# FPGA Implementation and Analysis of Different Multiplication Algorithm

Manoj M. Kamble
Student (E&TC)
K.K.Wagh IEER
Nasik, India

Sunita P. Ugale, PhD
Associate Professor (E&TC)
K.K.Wagh IEER
Nasik, India

## ABSTRACT

Many of the today's real time signal processing algorithm included multiplication as its processing heart. In case of signal and image processing, it mostly used functional unit. In this paper we are simulating different multiplication algorithm with their effective architecture. Also paper introducing new multiplication technique using barrel shifter which gives some sort of modification in previously described shift and add multiplication algorithm. Research targeting mainly four algorithms as Vedic vertical crosswise multiplication algorithm, Array multiplier, Shift and add multiplier, Wallace tree multiplier. Further work will carried comparative study of different multiplier with respect to some parameters like logical resources used, delay, power consumption and area. For implementation and parametric analysis, experimental setup uses sparten-3 XC3S400 FPGA as a hardware platform, VHDL coding language for hardware description. Xilinx ISE-simulation tool has many inbuilt compatible facility for parameter analysis like XPE for power analysis. Finally Paper comprises simulation results for 8-bit, 16-bits and 32-bits each of above mentioned multiplier.

## Keywords

Array multiplier, Vedic, Wallace tree, shift and add, Barrel shifter.

## 1. INTRODUCTION

Now days, for many application specific processor implementations such as real time signal processing, image processing, encryption, data manipulation requires high speed integrated circuit. Many signal processing algorithm make use of math-multiplier. There are different multiplication algorithm presently available and have implemented on SOC. Booth multiplier, Wallace tree multiplier, Karatsuba multiplier, Braun algorithm are more popular multiplication algorithm [2]. But many of them having their own restriction in terms of speed, on chip area, use of logic gates, energy, cost per cell, so in our project we are analyzing these many parameters and going to introduced one modified multiplication algorithm based on barrel shifter.

Barrel shifter has ability to shift any number to left or right (as per requirement) by N-position within only one clock cycle [1]. If it analyze this convenient barrel shifter behavior can used to implement multiplier that can multiply multiplicand number with multiplier number which should be perfect representation in power of 2 [3]. Research proposes new method based on barrel shifter used for multiplication of numbers which are not perfect two power representation.

## 2. DESIGN SUMMERY OF EXISTED ALGORITHMS

### 2.1 Vedic vertical crosswise multiplier

Multiplication methods that many of processors are using today has inspired by Vedic multiplier introduced in Indian Vedas with different 16 sutras. Vedic multiplier uses bit-wise multiplication with simultaneous product term finding and it's column-wise addition. It is one of the best benchmark for fast multiplication algorithm. Paper describes vertical crosswise multiplication method with effective use of its basic block [4].

Paper describes design of basic Vedic block which will perform two bit operation, then four such basic block can used for performing operation on 4-bit data which can be defined by block diagram as given below-

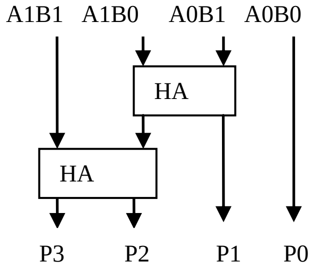

**Fig.1. 2X2 bit Vedic basic element**

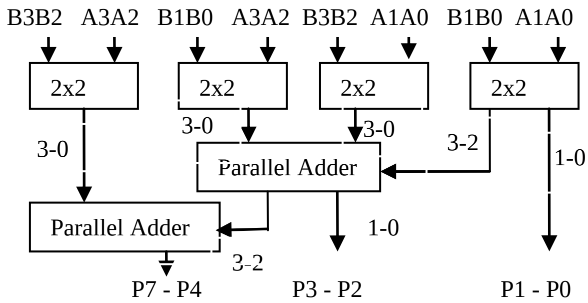

**Fig.2. 4X4 bit Vedic using 2x2 bit element**

Likewise to design further higher bit multiplier it need to use four blocks from its predecessor with connectivity as shown in figure 2.

