Design and Analysis of Multipliers using Radix-8 Booth Encoding Technique for Low Power and Area Consumption

D.Chandrika Sowmini, A.Lavanya, D.Jagadeesh Sai, C.Raja, M Aditya

***Abstract**: As innovation scaling is arriving at its points of confinement, new methodologies have been proposed for computational efficiency. Different techniques have been proposed with advancements in technology to model high-speed along with low power consumption and smaller area multipliers. For the radix-4 booth propagation algorithm for low-power and low complexity applications, an efficient approximate 8 bit redundant multiplier is used. To minimize the complication present in modified booth encoder, approximate Booth RB encoders have been introduced by modifying the truth table with incorrect bits, which resulted in a reduction of the power delay product. Approximate computing is a relevant technique for low power and high performance circuits as used in error-tolerant applications. Approximate or inexact computing is an attractive design methodology for low power design but accomplished by loosening up the necessity of precision. It becomes critical to maintain full accuracy to attain reduced power utilization. In this paper, the design of approximate redundant binary (RB) multipliers is studied and modified to build less complex multiplier with Radix-8 modified booth encoding technique to reduce area and complexities of architectures.*

***Keywords**: Redundant Binary Multipliers, Radix-8, Low power, Approximate multipliers, Modified Booth Encoding, Area optimization.*

I. INTRODUCTION

The main requirements of the applications for which multipliers are needed, such as computer applications, processing of signals and processing of images, have always been high speed multipliers. In the DSP blocks are primarily computerized by ALU, whose major component is multiplication and addition.[3] The main operation in the element processing are multiplications which generally need high utilization of power and energy Compared with many other arithmetic operations, multiplication is time taking and requires high power. Digital multipliers, in many instances foist a speed limit on the whole design due to the critical paths.[1] Many architectures and numerous high performance algorithms have been proposed to achieve the process of multiplication more aptly [4]. In the first step generation of partial products takes place. Then in the next step summation in pair wise form of partial product rows are performed until the number of left over partial products are two. Thereafter third step the two left over rows are summed to generate the final product by fast carry-propagate adder. So, the conventional multipliers with high speed are not competent of on the condition that the high speed computational needs at low power requirements.[4]The Consequently implementing multipliers with high performance is one of the leading objectives for designers of system to enhance the entire performance of the systems[2]. So here in the beginning step, Modified Booth Encoding (MBE) is used strikingly to optimize the partial product by half [10]. Wallace trees which are some of the Partial product reduction schemes are used for partial product rows summation in the next step. In the step three, the final multiplication product is obtained by the two rows from partial product [5] summing tree which are added by means of fast carry-look ahead or carry save adder or carry select adders can be used [2]. A number of proceedings were put forward in the recent years to with the above three steps to achieve an efficient multiplier.[2] By the module of modified booth encoding the partial product terms are reduced to half and thus credit the design of multiplier by decreasing the size and improving the speed of the summing tree [3]. Hence, the multipliers with high speed, low power and area[8] [9] in VLSI design has always been and is still a research subject. One of the signed digit representations, for fast parallel arithmetic, redundant binary (RB) [1] representation is used. The specifications high modularity and addition without carry, made the usage of the representation in redundant binary in the design of high-speed multipliers [2]. The redundant binary number representation is engaged to perform signed digit arithmetic due to the fact that a redundant number can be represented in various forms. Using redundant binary addition trees multipliers can be designed fast which posses advantageous features. Redundant Binary multipliers give reassuring output for high performance multiplication but increasing complexity and delay [2].

Revised Manuscript Received on December 30, 2019.
* Correspondence Author

D.Chandrika Sowmini*, Student, Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India.
A.Lavanya, Student, Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India.
D.Jagadeesh Sai, Student, Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India.
Dr.C.Raja, Professor, Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India.
M.Aditya, Assistant Professor, Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India.

