JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

# Design Power Efficient and High Speed Multipliers using of Approximate Compressors

**[1]Monika Singh, [2]Prashant Chaturvedi**
[1]Research Scholar, [2]Professor
[1&2] Department of Electronics & Communication Engineering,
[1&2] Lakshmi Narain College of Technology, Bhopal, India

***Abstract***— Multiplier is an essential block of a computing device. Multipliers are commonly used for signal and image processing applications where multipliers are used to perform various tasks like convolution, correlation and filtering. Multipliers are not only a high delay component but also dissipate high amount of power. It is necessary to increase the speed of multipliers as the demand of high speed processors is increasing. Moreover, multipliers are an integral part of any processor and are utilized by the processors to complete signal processing tasks. This paper presents the new compressors for partial product reduction. Compressors are used to reduce partial products. Compressors take more inputs at a time and are able to process partial products in a faster way compared to conventional adders. Use of compressors in the partial product reduction helps to minimize the delay but not area consumption. Our proposed compressors aim to reduce the delay and power consumption together. Single and multicolumn compressors are proposed and used in the partial product reduction. Experimental results showed that our proposed approaches are performing better than existing approaches.

Keywords—***Urdhwa Multiplier, 4:3 Compressor, 7:4 compressor and Xilinx Vertex family***

## I. INTRODUCTION

By mid 20thcentury, due to the technology advancements in semiconductor device fabrication, it is seen that the semiconductor devices could perform the functions of vacuum tubes [1, 2]. The development of small chips by incorporating large amounts of minute transistors was an enormous improvement over the manual assembly of discrete electronic components. With the invention of Integrated Circuits(IC) the approach towards circuit design changed which ensured mass production capability and reliability. This led to the rapid adoption of standardized ICs in place of designs using discrete transistors. There are two main advantages of ICs over discrete circuits - cost and performance. The production cost of an IC is less, as the chips with the entire circuit is printed as a single entity by photolithography and not constructing a separate component at a time.

Performance of an IC is also high, since the components switch quickly and consume little power as they are small and close together. With the advent of ICs in this modern era, all the technology developments aim at ease of use with improved computational capability. Almost all applications involve basic arithmetic operations among which, addition and multiplication are the most important and widely used functional blocks. Hence, adders and multipliers play a major role in these applications [3]. To bring about compactness and faster operation, the normal arithmetic mechanism is replaced by many techniques that promise minimum hardware with early and correct results.

Many adders like carry save adder, carry select adder, carry look ahead adder, carry skip adder and many others were developed with an aim to result in faster outputs and less hardware. Improvising adders alone did not meet the requirements of the modern world. So, the complex and time consuming blocks namely, the multiplier blocks were concentrated for improving the efficiency further. This chapter gives an overview of multiplier basics, their types and the effective way for optimization [4, 5].

With the arrival of VLSI technology, processors have greatly improved their performance in terms of area and power. Its broad usage has made it as a most common module in almost all applications. Computation using these processors was prevalent and most common over the past few decades. These computations are commonly done using algorithms. These algorithms can be implemented by two methods namely, the conventional hardware design and software program method. In the first technique, the circuit is implemented which performs the desired function effectively but the disadvantage is that, the circuit cannot be used for implementing another function with the same architecture. The second approach is the software program method, where the functionality can be

altered through codes, but the performance of the system reduces compared to the first method [6, 7].

## II. MULTIPLIER

Array multiplier is well known due to its regular structure. The Multiplier is based on add and shift algorithm. The multiplicand is multiplied with each one bit of the multiplier to generate the partial products. The partial products for each bit of the multiplier is rearranged by shifting their position and then added. The addition can be performed with normal carry propagate adder. For an N bit length multiplier it requires N-1 adders [8, 9].

Partial products are summed with carry save adder for fast multipliers. Typically it has a delay of 2 inverters which is independent of the partial product's width. The CPA tends to have a delay of more than 15 inverters, which depends on the width of the partial products. The Fig. 1 shows the structure of a 4×4 array multiplier to add the Partial products implemented with the CSA for unsigned numbers. A carry save redundant form is formed for the first partial product row conversion. Then the consecutive rows are summed up with these redundant results generated from the previous operation. The CSA directly produces the least significant N bit product from the addition of the final row. The most significant bits are arrived in the form of carry save redundant and requires Carry Propagate Adder of M bits.