## 2.2 Generic Array Multiplier

Array multiplier is one of the bit-by-bit multiplication implementation approaches. Partial products are first generated and stored in memory from where it given to array of summation. As it performs operation on data in array form, so dedicated memory space is first prior to its implementation. Broun has described array multiplication with carry propagation adder [2]. It is seems to be as a big pour of HA and FA with specific pattern which remain same to all bit configurations as below-

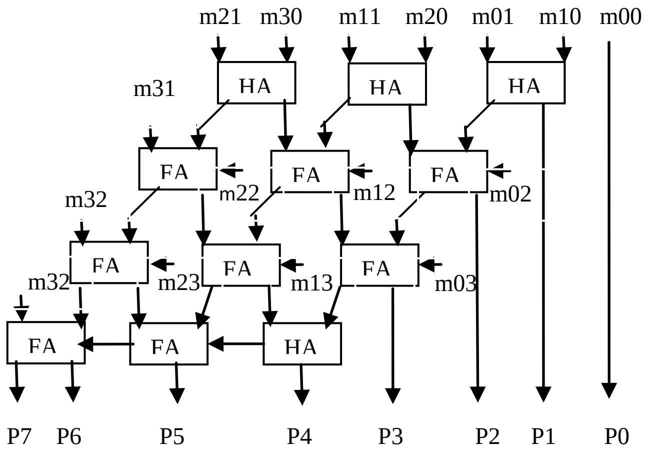

**Fig.3. 4x4 bits Array multiplier**

Thus, first row will contain N-1 of HA and last row is for final carry adder, in between stages filled by FA.

## 2.3 Generic Shift and add multiplier

Shift and add algorithm takes use of bit shifter and parallel adder. Both multiplicand and multiplier have dedicated locations. It checks for each bit of multiplier and take decision about what to do with multiplicand [2]. We have implemented this algorithm by FSM modeling so as to get advantages of minimum latency, delay and power as below –

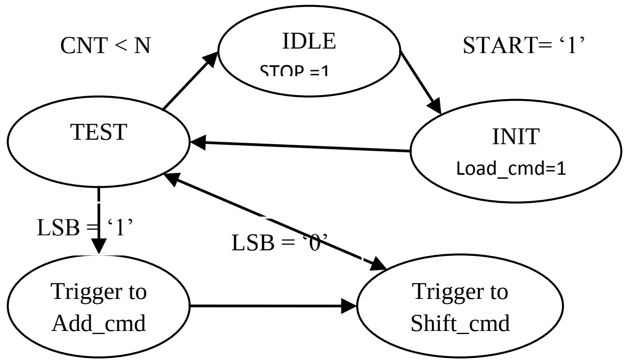

**Fig.4. FSM diagram for Shift & Add multiplier**

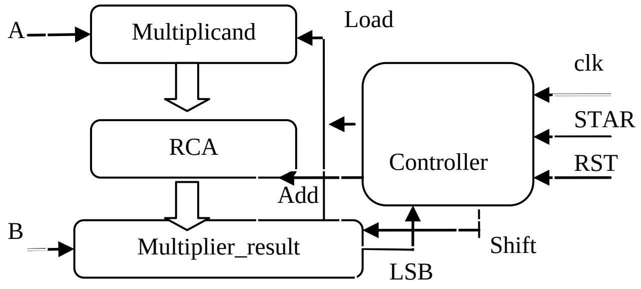

**Fig.5. Block Diagram of Shift and Add multiplier**

## 2.4 Wallace tree multiplier

Wallace tree multiplier has ability to reduce number of total partial product terms. It make use of Booth Recode radix-x (x = 2, 4, 8) method for generating minimum partial product terms and 3:2, 4:2, 5:2 compressors to reduce addition further [6]. Because of this dual stage structure, it is more complex to design but reliable in terms speed. Following figure shown its block diagram –

