ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Issue 4, April 2018

# FPGA Implementation of Approximate Multiplier with Image Sharpening Applications

Lidiya Babu[1], Lekha N M[2]
M Tech Scholar, Electronics & Communication Engineering Dept,Sahrdaya College of Engineering & Technology,Thrissur,India [1]
Asst Professor, Electronics & Communication Engineering Dept, Sahrdaya College of Engineering & Technology Thrissur, India [2]

**Abstract**: In this paper proposing an approximate multiplier that is of very high speed hence it will be an efficient one. Here the method is to round the numbers that need to multiply to the nearest exponent of two. By this way the complex part of multiplication can be omitted thus can improve the speed. This approximate multiplier is applicable to both signed and unsigned number multiplication. In this paper signed and unsigned operations are doing but finally implementing in the signed operation application i.e., image processing applications like image sharpening. The multiplication operation is implementing in FPGA SPARTAN kit. The efficiency of this approximate multiplier is analysing by the comparison with the accurate multipliers.

**Keywords**: Multiplier, Unsigned, FPGA, SPARTAN, Sharpening

## I. INTRODUCTION

Minimization of energy is one of the main design requirements in any electronic systems, especially the portable devices such as smart phones, tablets, and different gadgets [1]. In the design of digital signal processing and other applications multiplier is one of the main blocks. Many researches are going on to design multiplier with high-speed, low power consumption, regular structure, such that it occupies less area for compact VLSI implementation. Many algorithms where found out by different researchers in the past to do multiplication process. Every algorithm offers its own advantages and having trade-off between themselves by means of their speed, area, power consumption and circuit complexity.

Add and shift multiplication process is the common procedure in all multiplications. The main factor in parallel multipliers that determines the performance of the multiplier is the number of partial products being added. By the use of increased parallelism, the amount of shifts between intermediate sums and the partial products to be added. It will increase reduction in speed. Depending on the nature of application the selection of a parallel or serial multiplier is come into point. If we are considering the case of FIR filter, multipliers are the main component main component.

A system's performance is generally determined by the performance of the multiplier because the multiplier generally makes the system slower. Generally most multipliers are area consuming. Hence, optimizing the speed and area of the multiplier is a major design issue in all aspects. However, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas. As a result, a whole category of multipliers with different area-speed constraints has been designed with fully parallel. Some of the multipliers were designed in serial also. In between are digit serial multipliers where single digits consisting of several bits are operated on. While comparing with the performance criteria such as speed and area these multipliers will perform moderately. Multiplication is achieved by adding a list of shifted multiplicands according to the digits of the multiplier.

DSP blocks are the key component in various devices.DSP cores will implement image and video processing algorithms. Its output will be either image or video. So that we can done some approximations to improve the speed. Approximations can be done in the abstraction level. Approximation can be done by different techniques such as timing violation and function approximation method. In function approximation method here applying the approximation in adders, multipliers etc.

In this paper mainly focusing an energy efficient high speed multiplier. This approximate multiplier in short called as ROBA multiplier. This multiplier is applicable for both signed and unsigned numbers.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Issue 4, April 2018

1) This multiplier will modify the existing multipliers

2) Application in image processing part done with unsigned number multiplication

Section II includes the related works about existing multipliers, Section III includes proposed scheme and hardware implementation. Section IV includes implementation of multiplier. Section V includes analysis and discussions. Section VI includes applications and finally section VII includes conclusion part.

## II. Earlier Works Related

In this paper [1], an Advanced Array Multiplier using different types of compressors was designed and implemented on FPGA. In this method multiplier was divided into three stages a partial product generation stage, a partial product addition stage and final addition stage. . Add and shift is the algorithm that followed here. But the main disadvantage of array multiplier is the worst-case delay of the multiplier proportional to the width of the multiplier. Also it has high power consumption. More digital gates are there in this multiplier so that it results in large chip area. Even though this array multiplier has these disadvantages the main advantage that appeared while comparing with serial is that it performs well than serial in terms of speed.

