Very Large Scale Integration Architecture of Finite Impulse Response Filter Implementation Using Retiming Technique

S. Jalaja, A. M. Vijaya Prakash

Abstract—Recursive combination of an algorithm based on Karatsuba multiplication is exploited to design a generalized transpose and parallel Finite Impulse Response (FIR) Filter. Mid-range Karatsuba multiplication and Carry Save adder based on Karatsuba multiplication reduce time complexity for higher order multiplication implemented up to n-bit. As a result, we design modified N-tap Transpose and Parallel Symmetric FIR Filter Structure using Karatsuba algorithm. The mathematical formulation of the FFA Filter is derived. The proposed architecture involves significantly less area delay product (APD) then the existing block implementation. By adopting retiming technique, hardware cost is reduced further. The filter architecture is designed by using 90 nm technology library and is implemented by using cadence EDA Tool. The synthesized result shows better performance for different word length and block size. The design achieves switching activity reduction and low power consumption by applying with and without retiming for different combination of the circuit. The proposed structure achieves more than a half of the power reduction by adopting with and without retiming techniques compared to the earlier design structure. As a proof of the concept for block size 16 and filter length 64 for CKA method, it achieves a 51% as well as 70% less power by applying retiming technique, and for CSA method it achieves a 57% as well as 77% less power by applying retiming technique compared to the previously proposed design.

Keywords—Carry save adder Karatsuba multiplication, mid-range Karatsuba multiplication, modified FFA, transposed filter, retiming.

I. INTRODUCTION

K ARATSUBA algorithm was revealed in 1962 by Anatolii Alexeevitch Karatsuba. For two n-digit numbers, it minimizes the multiplication at most $3n^{\log_2 3}$ approximately equals to $3n^{1.585}$ and $n^{\log 2}$ [1]. As a result, it is more efficient compared to the classical algorithm.

Early in 1970's, Recursive Karatsuba-Ofman algorithm [1] was designed and it is best suited for long integer multiplication and finite field applications. Similar asymptotically faster algorithms like the Toom–Cook algorithm as well as Schönhage–Strassen algorithms [7] are significantly speedy, for large inputs. The main highlights of Karatsuba algorithm are its divide-and-conquer approach and recursion which are simple and effective.

Jalaja S, Research Scholar, Visvesvaraya Technological University, Dept. of Electronics and Communication Engineering, Bangalore Institute of Technology, Bangalore, India (e-mail: jalajabit@gmail.com).

Vijaya Prakash A.M, Professor, Dept. of Electronics and Communication Engineering, Bangalore Institute of Technology, Bangalore, India (e-mail: am_vprakash@yahoo.co.in).

For binary number representation, Karatsuba algorithm solves in following steps:

Consider X (i) and Y (i) for two degree-1 polynomials:

$$X (i) = a_1 x + a_0 \text{ and, } Y (i) = b_1 x + b_0 \qquad (1)$$

Then, the product of two numbers can be computed in the following manner:

$$P(x) = (a_1 b_1)\, x^2 + (a_0 b_1 + a_1 b_0)\, x + a_0 b_0 \qquad (2)$$

where

$$(a_0 b_1 + a_1 b_0) = ((a_0 + a_1)\,(b_0 + b_1) - a_0 b_0 - a_1 b_1) \qquad (3)$$

Let there be three auxiliary variables P_0, P_1, and $P_{0,1}$ given by:

$$P_0 = a_0 b_0,\; P_1 = a1b1,\; P_{0,1} = (a_0 + a_1)\,(b_0 + b_1) \qquad (4)$$

Then, (2) can be written as:

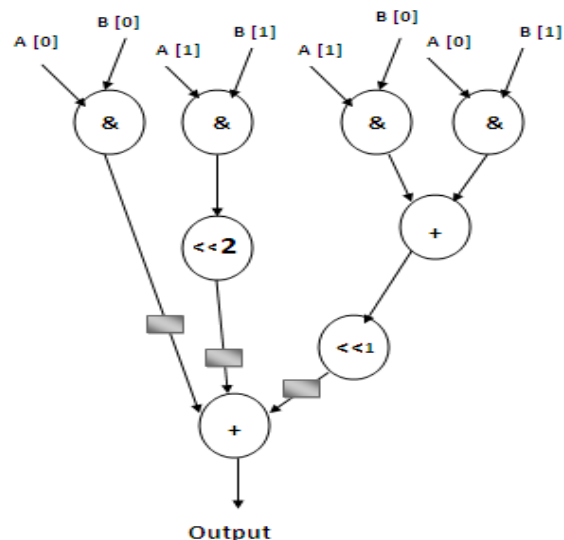$$P(x) = P_1 x^2 + (P_{0,1} - P_0 - P_1)\, x + P_0 \qquad (5)$$

