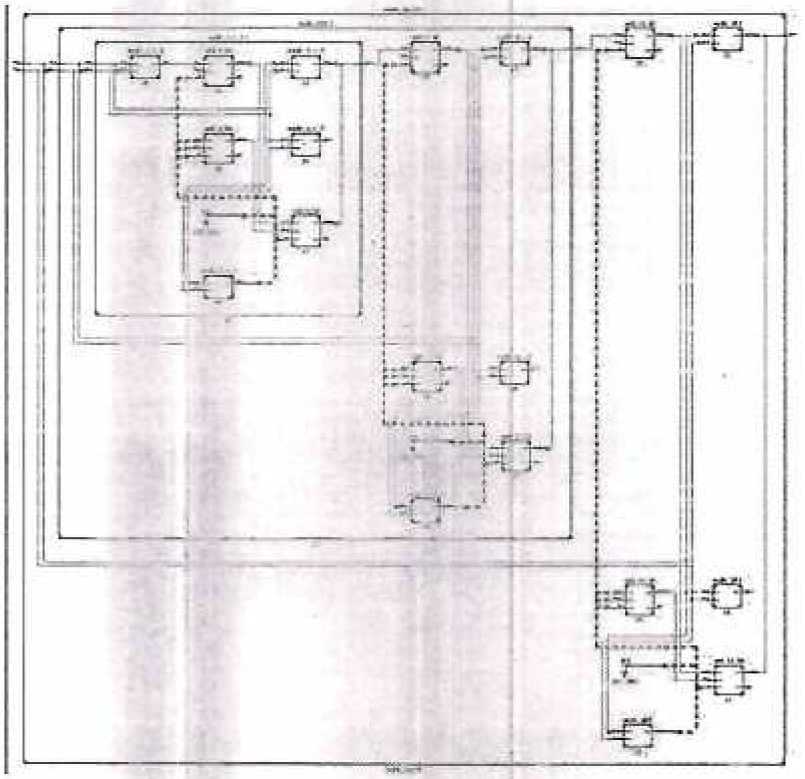


Fig 1 RTL Schematic 32*32 bit vedic multiplier

Fig 2 Simulation results

[illegible]