In the papers [4] [10] to overcome the limitation of array multiplier the speed of the multiplier is increased by the booth algorithm. Booth algorithm will reduce the number of partial products. Here, the multiplier considers two numbers of bits at a time for the multiplication process. The multiplication process for both signed and unsigned numbers can be done in this booth multiplier. The main disadvantage of the booth multiplier is the complexity of the circuit to generate a partial product bit in the Booth encoding.

In [7] [11] papers mentioned about the Vedic multiplier. When comparing to the array it is possible to multiply the larger numbers by using this method. For that here uses a special algorithm called vertical and cross wise algorithm. Here also the partial products will be formed and it got arranged according to this algorithm. Even though it have some good values disadvantage is that circuit complexity will be high so that more power will get consumed and also the speed will be less if the numbers that are dealing with is large.

Wallace tree [8, 12] is an efficient hardware implementation of a digital circuit that multiplies two integers. Wallace trees are irregular structure in that the informal description does not specify a systematic method for the compressor interconnections. Wallace tree is having less delay and the logic levels can be reduced. When the layout got introduced it is more complex one also it has irregular wiring.

In [16] it comprises of DRUM multiplier even though it had some advantages like by using DRUM it's possible to reduce the hardware section. It is not possible to done signed and unsigned operations.

Finally paper [15] proposing an efficient multiplier when comparing with the existing ones. It is an approximate multiplier so that we can round the values to the power of two only after that we are doing the calculations. This multiplier we can make it use in image processing applications. Thus we will get an energy efficient yet high speed multiplier.

## III. Proposed Approximate Multiplier

### A. *Multiplication Algorithm of ROBA Multiplier*

The main methodology in this multiplier is that the numbers that need to multiply are rounding to the power of two. For that consider numbers A and B, their rounded values are $A_r$ and $B_r$. Multiplication of A and B can be written as

$$A \times B = (A_r - A) \times (B_r - B) + A_r \times B + B_r \times A - A_r \times B_r \quad (1)$$

The terms $A_r \times B$, $B_r \times A$ and $A_r \times B_r$ can be implemented by just shifting operation. The hardware implementation of $(Ar - A) \times (B_r - B)$ is rather complex. Even though we are omitting this term from the whole expression multiplication, it will make only small variation in the final result. Hence we are omitting that part from the multiplication. So that the reduced expression can be written as

$$A \times B = Ar \times B + B_r \times A - A_r \times B_r \quad (2)$$

The numbers that need to multiply can be rounded up or down based on the numbers we are taking. Actually what we are doing is we are doing is we are rounding to the nearest exponent of two. If the number is in the form of $3 \times 2^{p-2}$ , it has two nearest values equal to $2^n$. In this case the distance between the number and the nearest exponents of two will be the same. In these cases we can take the larger value. It will help to reduce the hardware. We will get smaller logical expressions by rounding up in this case.

Exceptional case is in the case of number 3. In this case we are taking the rounded value as 2. Mainly we are applying this ROBA to the unsigned numbers. If a signed number is there we are adding the sign after the multiplication operation.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Issue 4, April 2018

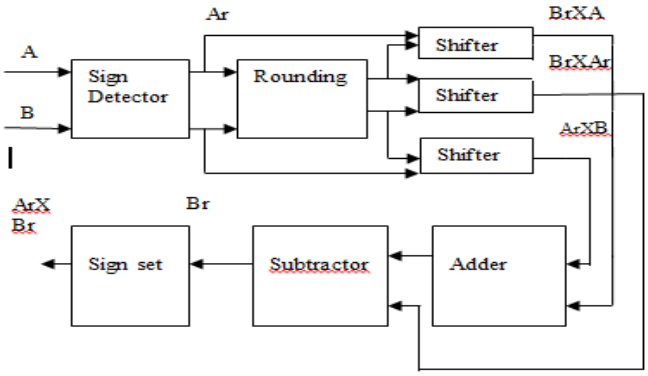


**Fig 1 Block Diagram of Multiplier**

*B. Hardware Implementation of Proposed Multiplier*

We are going to implement (2) with the fig 1.For that we are giving the inputs A and B[1]. First the signs of inputs are determined , and for each negative value, absolute value is generated. Next is the rounding block it will make the input numbers that we are giving to the nearest exponent of two. The rounding procedure is doing by the following equation