As shown in the Fig. 1 the CPA is implemented as a ripple carry adder at the final level. This method of array arrangement is regular, easy in designing and for layout formation. The critical paths of the array multiplier are the mid columns in the sequence assuming that the carry is faster than the sum generation. The register could be placed between the rows for the pipelining of the adders. The major concern is to design a compact CSA. The compactness could reduce the internal wiring and thus the capacitance of the wire. In a CSA design the entire sum and the carry will have approximately equal delay. The delay could be compensated by connecting the fastest carry with the sum. Even though the first row of partial products are added with 0s to have a regular structure they are insufficient. The adder propagation delay can be drastically reduced by adding the first three rows of partial product together which reduces it to two rows. Even the delay could be further reduced in the array multiplier by replacing the last row with a faster CPA like tree or look ahead adder.

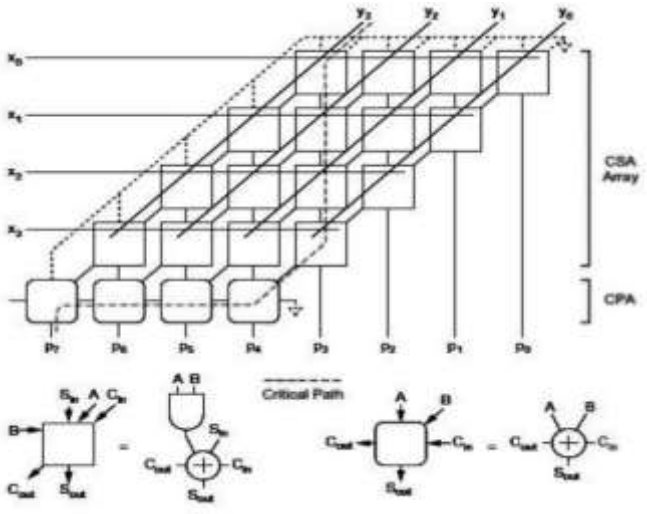



Fig. 1: Array Multiplier

## III. URDHWA MULTIPLIER

Multipliers are the blocks in DSP processors which requires large amount of storage and processing complexity. Vedic mathematics has well known an ancient method for mathematical operations which are easy to use Urdhwa triyambakam is a method used for multiplication calculations. It takes the advantage of vertical and crosswise calculations. In this technique multiplication of terms is obtained by using simple AND logical operations, full adders and half adders. In this paper a 8 bit Urdhwa multiplier is used. For this purpose of addition, we have been used various designs of adders and compressors. Here, these adders and compressors are deployed in such a way so as to provide optimized design of multiplier in terms of delay. Four half adders, two full adders, five 7: 4 compressor and ten 4:3 compressors has been used.

We have presented three designs for Urdhwa multiplier. These three designs vary in terms of 4:3 compressor designs. First design uses the simple 4:3 compressor which uses two full adders. Second design uses four XOR gates and two multiplexers (2×1). Third design of 4:3 compressors is two XOR-XNOR gates and four multiplexers (2×1). First design of 4:3 compressors is simplest one, but more delay. Next design has lesser delay as compared to the first design at the cost of higher complexity. But the last which is our proposed design has least amount of delay but has greatest complexity and occupies maximum space among all of them. We have designed 7:4 compressor using different designs of 4:3 compressor thus we got three variations of 7:4 compressor.

**• 4:3 Compressor**

To add binary numbers with minimal carry propagation we use compressor adder instead of other adder. Compressor is a digital modern circuit which is used for high speed with minimum gates requires designing technique. This compressor becomes the essential tool for fast multiplication adding technique by keeping an eye on fast processor and lesser area.