Fig. 1 Retimed DFG for 2-bit Karatsuba algorithm

At the first glance, Karatsuba algorithm uses one less multiplication, but three more additions when compared to classical algorithm like grade school multiplication algorithm. However, Karatsuba-Ofman algorithm will give better complexity, if they are applied recursively. Fundamental concept of Karatsuba multiplier is usually faster for higher order bits. The basic thumb rule of Karatsuba multiplier provides faster results for higher order bits. This critical drawback results in algorithm being not as popular a choice for small to mid input ranges. But, recursion and divide-and-conquer approaches of Karatsuba-Ofman algorithm can still be used to minimize the computational complexity and hence achieve better results as will be discussed in below sections. Apart from that, retiming concept of digital circuit is implemented to estimate propagation delays. Also, Cut-set retiming technique is applied to original graph and it results in two disjoint sub-graphs [4], [10], [11]. To get better result from retiming, the delay elements are moved to cutest intersect is as shown in Fig. 1. The design of FIR filter architecture for fixed and reconfigurable application [3] estimated to reduce the operational complexity. Retimed DFG-3 of mathematical formulation of transpose form type –II was derived. As a result, improvement in ADP and EPS than the exiting direct form blocks the FIR structure. To reduce critical path along with the existing retiming techniques, they used Novel node-splitting and node-merging techniques [14]. Also, an efficient cut-set retiming was adopted to minimize critical path without increase in the storage elements complexity and latency. The proposed work gives significant improvement in power and area.

Data Flow Graph (DFG) for 2-bit Karatsuba Multiplication is designed in Fig. 1. Backward retiming technique is applied to transposed and Fast FIR algorithm (FFA) to gain power reduction. Relatively new combination Karatsuba algorithm designs up to 256 bits reduces the multiplications complexity.

II. Modified Structure of Karatsuba Algorithm

Though the design space of the parent Karatsuba algorithm is rich in offering many variations and even combination of different algorithms, the algorithm is often overlooked while implementing small range to mid-range inputs. The parent algorithm also has architecture drawbacks while implementing in binary and poses significant challenges in achieving recursion. The parent algorithm results in a design, which trades area and timing parameters for low power. This is due to overhead in the recursion [6], [13] addition and subtractions while evaluating the partial products. In this work, we proposed the multi-combination architecture for multiplication for higher bit to get efficient result. The Karatsuba algorithm with conventional multiplication (CKA) and also carry save adder with Karatsuba algorithm (CSK) structures are adopted in FIR Filter.
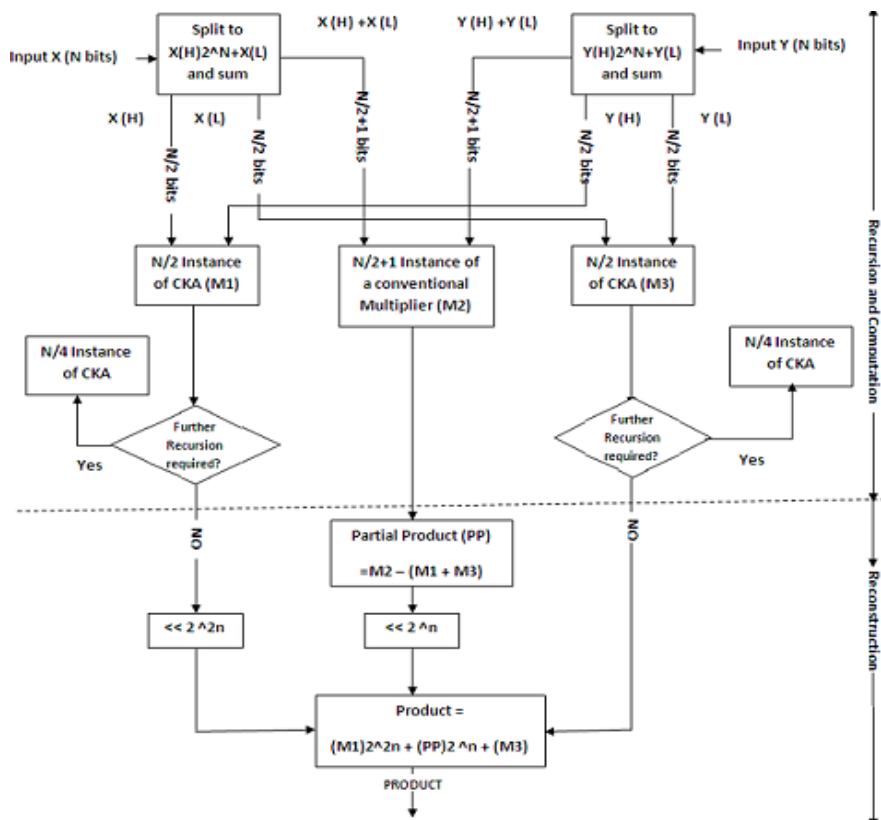