$$\vdots$$
$$A_r[i] = (\overline{A[i]} \cdot A[i-1] \cdot A[i-2] + A[i] \cdot \overline{A[i-1]}) \cdot \prod_{i=i+1}^{n-1} \overline{A[i]}$$
$$\vdots$$
$$A_r[3] = (\overline{A[3]} \cdot A[2] \cdot A[1] + A[3] \cdot \overline{A[2]}) \cdot \prod_{i=4}^{n-1} \overline{A[i]}$$
$$A_r[2] = A[2] \cdot \overline{A[1]} \cdot \prod_{i=3}^{n-1} \overline{A[i]}$$
$$A_r[1] = A[1] \cdot \prod_{i=2}^{n-1} \overline{A[i]}$$
$$A_r[0] = A[0] \cdot \prod_{i=1}^{n-1} \overline{A[i]}. \quad (3)$$

By using the above equation the numbers we can rounded. Here the value of n will be the number of bits of the input numbers that we are using. By determining the rounding values, using the three barrel shifters the terms $A_r$ x $B_r$, $A_r$ x B and $B_r$ x A can be calculated.

Kogge-Stone adder is being used here to calculate the summation part. After that the output of this adder and the result of Ar x $B_r$ are given as the input to the subtractor section. The output of this subtractor will be the absolute value of this multiplier.

The output of the subtractor part can be done by the following expression

Out = (P XOR Z) AND ({P<<1) XOR (P XOR Z)} or {(P AND Z) <<1})

Where P is $A_r$ x B + $B_r$ x A and Z is $A_r$ x $B_r$. The main benefit of using this expression is that we can make the implementation smaller and faster. Finally we are getting an output and if there is a sign change for output then we are adding it to the sign set.

## IV. IMPLEMENTATIONS

*A.Hardware Implementation of Proposed Multiplier*

Here the hardware implementation of the proposed multiplier is implementing using FPGA kit SPARTAN 3A. In that we can display the output in the 7 segment display in that kit. For that the inputs are giving to the switch there in the kit and the corresponding output will be generated after burning the program code to the kit.

SPARTAN 3A family of FPGA are used in high volume, cost effective electronic applications. This FPGA kit is programmed by loading configuration data into reprogrammable static CCL's it will control the functional elements and routing.

*B.Software Section*

1) Xilinx ISE

It is a software tool produced by Xilinx for synthesis and analysis of HDL designs. This proposed multiplier is synthesized using this tool and finally post routing and mapping will done. So that we can easily implement the multiplier section in FPGA kit. Here the purpose of Xilinx is that we are writing the program code there and its getting simulating. Thus we will get the correct results for the multiplication

2) Model Sim

It is a multi-language HDL simulation environment by Mentor graphics for simulation of HDL such as VHDL, Verilog, System C

After the synthesis in Xilinx tool the results can be simulated using Model Sim. And we can verify the multiplication results.

3) MATLAB

MATLAB (grid research centre) is a multi-worldview numerical figuring condition. A restrictive programming dialect created by Math Works, MATLAB permits network controls, plotting of capacities and information, usage of calculations, formation of UIs, and interfacing with programs written in different dialects, including C, C++, C#, Java, Fortran and Python

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Issue 4, April 2018

## V. RESULTS & DISCUSSIONS

A. *Comparative Study of Existing Multipliers*

Here the various multipliers were studied and compared the results.