The performance of any of the VLSI circuit can be lowered by the delay of any gate used. These circuits use critical path delays the entire circuit clock cycle in order to perform accurately[13]. Yet, in many worst-case models, there is certain chance that the critical path delay that is present is low. In those kinds of cases, the strategy of keeping down the worst-case conditions bring about incompetent designs [4]. For non-critical path, using the critical path delay as the overall cycle period will result notable waste of time [12]. Hence, the variable latency design was introduced to reduce the timing waste of conventional circuits.Booth techniques are widely used in multipliers, to improve the performance of the multiplier [2]. Booth encoding, aids in optimizing the number of partial product rows that are needed to be added for acquiring results. Grouping of the bits directly affects the number of partial product rows. Operation on the multiplicand is performed by the groups of bits. In a booth multiplier grouping is done by 2 bits for Radix-4 and 3 bits for Radix-8. Decreasing in the number of partial products, will lead to substantial delay and area reduction. Repeatedly summing one of two preordained values to a product P is the mechanism of Booth's algorithm. And then rightward arithmetic shift is done on P [6][8]. Partial products are the products formed during intermediate steps in computing larger products[14].By using the Wallace tree multiplier as an alternative to the array multiplier one can achieve better performance in terms of speed, a tree like structure of the partial products are added using adders reducing the combinational delay along with the number of adder cells needed which leads to the cutback in hardware area required [4]. O(log3/2(N)) is the propagation delay in the multiplier design comprising Wallace tree structure. Hereby, there is study on the approximate design of a radix-8 Booth multiplier which is one of the most popular schemes for signed multiplication. In the first step of a radix-8 Booth multiplier, partial products are generated by a radix-8 modified Booth encoding (MBE) [7]. The number of partial products can be minimized by a factor of three with the use of radix-8 MBE. The implementation of the MBE remarkably affects the area, delay and power consumption of the Booth multipliers. Extra partial product rows are being generated due to grouping 2 in radix-4, in the conventional MBE algorithm [3]. This leads to an partial product array with irregularity and requiring a complex reduction tree and the hardware implementation becomes intense. In this paper, Radix-8 MBE algorithms are proposed and analyzed. Multipliers are designed based on the proposed Radix-8 MBEs, in which a regular partial product array is achieved reducing the number of partial product rows and computation. Delay, area, power utilization comparisons are also provided through simulation.

II. DECIMAL MULTIPLICATION:

1. Assuming the number were whole number, multiply them.
   i. On to the right side, put the numbers in rows – but alignment of the decimal points should not be done.
   ii. In the top number each digit is multiplied by each digit in the bottom number, as of from the right.
   iii. Then the products should be summed.
2. By starting at the right now decimal point should be placed in the answer in such a way that moving the number of places which is equal to the sum of the decimal places in both numbers that are multiplied.

Shift and add multiplier general architecture is shown in the figure-- for a 32 bit multiplication. A value of the multiplicand is summed and depending on the value of multiplier LSB bit, a value of the multiplicand is added and piled. One bit of the multiplier is shifted towards the right at every clock cycle and tested. Shift operation is performed only when the value is 0, in case if it’s 1, accumulator is summed with the multiplicand and then right shifted by one bit. Product gets into the accumulator only after all the bits in multiplier are tested. The maximum delay is N cycles. In asynchronous circuits, the circuit is advantageous.

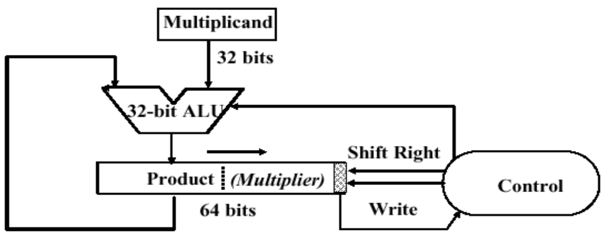

Fig.1 32-bit shift and add general architecture.

III. EXISTING TECHNIQUES

In general the redundant binary multiplier consists of two approximate Booth encoders and two 4:2 approximate compressors with a 16 bit Redundant Binary to Normal Binary convertor which gives the normal binary form as product [1]. To meet different accuracy requirements in redundant binary multipliers, the approximate and exact partial product arrays are used. A regular RB partial product array is also applied in the Redundant Binary multiplier by getting rid of the last Error Correcting Word (ECW)[11].The current designs proposed are compared with previous approximate multipliers; the comparison results show that the proposed designs are better than current when considering both the power and area.