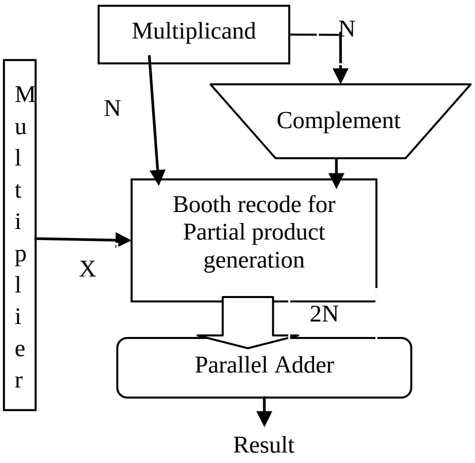

**Fig.6. Wallace tree with Booth encoder**

## 2.5 Multiplier using Barrel shifter

Here proposed algorithm makes use of barrel shifter; recursively for final computation. As it depicted above that barrel shifter can effectively used for number multiplication those are perfectly represented in power of two. But it can be used recursively if stated condition is not satisfied. Just below we have listed step to be followed by our algorithm-

1. Suppose X and Y are two each of N-bit numbers.
2. Count the number of ones in each number.
3. Number with smaller ones count will treat as a multiplier and other will be multiplicand.
4. Finding further successive power two representations of multiplier.
5. Now shift multiplicand by power of two to left
6. Give shifted result to adder
7. Repeat step 4 to 7 till power of two become zero.

In step 3, selecting less ones count number as a multiplier so as to avoid unwanted shifting operation per clock cycle.

Algorithm explanation with example gives as below-

Let, consider two number as-

A=11 (bX1011), B=9(bX1001), ones count OA(3) and OB(2)

B having two power indexes as $2^2 + 2^0$

Shifting of A by 3 = (bX01011000) (1)

Shifting of A by 0 = (bX00001011) (2)

Final result will be as = (1) + (2) = (bX01100011) = 99.

Flowchart of algorithm gives clear idea of operation sequence-

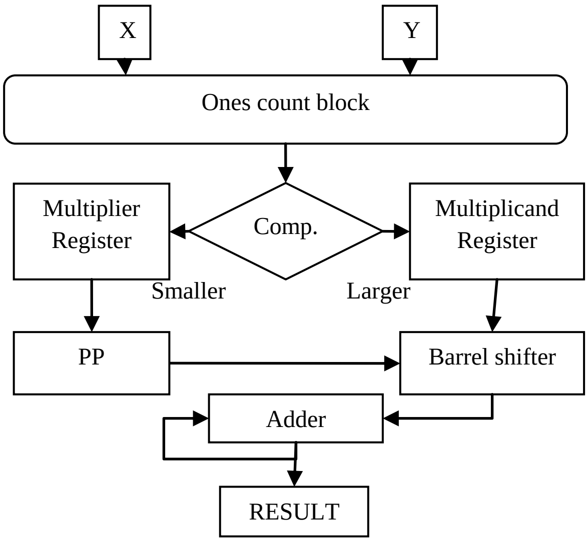

**Fig.7 Flowchart for multiplier using barrel shifter**

## 3. SIMULATION RESULTS

All the above mentioned multiplier that have been simulated for different bit configuration and come out with following result shown table-1 –

Let's see, if it compare all multiplier at same bit configuration say 32 – bits then shift and add multiplier having more advantages except delay. Wallace tree multiplier has high speed but uses greater logical resources. Out of the above multiplier, Vedic based algorithm has moderate parameter characteristics. Graphical parametric comparison for 32-bits of multiplier given below-