**Booth Multiplier** — Device Utilization Summary

| Logic Utilization | Used | Available | Utilization |
|---|---|---|---|
| Number of 4 input LUTs | 49 | 1,408 | 3% |
| Number of occupied Slices | 25 | 704 | 3% |
| Number of Slices containing only related logic | 25 | 25 | 100% |
| Number of Slices containing unrelated logic | 0 | 25 | 0% |
| Total Number of 4 input LUTs | 49 | 1,408 | 3% |
| Number of bonded IOBs | 16 | 108 | 14% |
| Average Fanout of Non-Clock Nets | 4.00 | | |

**Array Multiplier** — Device Utilization Summary

| Logic Utilization | Used | Available | Utilization |
|---|---|---|---|
| Number of 4 input LUTs | 52 | 1,408 | 3% |
| Number of occupied Slices | 27 | 704 | 3% |
| Number of Slices containing only related logic | 27 | 27 | 100% |
| Number of Slices containing unrelated logic | 0 | 27 | 0% |
| Total Number of 4 input LUTs | 52 | 1,408 | 3% |
| Number of bonded IOBs | 16 | 108 | 14% |
| Average Fanout of Non-Clock Nets | 2.83 | | |

**Vedic Multiplier** — Device Utilization Summary

| Logic Utilization | Used | Available | Utilization |
|---|---|---|---|
| Number of 4 input LUTs | 29 | 1,408 | 2% |
| Number of occupied Slices | 15 | 704 | 2% |
| Number of Slices containing only related logic | 15 | 15 | 100% |
| Number of Slices containing unrelated logic | 0 | 15 | 0% |
| Total Number of 4 input LUTs | 29 | 1,408 | 2% |
| Number of bonded IOBs | 16 | 108 | 14% |

B.*Image Processing Applications*

To visualize the feasibility of the proposed multiplier in real applications, here compared the performances of ROBA in image processing application like sharpening. For sharpening each pixel of sharp image was extracted. Here the rounding approximated method got applied to the multiplier and finally it applied in an image to find out the sharpness value. By using this multiplier the speed can be improved because here the method is to rounding to the nearest exponent of two.

$$Y(i,j) = 2X(i+m,j+n) - \frac{1}{273}\sum_{m=-2}^{2}\sum_{n=-2}^{2} X(i+m,j+n).Masksharpening(m+3,n+3)$$

$$Mask_{sharpening} = \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

The method applying is that first of all taking an image and converting it to a text file and using xilinx software corresponding code is simulating and an output file will get generated.That output file can directly given to matlab.From that file output image can be seen.That output will be a sharpened imge.Also the input file will be visible.From that input and output image an analysis can be tabulated.Mainly MSE,PSNR,MSSIM.Also the shaprened range of input and output.

**Fig 2 Input and sharpened image**

The analysis of the above figure is as follows

Mean Square Error =7152.1505
Peak Signal to Noise Ratio=9.5864
Normalized Cross Relation=1.7913
Structural Similarity Index=.54192
Estimate Sharpness of Input Image=4.7339
Estimate Sharpness of Output Image=4.733

## VI. APPLICATIONS

Multiplier is an important factor in any kind of image processing applications. Some of the DSP based applications are audio signal processing, audio compression, digital synthesizers, biomedical field, speech recognition, speech processing etc. In all the above applications multiplier played an important role.

## VII. CONCLUSION

In this paper proposed a high speed multiplier which had high accuracy, and it is based on the rounding operation of the operands.ie., rounding to the nearest power of 2.By this way we can omit the complex part of multiplication and thus improving the speed and efficiency. Finally this multiplier is studied in the sharpening applications.

## ACKNOWLEDGMENT

I wish to record my indebtedness and thankfulness to all those who helped me to prepare this paper. First and foremost I thank God Almighty for his providence and for being the guiding light throughout the paper.

I want to express my thanks to my guide Ms Lekha N M, Asst Professor, Dept. of Electronics and Communication Engineering for her valuable advice and guidance towards

this work. I received motivation, encouragement and hold up from her during the course of work.