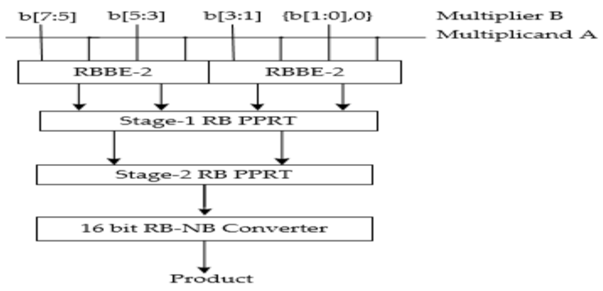

Fig.2 The block diagram of the RB multiplier using Radix-4 encoding

A conventional Booth multiplier consists of three modules: Booth encoder with generated partial products, compressors which accumulate the partial products and fast adder used for summing the partial products and generate the final product. The approximate design of radix- 4 Booth encoding is studied in this section.

A more efficient approximate radix-8 Booth encoding method is proposed in this section by carefully considering the error characteristics. Furthermore, an approximate partial product array the Booth encoding, produces is also designed to make it regular, such that a reduction stage is saved. Approximate Booth multipliers are designed based on the approximate Booth encoder and the regular approximate partial product array.

A. Review of Approximate Radix-4 Booth Encoding Method

The K-map for the radix-4 approximate modified Booth encoder [1], here 0(circled) denotes the modification on which 1 is made 0 and 1 denotes that a circled 0 is replaced by 1.So that here 6 of the values are modifies to make the Booth encoder simple with making it approximate. The truth table is symmetrical is the main principle which brought the approximate design. So now the 0 to 1 and 1 to 0 replacements, 3 of each gives the output in the following way:

$$app_{ij6-1} = (b_{2i} + b_{2i-1})(b_{2i+1} \oplus a_i)$$

$$E_i = (b_{2i+1}\overline{b_{2i}}) + (b_{2i+1}\overline{b_{2i-1}})$$

The K-Map of the above equation can be given by:

Table.1 K-map for Approximate Radix-4 BE

$a_j a_{j-1}$ \ $b_{2i+1}b_{2i}b_{2i-1}$	000	001	011	010	110	111	101	100
00	0	0	0	0	1	①	1	⓪
01	0	0	⓪	0	1	①	1	0
11	0	1	1	1	0	0	0	0
10	0	1	①	1	0	0	0	⓪

The conventional modified booth encoder when implemented in gate level consists of four 2-input XNOR gates, one 2-input XOR gate, one 3-input OR gate, one 2-input OR gate and one 2-input NAND gate while the above one [1] requires only one 2-input XOR gate with one 2-input AND gate and one 2-input OR gate.

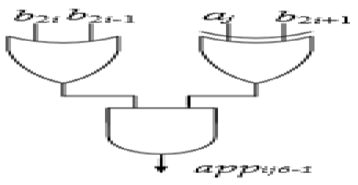

Fig.3 The gate-level circuit.

B. Review of Radix-4 Approximate New MBE

The K-map for the radix-4 with modified new Booth encoding [1], here 0(circled) denotes the modification on which 1 is made 0 and 1 denotes that a circled 0 is replaced by 1. So that here 6 of the values are modifies to make the Booth encoder simple with making it approximate. But in this one, there are more bits that are changed from '0' to '1' than those of which are changed from '1' to '0'. Hence the results acquired from the new design are larger than its counterpart and the gives the output in the following way:

$$app'_{ij6-1} = b_{2i+1} \oplus a_j + b_{2i}b_{2i-1}$$

$$E'_i = b_{2i+1}$$

The K-Map of the above equation can be given by:
Table.2 K-map for Approximate Radix- New MBE

$a_j a_{j-1}$ \ $b_{2i+1}b_{2i}b_{2i-1}$	000	001	011	010	110	111	101	100
00	0	0	①	0	1	1	1	1
01	0	0	1	0	1	1	1	①
11	①	1	1	1	0	1	0	0
10	①	1	①	1	0	1	0	⓪

The new modified booth encoder when implemented in gate level consists of two 2-input OR gates, one 1-input XOR gate

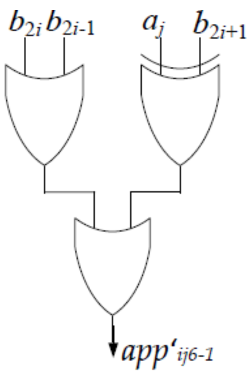

Fig.4 The gate-level circuit of the new modified booth encoder.