Fig. 2 Architecture of Conventional- Karatsuba algorithm

Recalling (4) and (5) the resultant product of polynomial can be written as:

$$P(x) = P_1 x^2 + (P_{0,1} - P_0 - P_1)\, x + P_0$$

where

$$P_{0,1} = (a_0 + a_1)(b_0 + b_1),\ P_0 = a_0 b_0,\ P_1 = a_1 b_1.$$

The computation of P_0 and P_1 involves terms which are of n/2 bit length and $P_{0,1}$ involves terms which are of "n/2+1" bit length. Computing for $P_{0,1}$ becomes expensive and results in extra overhead during recursion. A simple solution would be the use of zero padding to adjust bit lengths and go for a reconfigurable generalized architecture [12], but this still requires some additional operations like adding, shifting, multiplexing, and n bit one multiplication. This can be done with the help of combination of algorithms.

Recursion is beneficial for powers of "$2^{n/2}$", as it results in reusability of modules and hence also provides an opportunity for parallel architecture implementation. However, it becomes expensive for the powers of "$2^{n/2}+1$". This overhead in recursion, addition and subtractions while evaluating the partial products becomes the major drawback of Karatsuba algorithm for small to mid input range.

In this work, we propose the implementation of the '$2^{n/2}+1$' blocks using a conventional multiplier and use of recursion only on the '$2^{n/2}$' blocks is as shown in Fig. 3. In this work we use the multiplier provided by Cadence 90nm fast technological library as Conventional Multiplier for all purposes.

Figs. 2 and 3 give the Architecture of Conventional-Karatsuba algorithm and Carry Save adder Karatsuba algorithm.
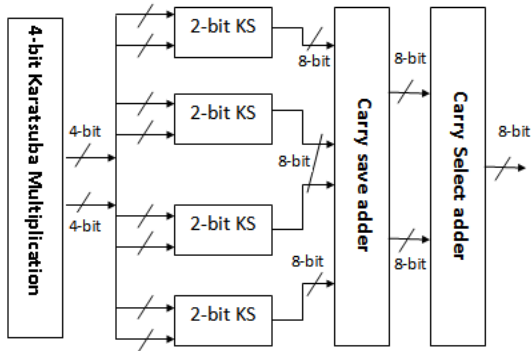

Fig. 3 Architecture of Carry Save adder Karatsuba Algorithm

Fig. 3 shows the implementation of a 4×4 multiplier. It requires four 2×2 multipliers of modified Karatsuba multiplier and an 8-bit Carry Save adder. The Carry Save adder takes three 8-bit input operands followed by Carry Select adder provides an 8-bit product. Same structure is design for NxN multiplier based on Karatsuba concepts with N/2 x N/2 multipliers and a single 2n-bit Carry Save adder along with Carry Select adder. With these proposed architectures, we achieve better performance in-terms of power and speed [9].