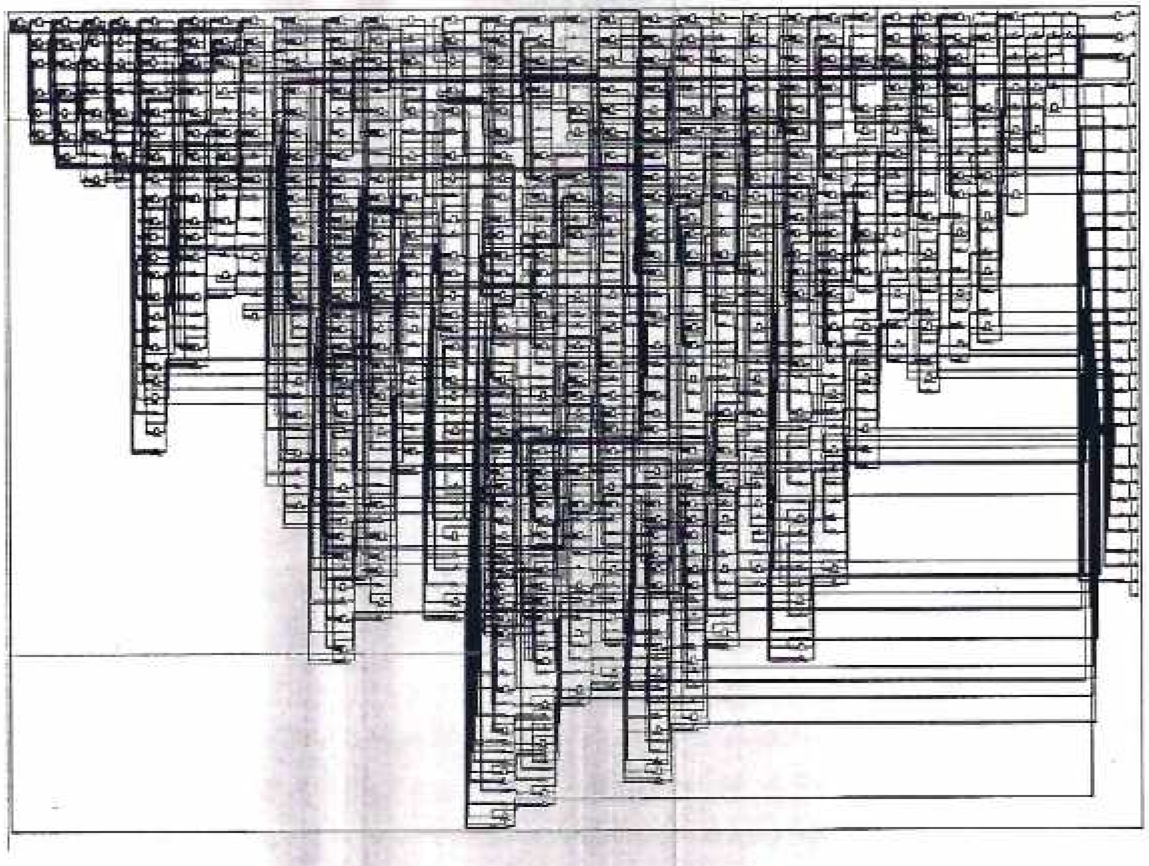


Fig 3 Technology Schematic

4 Conclusion and Further Scope

Urdhva tiryakbhyam sutram, Nikhilam sutram and Anurupyena sutram are the important among such vedic algorithms which can reduce the delay, power and hardwarerequirements for multiplication of numbers. The hardware realization of the Vedic mathematics algorithms is easily possible through the FPGA implementation of these multipliers. The computational speed drastically reduces if all those methods are effectively used for the hardware implementation. Hence there is a chance for implementing a complete ALU using Vedic mathematics methods. Vedic mathematics is long been known but has not been implemented in the DSP and ADSP processors employing large number of

[illegible]

multiplications in calculating the various transforms like FFTs and the IFFTs. By using these ancient Indian Vedic mathematics methods world can achieve newheights of performance and quality for the cutting edge technology devic

REFERENCES

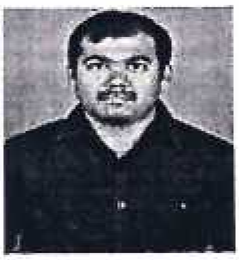
1. Swami Bharati Krishna Tirthaji Maharaja, "Vedic Mathematics", MotilalBanarsidass Publishers, 1965.
2. Rakshith T R and RakshithSaligram, "Design of High Speed Low Power Multiplier using Reversible logic: a Vedic Mathematical Approach", International Conference on Circuits, Power and Computing Technologies (ICCPCT-2013), ISBN: 978-1-4673-4922-2/13, pp.775-781.
3. M.E. Paramasivam and Dr. R.S. Sabeenian, "An Efficient Bit Reduction Binary Multiplication Algorithm using Vedic Methods", IEEE 2nd International Advance Computing Conference, 2010, ISBN: 978-1-4244-4791-6/10, pp. 25-28.
4. Sushma R. Huddar, Sudhir Rao Rupanagudi, Kalpana M and Surabhi Mohan, "Novel High Speed Vedic Mathematics Multiplier using Compressors", International Multi conference on Automation, Computing, Communication, Control and Compressed Sensing(iMac4s), 22-23 March 2013, Kottayam, ISBN: 978-1-4673-5090-7/13, pp.465- 469.
5. L. Sriraman and T. N. Prabakar, "Design and Implementation of Two Variables Multiplier Using KCM and Vedic Mathematics", 1st International Conference on Recent Advances in Information Technology (RAIT -2012), ISBN: 978-1-4577-0697-4/12.
6. Prabir Saha, Arindam Banerjee, Partha Bhattacharyya and Anup Dandapat, "High Speed ASIC Design of Complex Multiplier Using Vedic Mathematics", Proceeding of the 2011 IEEE Students' Technology Symposium 14-16 January,2011, IIT Kharagpur, ISBN: 978-1-4244-8943-5/11, pp.237-241.
7. Soma BhanuTej, "Vedic Algorithms to develop green chips for future", International Journal of Systems, Algorithms & Applications, Volume 2, Issue ICAEM12, February 2012, ISSN Online: 2277-2677.
8. Yogendra and Anil Kumar Gupta," Design of High Performance 8-bit Vedic Multiplier", International Conference on Advances in Computing, Communication, & Automation (ICACCA), September 2016.

[illegible]

ISSN: 2278-4632
Vol-11 Issue-01 2021

9. S. Jayakumar and S. Sumathi, "High – Speed Vedic Multiplier for Image Processing using FGPA," IEEE/10th International Conference on Intelligent Systems and Control (ISCO), January 2016.
10. Rendlesham Gupta, Rajdeep Shar, K. L. Baishnab Jishan Mehedi, "Design of high performance 8-bit Vedic Multiplier using compressor," IEEE/International Conference on Advances in Engineering and Technology (ICAET), May 2014.
11. J. Vinoth Kumar and C. Kumar Charlie Paul., "Design of Modified Vedic Multiplier and FPGA implementation in Multilevel 2d-DWT for Image Processing Applications," Second International Conference on Current Trends in Engineering and Technology - ICCTET 2014, July 2014.
12. Vaidyanathan Kuchai, Linganagouda Kulkarni, Subhash Kulkarni, "High Speed and Area Efficient Vedic Multiplier," International Conference on Devices, Circuits and Systems (ICDCS), April 2012

Binni Gollapall, Pg scholar, Vikas Group of Institutions, JNTUK, Andhra Pradesh, Specialization in VLSI Design



G SEKHAR REDDY, M.TECH IN GITAM UNIVERSITY, ASSISTANT PROF, AREA OF INTEREST :LOW POWER VLSI, Vikas Group of Institutions, JNTUK, Andhra Pradesh

[illegible]