PROPOSED ARCHITECTURE:

Booth Algorithm :

The multiplier is transformed into 2's complement form in the booth algorithm and a bit a-1 =0 is appended followed to the least significant bit. While iterating from i=0 to i=K-1, in the next step it inspects the pair a_i and its consecutive bit a_{i-1}. On the comparison the below exe4cution takes place

I. If the bits which are compared are same, product is left unaltered in the accumulator.
II. When the a_i=0 and the a_{i-1} = 1, then the P is added with multiplicand times 2i.
III. When the a_i=0 and the a_{i-1} = 1, then the P is subtracted with multiplicand times 2i.

At the end the signed product P is obtained, representation in two's complement of the multiplicand and product are also present similar to the multiplier. For any number system that can hold subtraction and addition, this algorithm can be generalized.

The iterations are ordered in such a way that it starts from LSB to MSB from i=0.The ordering of the iterations is from LSB to MSB starting from i = 0. For the multiplication by 2, the accumulator P is right shifted and in this incase the LSB is shifted outward. Resulting N bits of P by computation of addition and subtraction is then executed.

Radix-8 Modified Booth Algorithm:

The performance of arithmetic shift to the right on the product P by repeatedly adding one or two preordained values to the P is the technique used in Booth Algorithm.

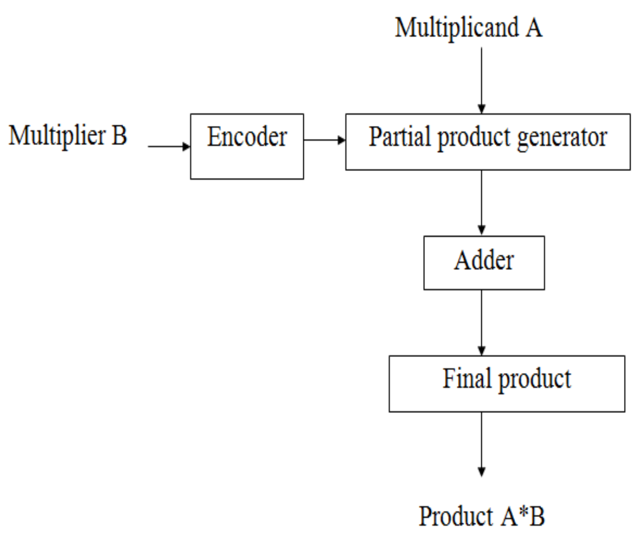

Fig.5 The architecture of the radix-8 modified booth encoding.

The multiplier architecture is typically divided into two as, i.e., Modified Booth and the adders which is different from the Redundant Binary multiplier which used Booth encoder along with compressors for the optimization of partial products and a converter. With the analysis of different architectures of the evolved multipliers, we can realize that the Modified Booth enhanced the speed, since it’s use of the technique Radix-8 along with the reduction of partial products by fifty percent. So now these, on combining the features of the multipliers are the implementation of the fast and low power multipliers. The two major components Booth Encoder is which decrypts the multiplier signal by the performing a functions and feed it’s to the second component Booth Selector which produces a partial product. Then, the adders are given with the partial products for summation. The carry is being stretched to the left by the positioning after the sum and the carry are added. The same technique or algorithm used in the Radix-4 is used for the Radix-8.The only difference lies in grouping the digits, as of in the radix-4 3 bits are grouped while in the radix-8 we consider quartets. Hence with this the intermediate partial product rows are being cut-off by 3 times the number of the bits in the multiplier. Hereby, we can say that time gain happens due to reduction in the addition of partial products. Every four bits grouped is codified as a signed digit using Table 4.

Table.3 Grouping of 8 bits in Radix-8 technique

A	A	A	A	A	A	A	A	A
8	7	6	5	4	3	2	1	0

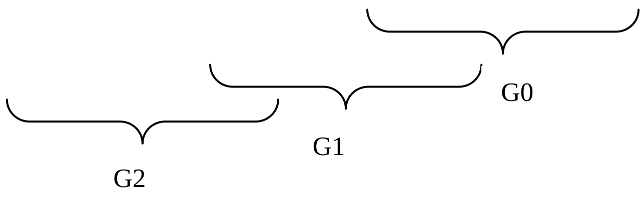

Table.4 Coding each 4 bits with a digit

Quartet	Value
0	0
1	1
10	1
11	2
100	2
101	3
110	3
111	4
1000	-4
1001	-3
1010	-3
1011	-2
1100	-2
1101	-1
1110	-1
1111	0

Multiplying with a digit of the multiplier, which has many digits, and then a product is formed. During the computation of larger products the products formed in the halfway are called the partial products.The product multiplying by multiplying A by 0, 1, -1, 2, -2, -3, -4, 3, 4 by the partial product generator designed. Multiply by zero implies that the product is "0". Multiply by" 1 "means that the product remains the same as the multiplier. A two’s complement of the number is obtained when multiplied by "-1". Multiplying with "-2" is to move left one as this rest as per table.