III. MATHEMATICAL FORMULATION FOR FIR FILTER DESIGN USING KARATSABA ALGORITHM

FIR filters are irreplaceable fundamental components in almost all multimedia and DSP systems involving communication, video, voice, and image processing applications. A video ghost canceller for broadcast television requires large filter tap length. Similarly, for different application the filter tap length will vary between a 4-tap filter to high tap filter. Most filters for voice, video, and image involves a dynamic range of large co-efficient which makes FIR Filters a tailor made option to implement with the proposed algorithm.

Albicocco et al. proposed the modified programmable FIR filter using Karatsuba formula [2]. Main advantage is that the n-tap filter will be implemented with three reduced dynamic range filters of N/2 Tap length. For the range in consideration the Karatsuba algorithm does not fare well in terms of area and complexity increased for higher multiplier when it is implemented for Karatsuba Filter. Once the number of bits is large, it affects glitches indirectly.

Tsao and Choi discussed the use of FFA algorithms based on polyphase Decomposition which are area efficient [16].

In this paper, we elaborated the use of CKA algorithm to implement an FIR filter with FFA algorithm and transpose form FIR filter. An n-tap filter of word length 2n in its general form is given by:

$$y(n) = \sum_{k=0}^{n-1} a(k)\, x(n-k) \qquad (6)$$

where x(n) is an input word length sequence, and a(k) is the Filter coefficient.

Equation (6) is expressed in Karatsuba algorithm [8] as:

$$Y_{HL}(n) = S(n)\, 2^{2N} + [U(n) - (S(n) + T(n))]\, 2^N + T(n) \qquad (7)$$

Formulation of a poly-phase decomposition of L-parallel FIR filter [11] is:

$$\sum_{K=0}^{L-1} Y_K Z^L Z^{-K} = \sum_{Q=0}^{L-1} X_Q Z^L Z^{-Q} \sum_{R=0}^{L-1} H_R Z^L Z^{-K} \qquad (8)$$

where K, Q and R = 0, 1, 2... L-1.

Using Karatsuba formula, the two input parameters for the filter design can be written as:

$$X_Q = X_{Q(H)}\, 2^N + X_{Q(L)}$$
$$H_R = H_{R(H)}\, 2^N + H_{R(L)} \qquad (9)$$

Substituting (9) in (8)

$$\sum_{K=0}^{L-1} Y_K Z^L Z^{-K} =$$
$$\sum_{K=0}^{L-1} \sum_{R=0}^{L-1} (X_{Q(H)}\, 2^N + X_{Q(L)})$$
$$(H_{R(H)}\, 2^N + H_{R(L)}) Z^{2L} Z^{-(R+Q)} \qquad (10)$$

We can derive auxiliary filters from (10) which make the reconstruction of CKA Algorithm easier.

$$S(n) = \sum_{Q=0}^{L-1} \sum_{R=1}^{L-1} X_{Q(H)} * H_{R(H)}$$

$$T(n) = \sum_{Q=0}^{L-1} \sum_{R=0}^{L-1} X_{Q(L)} * H_{R(L)}$$

$$U(n) = \sum_{Q=0}^{L-1} \sum_{R=0}^{L-1} \frac{(X_{Q(H)} + X_{Q(L)})}{(H_{R(H)} + H_{R(L)})} \qquad (11)$$

Simplifying the filter equations using auxiliary filters of (11) results in the reconstructed FIR Filter which is represented as

$$\sum_{P=0}^{L-1} Y^P Z^L Z^{-P} = H(n)2^{2N} + [K(n) - (H(n)+L(n)]2^N + L(n) \quad (12)$$

For two Parallel FIR Filters, (12) solves into

$$Y_0 + Y_1 Z^{-1} = \{((X_{0(H)} + X_{1(H)})2^N + (X_{0(L)} + X_{1(L)})Z^{-1}\}$$
$$* \{(H_{0(H)} + H_{1(H)})2^N + (H_{O(L)} + H_{1(L)})Z^{-1}\} \qquad (13)$$

This implies that:

$$Y_0 = (X_{0(H)}2^N + X_{0(L)})(H_{0(H)}2^N + H_{0(L)}) +$$
$$[(X_{1(H)}2^N + X_{1L})(H_{1(H)}2^N + H_{1(L)})]Z^{-2}$$

$$Y_1 Z^{-1} = [(X_{0(H)} + X_{1(H)})2^N + (X_{0(L)} + X_{1(L)})] *$$
$$[(H_{0(H)} + H_{1(H)})2^N + (H_{0(L)} + H_{1(L)})] - Y_0 \qquad (14)$$

To reduce the computational complexity, the parent filter is divided into three parts working in parallel.