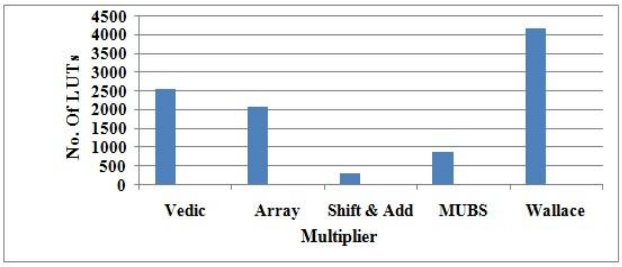

**Fig.8. Resource (LUTs) usage by multipliers**

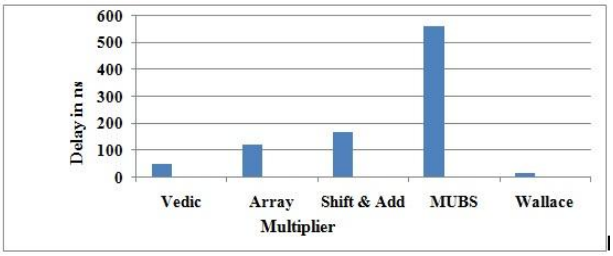

**Fig.9. Delay offered by multipliers**

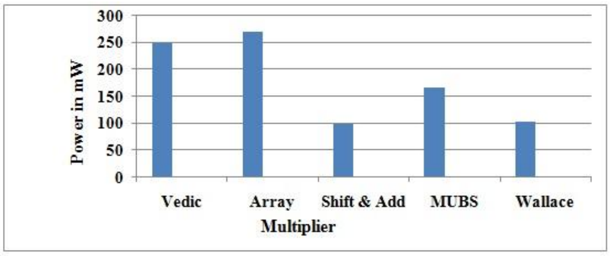

**Fig.10. Power consumption by multipliers**

**Table 1. Parameter comparison**

| Multiplier | Bits | No. of slices | No. of LUTs | Delay(ns) | Power (mW) |
|---|---|---|---|---|---|
| Vedic Math | 4 - bits | 19 | 33 | 19.22 | 62.42 |
| Array Multiplier | | 18 | 33 | 17.47 | 62.15 |
| Shift & Add | | 33 | 43 | 16.98 | 65.96 |
| Proposed multi. | | 37 | 71 | 31.96 | 73.18 |
| Wallace Tree Multiplier | | 14 | 31 | 6.32 | 71.21 |
| | | | | | |
| Vedic Math | 8– bits | 81 | 146 | 27.28 | 72.32 |
| Array Multiplier | | 69 | 133 | 32.314 | 73.49 |
| Shift & Add | | 50 | 69 | 35.61 | 70.22 |
| Proposed multi. | | 93 | 161 | 95.52 | 87.01 |
| Wallace Tree Multiplier | | 71 | 151 | 8.78 | 84.31 |
| | | | | | |
| Vedic Math | 16-bits | 346 | 622 | 36.44 | 108.29 |
| Array Multiplier | | 268 | 525 | 61.24 | 113.82 |
| Shift & Add | | 88 | 132 | 80.88 | 79.97 |
| Proposed multi. | | 222 | 393 | 236.96 | 113.20 |

| | | | | | |
|---|---|---|---|---|---|
| Wallace Tree Multiplier | | 449 | 756 | 10.13 | 95.25 |
| **Multiplier** | **Bits** | **No. of slices** | **No. of LUTs** | **Delay(ns)** | **Power (mW)** |
| Vedic Math | 32–bits | 1427 | 2566 | 46.27 | 248.03 |
| Array Multiplier | | 1050 | 2077 | 117.957 | 270.48 |
| Shift & Add | | 173 | 272 | 167.104 | 97.65 |
| Proposed multi. | | 495 | 857 | 562.24 | 165.44 |
| Wallace Tree Multiplier | | 2142 | 4167 | 13.38 | 102.78 |

## 4. CONCLUSION

Paper has covered study and logic analysis of different multiplication algorithm for its FPGA implementation. Deep study of VHDL or such a simulation language helpful for efficient hardware description of FPGA based algorithm. By the study, it is clear that Indian Vedic math sutra gives backbone for other logical implementation. Every algorithm has its own restriction in terms of power, cost and speed.