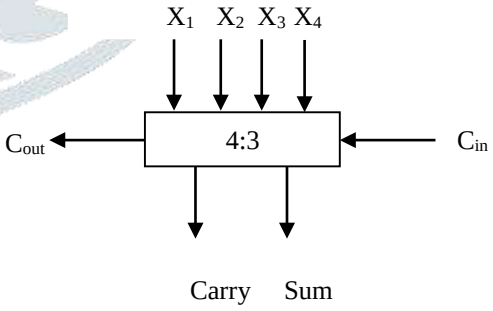
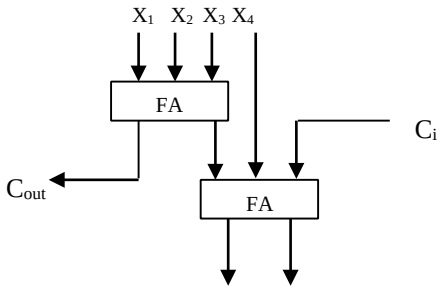



Fig. 2: Block Diagram of 4:3 Compressors

4:3 compressors are capable of adding 4 bits and one carry, in turn producing a 3 bit output. The 4:3 compressor has 4 inputs $X_1$, $X_2$, $X_3$ and $X_4$ and 2 outputs Sum and Carry along with a Carry-in (Cin) and a Carry-out (Cout) as shown in Figure 1. The input Cin is the output from the previous lower significant compressor.

The Cout is the output to the compressor in the next significant stage. The critical path is smaller in comparison with an equivalent circuit to add 5 bits using full adders and half adders. Similar to the 3-2 compressor the 4:3 compressor is governed by the basic equation the standard implementation

of the 4-2 compressor is done using 2 Full Adder cells as shown in Figure 2(a). When the individual full Adders are




broken into their constituent XOR blocks, it can be observed that the overall delay is equal to 4*XOR.

$$X_1 + X_2 + X_3 + X_4 + C_{in} = sum + 2*(Carry + C_{out})$$

$$Sum = X_1 \oplus X_2 \oplus X_3 \oplus X_4 \oplus C_{in}$$

$$C_{out} = (X_1 \oplus X_2).X_3 + (X_1 \oplus X_2).X_1$$

$$C_{arry} = U.C_{in} + (X_1 \oplus X_2 \oplus X_3 \oplus X_4).X_4$$

Where

$$U = X_1 \oplus X_2 \oplus X_3 \oplus X_4$$

Carry Sum

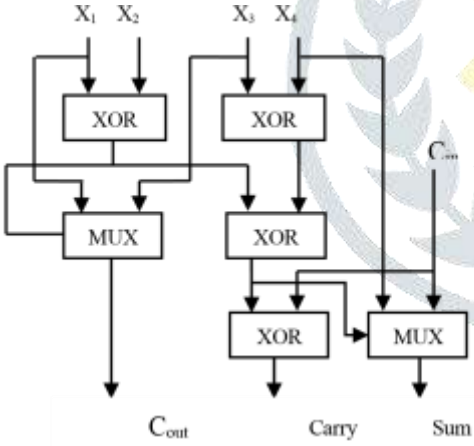



Fig. 4: 4:2 Compressors using XOR Gate

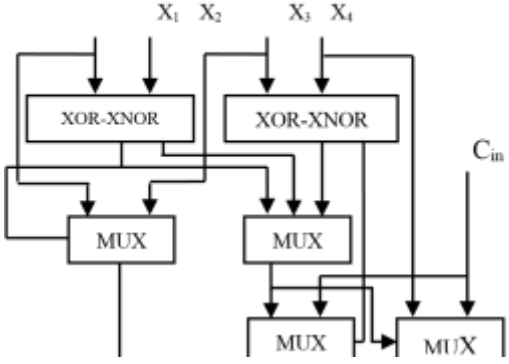



Figure 3: 4:3 Compressors using Full Adder

The standard implementation of the 4:3 compressor is done using 2 Full Adder cells as shown in Figure 2(a). When the individual full Adders are broken into their constituent XOR blocks, it can be observed that the overall delay is equal to 4*XOR.

## IV. PROPOSED TECHNOLOGY

The block diagram in Fig. 4 shows the existing architecture for the implementation of the 4:3 compressor with a delay of 3*XOR. The equations governing the outputs in the existing architecture are shown below

the SUM output gets the select bit before the inputs arrive and thus replacing some XOR blocks with multiplexer's results in a significant improvement in delay. Also the MUX block at

Fig. 5: 4:2 Compressors using XOR -XNOR Gate

thus the transistors are already switched by the time they arrive. This minimizes the delay to a considerable extent. This is shown in Fig. 5.

