International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-4, February 2020

Implementation of 64 Bit Complex Floating-Point Multiplier on FPGA using Vedic Mathematics Sutra- Urdhva Tiryagbhyam

N. Janardan, T. Lakshman Sai Kumar, Velmathi Guruviah

Abstract— Multipliers play crucial role in present days in the area of digital signal processing and in communication systems applications. The entire system performance depends on speed area and power of the multipliers. In our paper, we developed a 64x64 bit complex floating-point multiplier with 64bit IEEE 754 format multipliers having less delay. Vedic multiplier of ripple carry adder based is suggested for mantissa multiplication in IEEE 754 format. Suggested Vedic multiplier uses historic Vedic Indian mathematics sutra called Urdhva-Tiryagbhyam for Vedic multiplication. The architecture Proposed for 64x64 bit complex floating-point multiplier is in Xilinx ISE 14.2 FPGA navigator in Verilog HDL. Eventually, the outcomes of the suggested multiplier will differentiate with traditional booth multiplier and array multiplier which represents clearly that complex multiplication using suggested architecture gives less delay, power and low area.

Index Terms— IEEE 754 format, Vedic mathematics, Urdhva Tiryagbhyam, field programmable gate array (FPGA), booth multiplier, array multiplier.

I. INTRODUCTION

Multipliers are extensively utilized in virtual signal processing and communication systems. The functioning of a processor relies on working fastness of multipliers. Multiplier takes maximum of implementing time of the processor. So, there is requirement to develop fastest functioning multipliers in systems. The complicated multiplication manner is important computation in radio communication technology. Radio communication channel terms are time dependent floating-point complex numbers. Because of, complicated multipliers in huge quantities are needed to develop the hardware modules. For that purpose, a Vedic multiplier which is quicker than the array multiplier had developed. Any have, the work has not take-up a complicated multiplication system.

Revised Manuscript Received on February 28, 2020.
***** Correspondence Author
N. Janardan*, M. Tech VLSI School of electronics Engineering, VIT Chennai, Tamil Nadu, India. Email: janardhannagalingam@gmail.com
T. Lakshman Sai Kumar, M. Tech VLSI School of electronics Engineering, VIT Chennai, Tamil Nadu, India. Email: tlakshmansaikumar@gmail.com
Velmathi Guruviah, professor, School of electronics Engineering, VIT Chennai, Tamil Nadu, India. Email: Velmathi.g@vit.ac.in

So that complicated Vedic multiplier with excessive speed is suggested and executed on FPGA. Anyhow, the methodology suggested is restricted to stable point multiplication.

So, in our paper, we developed 64bit complex floating multiplier with 64bit IEEE 754 format multipliers having less delay. A 64bit floating multiplier has 52bit for mantissa multiplication, 11bit for exponent and 1bit for sign representation. Eventually the outcomes of suggested multiplier are differentiated with traditional booth and array multipliers.

II. COMPLEX FLOATING-POINTMULTIPLIER

High speed Complex multipliers are mostly used in DSP systems as an example of DFT, FFT. To multiply 2 complex floating-point numbers, we have suggested an architecture of multiplier having 4 multipliers as shown in the figure 4. In that figure 4., We have 'a' and 'b' are the 2 complex number inputs. 'P' is the complex output. The imaginary and real parts of the input 'a' are a_{imag} and a_{real} and for the input 'b' are b_{imag} and b_{real} respectively. Similarly, p_{real} and p_{imag} are real and imaginary parts of the output multiplication of result 'p' respectively. In figure 2 the building block of 64bit floating point multiplier has shown.

$a = a_{real} + a_{imag}$ ---input 1
$b = b_{real} + b_{imag}$ ---input 2,
$p = p_{real} + p_{imag}$---output,

III. VEDIC MULTIPLICATION METHOD

The suggested multiplication technique for complex floating- point multiplier is historic Indian Vedic mathematic sutra called urdhva tiryagbhyam. urdhva tiryagbhyam is one of the finest historical Vedic mathematics sutras out of 16 sutras. The meaning of this technique is "vertically and cross wise".