IV. CKA AND CSK IN FIR FILTER DESIGN

The implementation of the n-tap filter and two parallel filters using the proposed algorithm are shown in Figs. 4 and 5, respectively. FIR coefficient and word length multiplication are replaced by CKA and CSK for N-bit unsigned multiplication. The Filter requires on ripple carry adder before the computation of filter coefficient followed by 3-parallel sub filter of word length N/2. After processing, the coefficient design is modelled by using 3N/2 multiplier and four more ripple carry adders as shown in Fig. 5. Consequently, one fourth of hardware cost is reduced compared to the traditional existing parallel filter [5]. Also, Cut-Set retiming is applied after the product computation to get faster result is as shown in Fig. 5.

V. RESULTS AND DISCUSSION

The proposed CKA and CSK Multipliers are implemented for varying bit lengths of 2^N, and the basic performance matrix is derived in terms of area, power, and operating frequency. Based on the results of the proposed multiplier, Transposed and FFA based symmetric FIR filter is implemented over a different coefficient in the efficient word lengths of 16 bits.

The performance matrix is also derived for the parent Karatsuba algorithm and the Default Optimal Multiplier used in Cadence RC Compiler (referred as Conventional Multiplier algorithm) which are used in comparing the efficiency of the proposed algorithm.
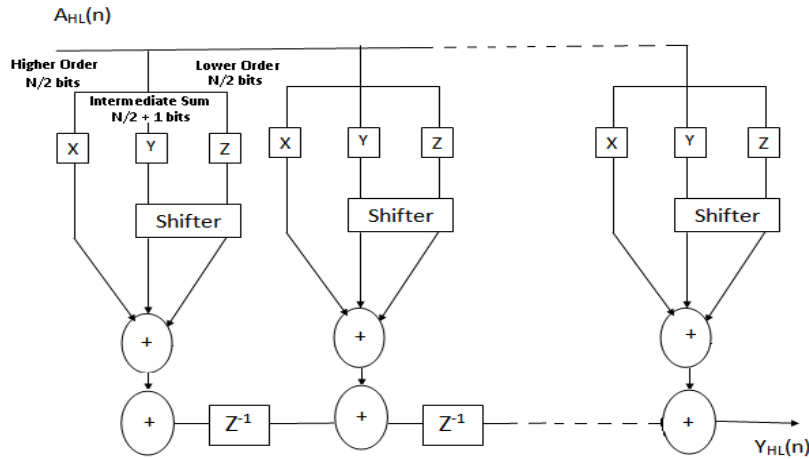

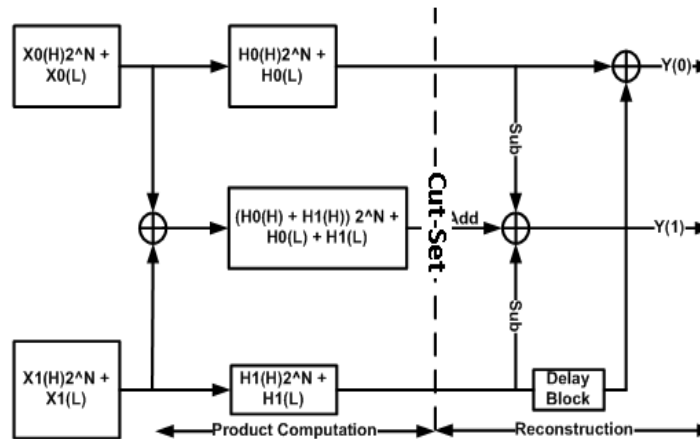Fig. 4 Karatsuba algorithm based Transposed FIR filter

Fig. 5 Two Parallel FIR Filters using CKA

Simulate and synthesis cycle is implemented by using Cadence Tool Suites - NC Sim for Simulation and RTL Compiler for Synthesis. The design is synthesized by using 90 nm CMOS Technology library. The synthesis used incremental step optimization flow for all the designs.