## REFERENCES

[1]. Zendegani, Reza, Mehdi Kamal, Milad Bahadori, Ali Afzali-Kusha, and Massoud Pedram(2017)"RoBA multiplier: A rounding-based approximate multiplier for high-speed yet energy-efficient digital signal processing." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems 25, no. 2 (2017): 393-401*

[2]. Laxman S, Darshan Prabhu R, Mahesh S Shetty , Mrs. Manjula BM, Dr. Chirag Sharma(2012) "FPGA Implementation of Different Multiplier Architectures" International Journal of Emerging Technology and Advanced Engineering Volume 2, Issue 6, June ISSN: 2250- 2459

[3]. S.P.Pohokar, R.S.Sisal, K.M.Gaikwad, M.M.Patil, Rushikesh Borse (2015) "Design and Implementation of 16x16 Multiplier using vedic mathematics" International Conference on Industrial Instrumentation and control(ICIC)College of Engineering Pune, India, May 28-30

[4]. V.Priyanka Brahmaiah ,L.Dhrma Teja, Dr Y.Padma sai (2013) "Study on comparison of various multipliers" International Journal of Electronics and communication engineering & technology(IJECET), ISSN 0976-6464 volume 4 Issue 5 September – October, pp-132-142.

[5]. Ruchi Sharma (2012) "Analysis of Different Multiplier with Digital Filters Using VHDL Language" International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-2, Issue-1, October

[6]. Mohammed Hasmat Ali, Anil Kumar Sahani (2013) " Study, Implementation and Comparison of Different Multipliers based on Array, KCM and Vedic Mathematics Using EDA Tools" International Journal of Scientific and Research Publications, Volume 3, Issue 6, June 2013 ISSN 2250-3153

[7]. S Venkateswara Reddy (2013)"Design and Implementation of 32 Bit multiplier using vedic mathematics" International journal of advanced research in electrical electronics and Instrumentation engineering Volume 2 , Issue 8 August 2013.

[8]. Vidhi Gupta, J. S. Ubhi "Analysis And Comparison Of Various Parameters For Different Multiplier Designs" International Journal Of Scientific Research And Management Studies (IJSRMS) ISSN: 2349-3771Volume 1 Issue 4, Pg: 136-147

[9]. Dipika Chauhan, Prof. Kinjal vagadia, Prof.Kirit patel (2015) " design and Implementation of High speed 64-bit multiply and Accumulator Unit using FPGA" International Journal of Innovative research in Computer and Communication Engineering Volume 3 Issue 5 May 2015.

[10]. Aditya Kumar Singh, Bishnu Prasad De, Santanu Maity (2012) "Design and Comparison of Multipliers Using Different Logic Styles" International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-2, May.

[11]. S.dhivya, Mr.T.Nallusamy (2015) "radix-8 modified booth recoder for high speed add-multiply operator" international conference on engineering trends and science & humanities issn: 2348 – 8549

[12]. S Muneer Ahamed 1, V Madhuri (2014)"Implementation Of High Speed 8-Bit Vedic Multiplier Using Barrel Shifter" SSRG International Journal of Electronics and Communication Engineering (SSRG-IJECE) – volume1 issue10 Dec ISSN: 2348 – 8549

[13]. Sulakshna Thakur, Pardeep Kumar (2015)"Area-Efficient & High Speed Ripple Carry based Vedic Multiplier" SSRG International Journal of Electronics and Communication Engineering (SSRG-IJECE) – EFES AprilISSN: 2348 - 8549 http://www.internationaljournalssrg.org Page 6

[14]. M.Rohini, M.Madhumetha, S.Ranjith (2015")An Survey on Adders & Multiplier For High Speed & Low Power Application" International Conference on Futuristic Trends in Computing and Communication (ICFTCC-2015) ISSN: 2348 – 8549 www.internationaljournalssrg.org Page 55

[15]. Rampur, Pramod V., M. Jagadish, and G. Yogeesha (2014). "Design and Implementation of Advanced Array Multiplier for Binary Multiplication on FPGA."

[16]. Hashemi, Soheil, R. Bahar, and Sherief Reda.2015) "Drum: A dynamic range unbiased multiplier for approximate applications." In Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, pp. 418-425. IEEE Press

## BIOGRAPHY

### First Author

4th semester M Tech ,Embedded Systems,Department of electronics and Communication,Sahrdaya College of Engineering & Technology,Thrissur.

### Second Author

Asst Professor, Department of electronics and Communication, Sahrdaya College of Engineering & Technology, Thrissur. Completed ME in Embedded Systems.20 years of industrial and 2 years of teaching experience