The equations governing the outputs in the proposed architecture are shown below

$$Sum = (X_1 \oplus X_2).X_3 + (X_1 \oplus X_2).(X_3 \oplus X_1).C_{in}$$

$$C_{out} = (X_1 \oplus X_2).X_3 + (X_1 \oplus X_2).X_1$$

All the designs have been captured by VHDL and the functionality is verified by RTL and gate level simulation shown in table 1. Designs are implemented on a Xilinx Spartan 3 FPGA using VHDL as the RTL language with the help of Xilinx ISE design suite 14.1.

TABLE 1: Device utilization summary (Vertex-4) of 4:3 Compressors

| Design | No. of Slices | No. of 4 input LUTs | MCPD (ns) |
|---|---|---|---|
| 4:3 Compressor | 3 | 6 | 6.257 |
| 4:3 Compressor using XOR gate | 2 | 4 | 5.663 |
| 4:3 Compressor using XOR-XNOR gate | 2 | 3 | 5.280 |

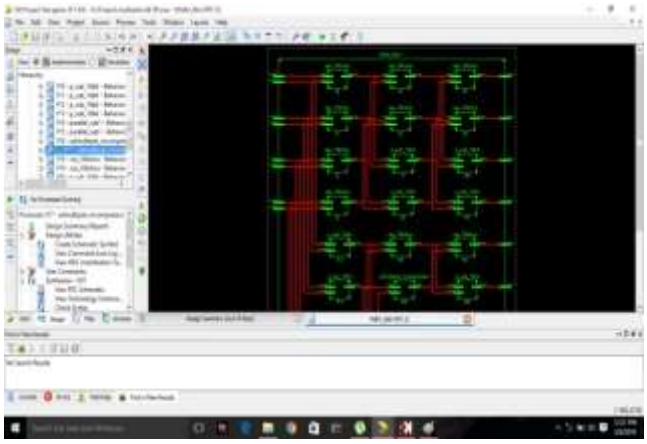
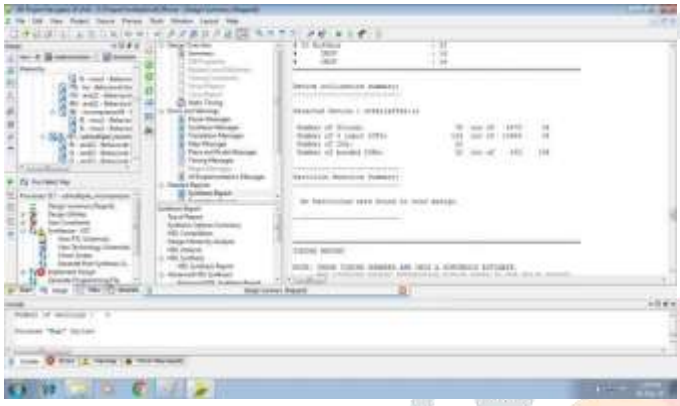


Fig. 7: RTL view of Modified Compressor based Urdhwa Multiplier



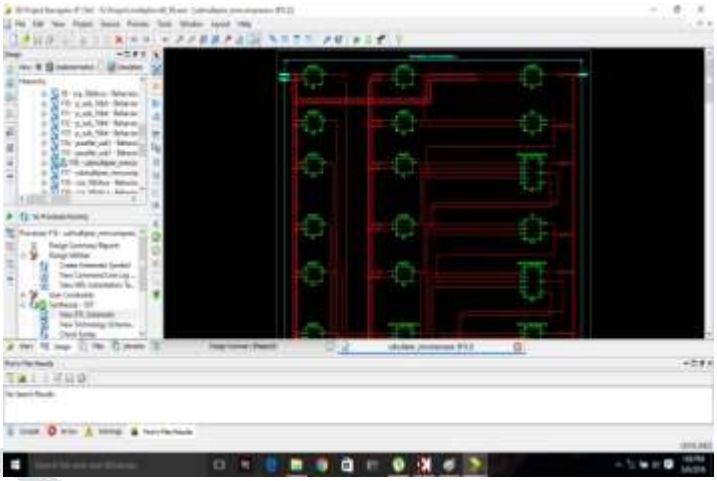
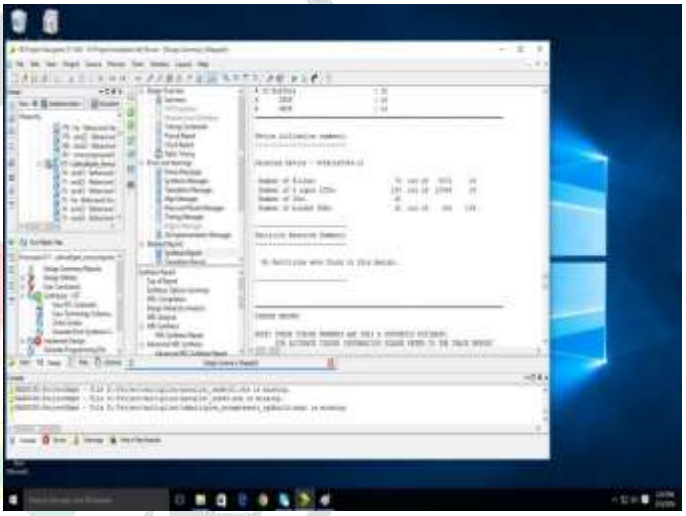