The Section V analyzes the results of the proposed CKA Multiplier, CSK Multiplier, and its FIR Filter implementation. The proposed architecture provides better results compared to the existing architecture [2], [3]. It is observed that the modified Transpose and FFA Filter synthesis result reduced the power consumption as well as significant area reduction [2,3]. By applying the aspect of backward retiming, a register is deleted from the outbound edge, and a new register is inserted in the cut-set line as shown in Figs. 1 and 5. It is evident that there is a considerable increase in speed of the proposed architecture. The synthetisation of 16, 32 and 64 Taps of Transpose and symmetric FFA filter is implemented for the word length of 16 bits. The performance evaluation results are summarised in Tables I and II. Table III shows less delay after applying retiming techniques for higher order bit multiplication. The proposed multiplication algorithm provides better performance than [15]. The proposed work scaled up-to N-bits value and it is suitable in real applications.

TABLE I
DELAY, AREA, AND POWER FOR TRANSPOSE FIR FILTER AND FFA IMPLEMENTATION USING CKA

No. of Taps T	Karatsuba FIR filter	Transpose FIR Filter	FFA Filter	Transpose FIR Filter with retiming	FFA Filter with retiming
Power (mW)					
64	116.78[2]	57.2	7.32	34.35	8.57
32	59.58[2]	27.5	3.7	16.78	3.7
16	32.08[2]	9.1	1.41	6.31	1.99
Delay (ns)					
64	1.6[2]	3.2	4.009	2.35	4.031
32	1.5[2]	3.1	4.01	2.35	4.03
16	1.47[2]	2.87	4.01	2.35	4.03

Block size 16 with different filter order with the proposed structure is compared with [2], [3]. The proposed block of FFA

using CSA and CKA gives a more than half of the savings in terms of APD for different word length compared to [16].

TABLE II
DELAY, AREA, AND POWER FOR TRANSPOSE FIR FILTER AND FFA IMPLEMENTATION USING CSA

No. of Taps T	Karatsuba FIR filter	Transpose FIR Filter	FFA Filter	Transpose FIR Filter with retiming	FFA Filter with retiming
Power (mW)					
64	116.78[2]	49.2	8.7	26.6	10.2
32	59.58[2]	23.8	4.4	13.58	4.4
16	32.08[2]	7.75	1.6	4.8	2.36
Delay (ns)					
64	1.6[2]	2.9	3.9	1.65	3.85
32	1.5[2]	2.95	3.86	1.65	3.9
16	1.47[2]	2.7	3.86	1.65	3.87

TABLE III
DELAY(nS) RESULTS FOR CKA AND CSA

No. of Bits	CKA	CKA + retiming	CSA	CSA + Retiming
64	8.28	5.37	4.2	2.7
128	16.2	10.22	6.9	4.2
256	31.3	17.52	9.6	6.2

VI. CONCLUSION

VLSI architecture of modified Karatsuba Algorithm exploited using different design model is introduced in this paper. The reusability of each sub-module is beneficial in area and power [2], [3], [16]. To improve the speed of the architecture, the recursion and retiming is implemented.

The evaluation of performance parameters is compared with the existing structure [2]. The proposed architecture experimental results significantly achieve low power consumption, especially for high dynamic range. When filter tap length increases, the proposed structure shows a good performance without applying retiming as well as with retiming concept.

REFERENCES

[1] A. Karatsuba and Y. Ofman, "Multiplication of many-digital numbers by automatic computers," Doklady Akad. Nauk SSSR, 1962, vol. 145, no. 2, pp. 293–294, 1962.

[2] Albicocco, P.; Cardarilli, G.C.; Pontarelli, S.; Re, M., "Karatsuba implementation of FIR filters," Signals, Systems and Computers (ASILOMAR), 2012 Conference Record of the Forty Sixth Asilomar Conference on, vol., no., pp.1111,1114, 4-7 Nov. 2012

[3] Basant Kumar Mohanty and Pramod Kumar, "High Performance FIR Filter architecture for fixed and reconfigurable applications" IEEE transcation on 2015.