1. Chips which are having less accessing area will select shift and add multiplication algorithm having less logical element use, while Wallace tree can be prescribed for chips having large area.

2. From the delay analysis, Wallace tree algorithm based on Booth recoder is good for fast application implementation. Array multiplier is very slow as compare to other algorithms.

3. Power consumption of shift and add algorithm is less because of less number of signal switching per operation, while Vedic having more power consumption.

4. Here, from the above exercise of algorithm analysis and FPGA implementation, selection of multiplication algorithm is depends on application.

Thus, research work involves simulation and implementation of most commonly used multiplication algorithms on same platform. If work carries for higher bit configuration algorithms then there will be clear distinction between them. Depends on user operation one can prefer fast algorithm like Vedic, Wallace or low design resources algorithm like shift and add

## 5. ACKNOWLEDGMENT

The authors would like to acknowledge the support of K.K.Wagh IEER, Nasik for providing the research facilities.

## 6. REFERENCES

[1] M. Seckora, "Barrel Shifter or Multiply Divide IC Structure," U.S. Patent 5,465,222, November 1995.

[2] Cem Ergun, seminar at Eastern Mediterranean University on "Multiplication & Division Algorithms".

[3] P. Saha, A. Banerjee, P. Bhattacharyya, A. Dandapat, 2011 "High Speed ASIC Design of Complex Multiplier Using Vedic Mathematics" IEEE Students' Technology Symposium.

[4] Mrs.Toni J.Billore, Prof.D.R.Rotake May-Jun. 2014 "FPGA implementation of high speed 8 bit Vedic Multiplier using Fast adders" IOSR Journal of VLSI and Signal Processing, PP 54-59.

[5] A. A. Karatsuba: The Complexity of Computations. Proceedings of the Steklov Institute of Mathematics.

[6] H. Bansal, K. G. Sharma, T. Sharma "Wallace Tree Multiplier Designs: A Performance Comparison Review" Innovative Systems Design and Engineering, 2014.

[7] M. Young, the Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[8] S. Vaidya, D. Dandekar "DELAY-POWER PERFORMANCE COMPARISONOF MULTIPLIERS IN VLSI CIRCUIT DESIGN" International Journal of Computer Networks and Communications (IJCNC), Vol.2, No.4, July 2010.

[9] Kripa Mathew, S. Asha Latha, T. Ravi "Design and Analysis of an Array Multiplier Using an Area Efficient Full Adder Cell in 32nm CMOS Technology", The International Journal of Engineering and Science, volume 2, 2013.

[10] Ch. Harish Kumar, "Implementation and Analysis of Power, Area and Delay of Array, Urdhva, Nikhilam Vedic Multipliers", International Journal of Scientific and Research Publication, volume-3, January 2013.

[11] Sweta Khatri, Ghanshyam Jangid, "FPGA Implementation of 64-bit Fast multiplier using barrel shifter", IJRASET, volume-2, July 2014.

[12] A.Wasil Raseed Ahmed, "FPGA Implementation of Vedic multiplier using VHDL",

[13] Asmita Haveliya, "FPGA Implementation of Vedic Convolution Algorithm", IJERA, volume-2, 2012.

[14] Jagdeshwar Rao M and Sanjay Dubey, "A High Speed Wallace Tree Multiplier Using Modified Booth Algorithm for Fast Arithmetic Circuits", IOSRJECE, volume- 3, Sep. 2012.

[15] B. Dinesh, V. Venkateshwaran, P. Kavinmalar, "Comparison of Regular and Tree based Multiplier Architectures with Modified Booth Encoding for 4-bits on Layout Level using 45nm technology", IOSR Journal of VLSI and signal processing, May-2014.