Fig. 9: RTL view of XOR-XNOR Compressor based Urdhwa Multiplier



• **7:4 Compressor**

Similar to its 4:3 compressor counterpart, the 7:4 compressors as shown in Figure 3, is capable of adding 7 bits of input and 2 carry's from the previous stages, at a time. In our implementation, we have designed a novel 7:4 compressor utilizing two 4:3 compressors, two full adders and one half adders. We have designed 7:4 compressor using different designs of 4:3 compressor thus we got three variations of 7:4 compressor.

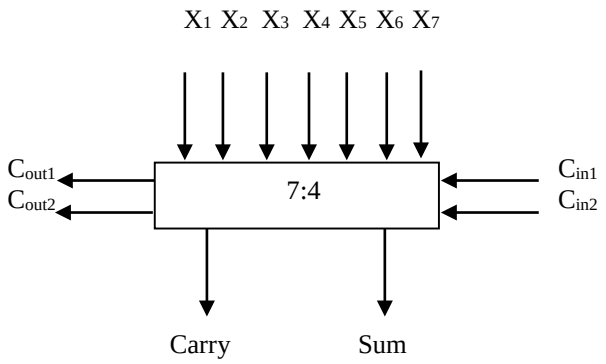



Fig. 6: Block Diagram of 7:4 Compressors

## V. SIMULATION RESULT

It is a hardware description language that can be used to describe the structure and/or behaviour of hardware designs and to model digital systems. Having designed the various DSP configurations, we now proceed to the software synthesis of this designs using VHDL.

In the fig. 7, input of the modified compressors based multiplier is represented by a [7:0], b [7:0] and output of the modified compressors based multiplier is represented by c [15:0] respectively.

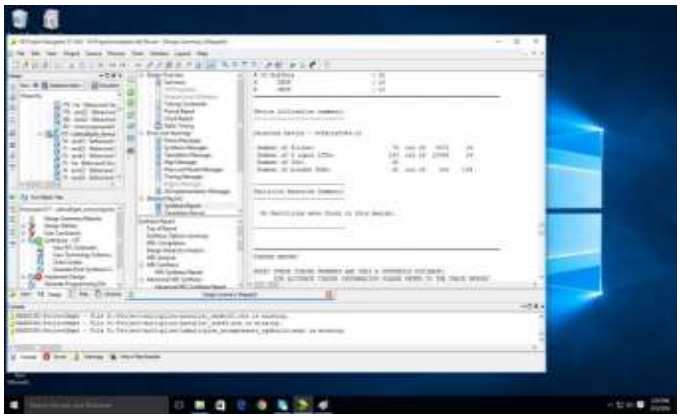


Fig. 8: Device Utilization Summary of Modified Compressor based Urdhwa Multiplier

In the fig. 9, input of the XOR-XNOR compressors based Urdhwa multiplier is represented by a [7:0], b [7:0] and output of the XOR-NOR compressors based Urdhwa multiplier is represented by c [15:0] respectively.