3.1 2X2 BIT VEDIC MULTIPLIER

Implementation of 2bit Vedic multiplication block based on urdhva tiryagbhyam sutra by considering 4 AND gates for vertical addition of bits. 2 half adders for cross wise multiplication of bits are being used. 2bit Vedic multiplier is used as basic building block to construct 4bit Vedic multiplier and then 8bit VM then goes on up to 64bit VM.

That 64bit VM is used for mantissa multiplication in IEEE 754format by padding zeros for extra bits of mantissa which is shown in figure 3.

The following diagram gives clear explanation of Vedic multiplication technique of urdhva tiryagbhyam for 2,3,4bit multiplications.

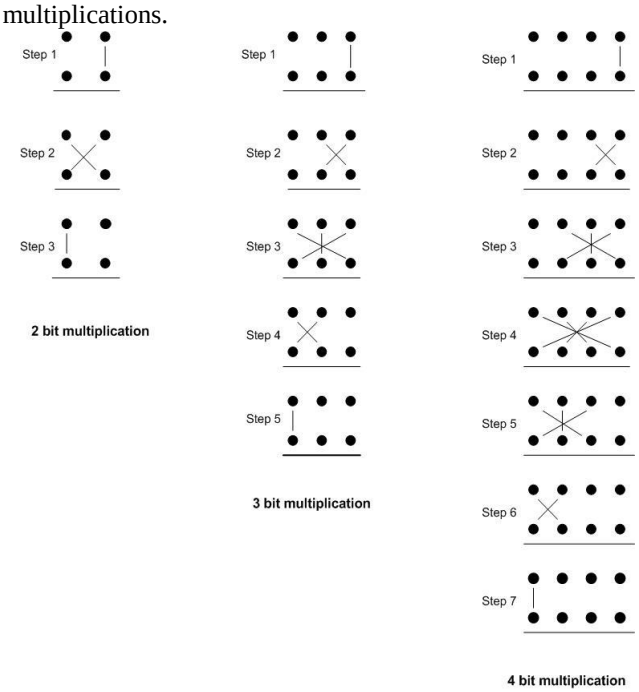


Figure 1. Vedic multiplication of 2,3,4bit using urdhva tiryagbhyam sutra

IV. BLOCK DIAGRAM

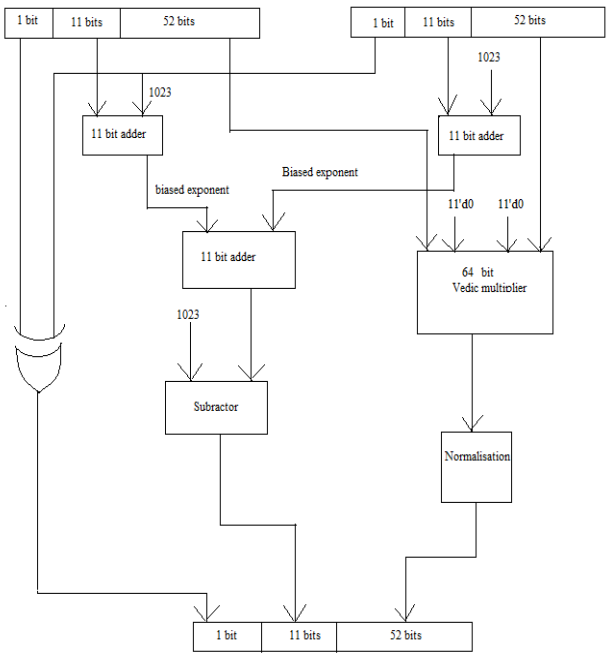


Figure 2. IEEE 754 format of 64bit floating point multiplier

4.1 Biased Exponent And Normalisation

In the above IEEE 754 format standard block diagram, two numbers are taken for multiplication with mantissa, exponent and sign bits accordingly to the required bits of format. But before giving two numbers for multiplication we have to add the number of bits that have taken left shift to get the 1point normalization of the considered numbers 'a' and 'b' to achieve the biased exponent and then take the bits after 1 in the 1point normalization for the mantissa. If that mantissa is having less than 52 bits then pad the remaining bits to zero. By doing biasing and normalization only we can further proceed to the multiplication in IEEE 64bit standard floating-point multiplier. This is called as biased exponent and normalization.