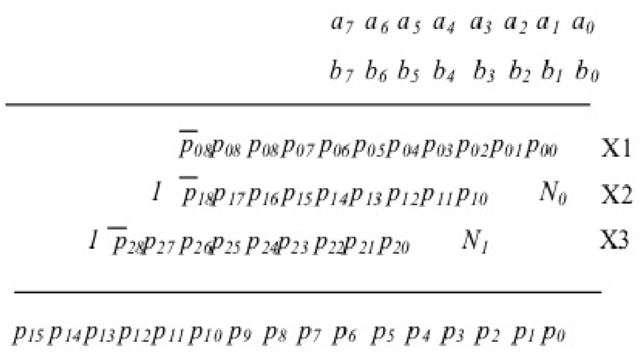
$a_7\ a_6\ a_5\ a_4\ a_3\ a_2\ a_1\ a_0$
$b_7\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0$
$\overline{p_{08}}\ p_{08}\ p_{08}\ p_{07}\ p_{06}\ p_{05}\ p_{04}\ p_{03}\ p_{02}\ p_{01}\ p_{00}$ X1
$1\ \overline{p_{18}}\ p_{17}\ p_{16}\ p_{15}\ p_{14}\ p_{13}\ p_{12}\ p_{11}\ p_{10}$ N_0 X2
$1\ \overline{p_{28}}\ p_{27}\ p_{26}\ p_{25}\ p_{24}\ p_{23}\ p_{22}\ p_{21}\ p_{20}$ N_1 X3
$p_{15}\ p_{14}\ p_{13}\ p_{12}\ p_{11}\ p_{10}\ p_9\ p_8\ p_7\ p_6\ p_5\ p_4\ p_3\ p_2\ p_1\ p_0$

Table.5 Example showing the 3 partial product rows added in Radix 8

Components	No.
Adders / Subtractors (10-bit adder)	6
Multiplexers (10-bit 16-to-1 multiplexer)	4
Xors (1-bit Xor2)	85

IV. RESULTS

In this paper, the design and architecture of the Approximate Redundant Binary multiplier based on the technique of radix-4 booth encoding is studied, analyzed and implemented along with the proposal of new modified radix-8 booth encoding technique. On implementation of both the techniques we concluded the results as the analysis of power, area and delay. Table.5 Comparison in terms of Area, power and Delay.

The above table shows the area in terms of LUTS, a slice consists of one LUT combined with a Flip Flop which is here shown as a architecture of a LUT Flip Flop pair. So the number of these architectures used can directly effect the area of the design. The Slice logic distribution and utilization with the IO utilization is tabled in below for both the multipliers with Radix-4 encoding technique and Radix-8 encoding technique.

Table.6 Macro Statistics using Radix-4 encoding technique

Multiplier	ARB Radix-4	Radix-8 Encoding
Number of Slice LUTs	144	129
Number of Slices occupied	51	37
Number of bonded IOBs:	102	32

Along with above analysis here the Macro Statistics of both the designs are shown in the following tables.

Table.6 Slice Logic Distribution, Utilization and IO Utilization

Components	No.
Adders / Subtractors (9-bit adder)	1
Registers (16-bit register)	1
Multiplexers (16-bit 8-to-1 multiplexer)	4
Xors (1-bit Xor2) (1-bit Xor3) (1-bit Xor4)	155

Table.7 Macro Statistics using Radix-8 Encoding technique

	Booth Multiplier	Array Multiplier	ARB Multiplier (Radix-4 Encoding)	Radix-8 Encoding Technique
Area in terms of LUTs	142	138	144	129
Delay in ns	28.157	29.968	25.336	15.704
Power Consumption in mW	34	61	48	29

V. CONCLUSION

Generally multiplication can be performed with different types of multipliers available but the increase in power and delay effects the performance of the system. With the analysis it is clearly shown that Radix-8 is better than the other techniques. The Approximate Redundant Binary Multiplier uses the Radix-4 encoding technique which needs the partial product reduction tree mechanism to minimize the number of partial product rows as the numbers of partial product rows obtained in Radix-4 are 5 which again need complex adding circuits for summation. Instead, the use of Radix-8 give 3 partial product rows that can be summed by the full adders so that entire circuit become less complex and the propagation delay decreases along with the power consumption and area required.