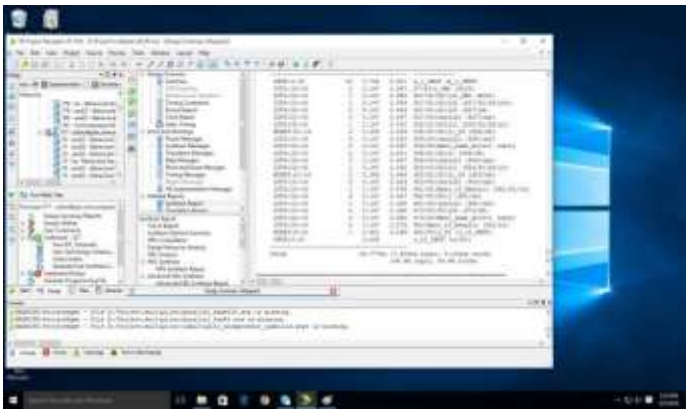


Fig. 9: Device Utilization Summary of XOR-XNOR Compressor based Urdhwa Multiplier

## VI. CONCLUSION

This paper proposes new designs of various compressors for high speed and low delay. The design are verified and the requirements. This thesis presents the design of accurate and approximate compressors for various multipliers. Results show that our proposed multipliers can provide low power and high speed. This multiplier is utilized for image processing tasks such as contrast enhancement and diagnosis of cancer cells. From the analysis of the results, it is found that the proposed structures perform better than state of the art multipliers.

REFRENCES


[1] Tianqi Kong and Shuguo Li, "Design and Analysis of Approximate Compressors for High- Accuracy Multipliers", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 29, No. 10, October 2021.

[2] A. G. M. Strollo, E. Napoli, D. De Caro, N. Petra, and G. D. Meo, "Comparison and extension of approximate 4-2 compressors for lowpower approximate multipliers," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 67, no. 9, pp. 3021–3034, Sep. 2020.

[3] Z. Gu and S. Li, "A Division-Free Toom–Cook Multiplication-Based Montgomery Modular Multiplication," IEEE Trans. Circuits Syst. II Express Briefs, vol. 66, no. 8, pp. 1401–1405, Aug. 2019.

[4] R. Liu and S. Li, "A design and implementation of montgomery modular multiplier," in Proceedings - IEEE International Symposium on Circuits and Systems, 2019, vol. 2019-May.

[5] Ranjan Kumar Barik, Manoranjan Pradhan and Rutuparna Panda, "Time Efficient Signed Vedic Multiplier Using Redundant Binary Representation", Journal Engineering, Vol. 17, Issue 3, pp. 60–68, march 2017.

[6] Basant Kumar Mohanty, and Pramod Kumar Meher, "High-Performance FIR Filter Architecture for Fixed and Reconfigurable Applications", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 78, No.06, April 2016.

[7] L kholee phimu and Manoj Kumar, "Design and Implementation of Area Efficient 2-Parallel Filters on FPGA using Image System", International Conference on Energy, Communication, Data Analytics and Soft Computing, IEEE 2017.

[8] Vijaya Lakshmi Bandi, "Performance Analysis for Vedic Multiplier Using Modified Full Adders", International Conference on Innovations in Power and Advanced Computing Technologies, IEEE 2017.

[9] G. Challa Ram and D Sudha Rani, "Area Efficient Modified Vedic Multiplier", International Conference on Circuit, Power and Computing Technologies (ICCPCT), 2016.

[10] S. P. Pohokar, R. S. Sisal, K. M. Gaikwad, M. M. Patil and Rushikesh Borse, "Design and Implementation of 16 x 16 Multiplier Using Vedic Mathematics", International Conference on Industrial Instrumentation and Control (ICIC) College of Engineering Pune, India, PP. No. 01-06, 2015.

[11] Gokhale and P. D. Bahirgonde, "Design of Vedic Multiplier using Area-Efficient Carry Select Adder", 4th IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI-2015), Kochi, pp. 10-13, August 2015, India.

[12] G. Gokhale and Mr. S. R. Gokhale, "Design of Area and Delay Efficient Vedic Multiplier Using Carry Select Adder", 4th IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI2015), Kochi, pp. 10-13, August 2015, India.

[13] Amina Naaz. S, Mr. Pradeep M.N, Satish Bhairannawar and Srinivas halvi, "FPGA Implementation Of High Speed Vedic Multiplier using CSLA For Parallel Fir Architecture", 2nd International Conference on Devices, Circuits and Systems (ICDCS), IEEE 2014.