EXAMPLE:

85.125
85= 1010101
0.125= 001
85.125= 1010101.001
= 1.010101001*2^6
Sign bit = 0

For Double precision
The result after exponent biasing = 1023+6=1029
1029= 10000000101
Mantissa= 010101001
We will add zeros to complete 52 bits

The double precision format is:
0→sign bit
10000000101→exponent
0101010010000000000000000000000000000000000000000000→ Mantissa

V. PROPOSED METHODOLOGY

In the above section we, have discussed about the biased exponent and normalization in IEEE 754 standard format. In this section we discuss about the 53bit Vedic multiplier in the format. That 53bit Vedic multiplier is implemented by 64bit Vedic multiplier by padding remaining bits with zeros. For implementing this Vedic multiplier we followed the technique of urdhva tiryagbhyam of Vedic mathematics. In this 64bit Vedic multiplier we have taken ripple carry adder to add the resulted product terms of the multiplication.

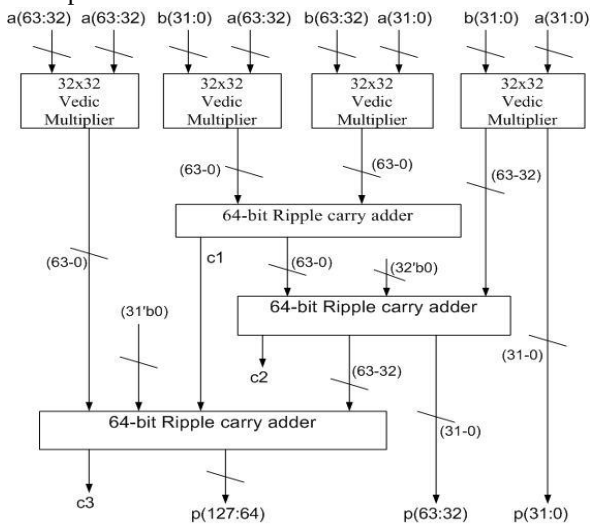


Figure 3. 64bit Vedic multiplier

5.1 Proposed Architecture

we have suggested that the 64bit complex floating - point multiplier can be implemented by taking four 64bit floating real multipliers in a row with their inputs as a_{real} a_{imag} , b_{real} b_{imag} as their inputs and the outputs of these four 64bit real floating multipliers are given to the floating point adder and floating point subtractor to get the final outputs as p_{real} and p_{imag} respectively.