REFERENCES

1. Weiqiang Liu, Tian Cao, Peipei Yin, Yuying Zhu,(2018) "Design and Analysis of Approximate Redundant Binary Multipliers". IEEE Transactions on computers, vol. xx, no. xx
2. R.S.Keote, P.Y.Karule (2016), " VLSI design of 64bit × 64bit high performance multiplier with redundant binary encoding", IEEE Conference ICACCA.
3. M. Sandhya Rani, K. Naveen Kumar Raju (2018) "Design and Implementation Radix based Booth Multiplier Using High Speed Applications" IJSRCSEIT | Volume 3 | Issue 5 | ISSN : 2456-3307.
4. Weiqiang Liu, Liangyu Qian, Chenghua Wang , Honglan Jiang (2017) "Design of Approximate Radix-4 Booth Multipliers for Error Tolerant Computing", IEEE Transactions on computers, Volume: 66, Issue: 8.
5. Honglan Jiang, Fabrizio Lombardi, Jie Han (2019) "Low-Power Approximate Unsigned Multipliers With Configurable Error Recovery" IEEE Transactions on Circuits and Systems | Volume 66 | No.1
6. Inumula Veera Raghava Rao, "A Novel Low-Power 5th order Analog to Digital Converter for Biomedical Applications", International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8, Issue-6.
7. K.K. Chaddha, R. Chandel, Design and analysis of a modified low power CMOS full adder using gate-diffusion input technique, J. Low Power Electron. 6 (4) (2010) 482–490.
8. Inumula Veera Raghava Rao, "Object Tracking and Object Behavior Recognition System in High Dense Crowd Videos for Video Supervision: A Review", Journal of Advance Research in Dynamical & Control Systems, Vol. 10, 2018.
9. Laya Surendran E K, Rony Antony P (2014) "Implementation Of Fast Multiplier Using Modified Radix-4 Booth Algorithm With Redundant Binary Adder For Low Energy Applications" ICCSC
10. Inumula Veera Raghava Rao, "A novel image recognition method using multiple component resemblance score synthesis and genetic algorithm", IOP Conf. Series: Journal of Physics: Conf. Series 1139 (2018) 012069 IOP Publishing.
11. M. Sainath, P.Muralikrishna (2018) "Approximate Radix-4 Booth Multipliers for Error Analysis" IJESC | Volume 3 | Issue NO.
12. Weiqiang Liu, Liangyu Qian, Chenghua Wang , Honglan Jiang (2017) "Design of Approximate Radix-4 Booth Multipliers for Error Tolerant Computing", IEEE Transactions on computers, Volume: 66, Issue: 8.

13. S. Wariya, R. Nagaria, S. Tiwari, Performance analysis of high speed hybrid CMOS full adder circuits for low voltage VLSI design, VLSI Design (2012) 1–18.
14. H. Mahdiani, A. Ahmadi, S. Fakhraie, and C. Lucas, "Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, pp. 850-862, 2010.

AUTHORS PROFILE

D.Chandrika Sowmini currently doing Bachelor of Technology in Department of Electronics and Communication Engineering at KLEF (Deemed to be University).

A.Lavanya currently doing Bachelor of Technology in Department of Electronics and Communication Engineering at KLEF (Deemed to be University)..

D.Jagadeesh Sai currently doing Bachelor of Technology in Department of Electronics and Communication Engineering at KLEF (Deemed to be University).

Dr.C.Raja currently working as Professor in KLEF (Deemed to be University). He is specialized in Medical image processing. To his credit he has published 6 research papers in SCI, 6 research papers in Scopus indexed international journals and international–level conferences. His research interests are digital image processing, wavelets, optimization (swarm intelligence) and deep learning.

M.Aditya currently working as Assistant Professor in KLEF (Deemed to be University). His research field includes Embedded Systems and VLSI. He also published papers in various international journals