[4] C.E Leiserson and J.B. Saxe, "Retiming synchronous circuitry," Algorithmica, Vol. 6, no. 1, pp. 5-35, 1991.

[5] Chao Cheng; Parhi, K.K., "Low- Cost Parallel FIR Filter Structures With 2-Stage Parallelism," Circuits and Systems I: Regular Papers, IEEE Transactions on, vol.54, no.2, pp.280, 290, Feb. 2007

[6] Chin-Bou Liu, Chua-Huang Huang, and Chin-Laung Lei, "Design and Implementation of Long-Digit Karatsuba's Multiplication Algorithm Using Tensor Product Formulation", The Ninth Workshop on Compiler Techniques for High-Performance Computing, 2003, pages 23-25.

[7] D. Knuth, "The Art of Computer Programming, Volume 2. Third Edition", Addison-Wesley, 1997. Section 4.3.3.A: Digital methods.

[8] Ehtiba, F.O.; Samsudin, A., "Multiplication and exponentiation of big integers with hybrid Montgomery and distributed Karatsuba algorithm," Information and Communication Technologies: From Theory to Applications, 2004. Proceedings. 2004 International Conference on, vol., no., pp.421, 422, 19-23 April 2004

[9] Gary C.T. Chow, Ken Eguro, Wayne Luk, Philip Leong, "A Karatsuba based Montgomery Multiplier", Embedded and Reconfigurable Computing Group, Microsoft Research, Redmond, USA, pp-1-4, Dec-2013.

[10] J. Monteriro, S. Devadas, and A.Ghosh, "Retiming sequential circuits for low power," International journal of high speed electronics and systems, Vol. 7, no.02, pp. 323-340, 1996.

[11] K.K Parhi, VLSI Digital Signal Processing Systems: Design and Implementation, New York, NY, USA: Wiley, 1999.

[12] Koc, Cetin K.; Erdem, Serdar S., (2003): A Less Recursive Variant of Karatsuba-Ofman Algorithm for Multiplying Operands of Size a Power of Two. Proceedings of the 16th IEEE Symposium on Computer Arithmetic, 1063-6889.

[13] N. Nedjah and L. Macedo Mourelle, "A reconfigurable recursive and efficient hardware for Karatsuba-Ofman's multiplication algorithm," in Proc. IEEE Conf. on Control Applications, Jun. 2003, pp. 1076–1081.

[14] Pramod Kumar Meher, "On Efficient Retiming of Fixed-Point Circuits," IEEE Transactions Journal.

[15] Shri Prakash Dwivedi, "An Efficient Multiplication Algorithm Using Nikhilam Method". ArXiv: 1307.2735v1 (cs.DS) 10 Jul 2013.

[16] Yu-Chi Tsao; Ken Choi, "Area-Efficient Parallel FIR Digital Filter Structures for Symmetric Convolutions Based on Fast FIR Algorithm," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, vol.20, no.2, pp.366,371, Feb. 2012.

S. Jalaja holds M. Tech degree in VLSI Design and Embedded System from Visvesvaraya Technological University (VTU). She is a member of IEEE and presently serving as an Assistant Professor in Electronics and Communication Engineering department at Bangalore Institute of Technology, Bangalore (Karnataka), India. She is currently pursuing the Ph.D. degree in Electronic and Communication engineering from the VTU, Belgaum. Her research interest includes Analysis of various algorithms to design different VLSI architectures and ASIC implementation for digital signal processing applications.

Dr. A. M. Vijaya Prakash is graduated B.E from UVCE Bangalore, of Bangalore University in the year 1992, Post graduated in M.E from SDMCET Dharwad of Karnataka University in the year 1997. He joined Bangalore institute of Technology in 1997 and he has since been teaching Electronics related subjects. He obtained his Ph.D. degree in 2012 from Dr.M.G.R University Chennai in the year 2012. He is actively guiding five Ph.D students, along with that he has been actively associated with PG and UG projects in the area of VLSI and Image Processing. He has around 20 technical paper publications in journals and international conferences. Currently he has been working as Professor in Electronics and Communication Engineering Department, Bangalore Institute of Technology Bangalore. His research interests are Low Power VLSI, Image Processing, Reversible Logic and Nono Technology. He is a member of IMAPS and ISTE.