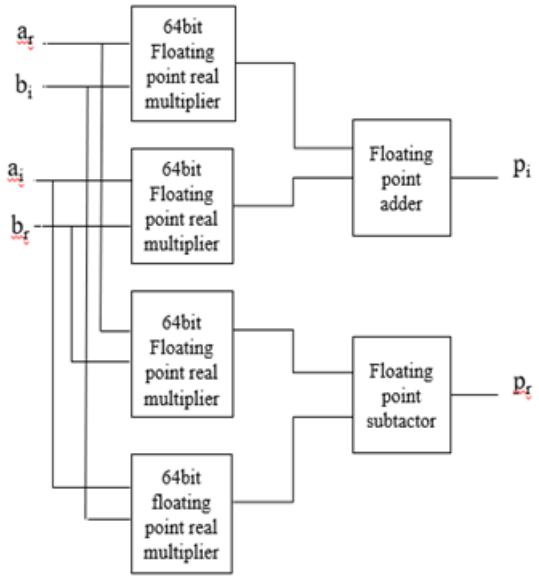
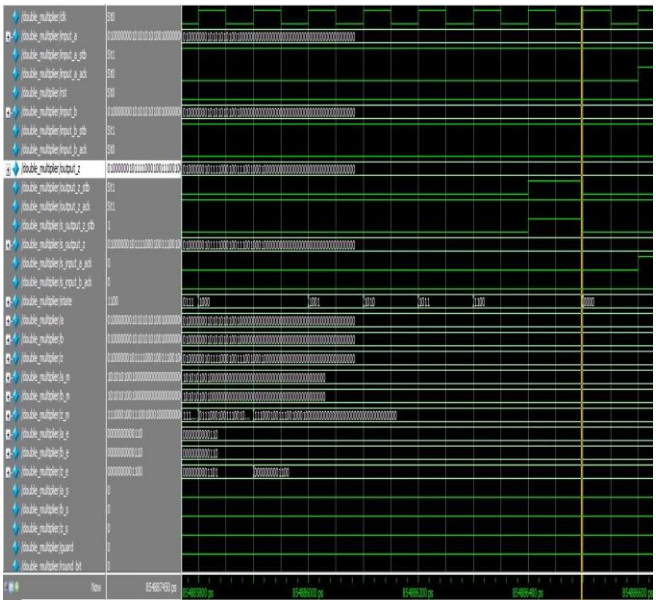


Figure 4. proposed architecture of 64bit complex floating multiplier

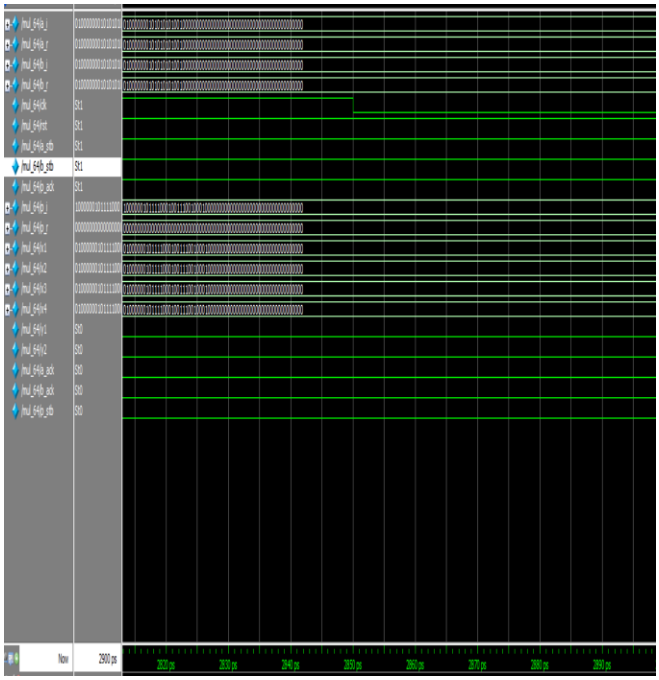
VI. SIMULATION RESULTS

After biasing the exponent with normalization of considered two numbers for the multiplication, we have to give them as inputs to the simulator. Then run the simulation, we will get the original result of multiplication also in biased form. The following are the simulated results of both 64bit real floating multiplier and 64bit complex floating multiplier.

6.1 Result of 64bit Real Floating Multiplier



6.2. Result of 64bit Complex Floating Multiplier



Simulation result of suggested multiplier solution
((85.125+j85.125) x (85.125+j85.125) = (0+j7246.26))

VII. COMPARISION BETWEEN SUGGESTED AND EXISTING MULTIPLIERS

Eventually, the outcomes we get from the simulation of suggested multiplication has compared with traditional booth multiplier and array multiplier.

Table-I: comparison between suggested and existing multipliers

Logic Utilisation	Booth Multiplier	Array Multiplier	Vedic Multiplier
No of LUTs	9247	6559	10877
No of IOs	331	379	254
Time delay	82.939ns	98.217ns	30.381ns
Power	120.29mw	132.23mw	91.30mw

VIII. CONCLUSION

Using four 64bit real floating-point multipliers, we have developed a 64bit complex floating multiplier in this paper. Outcomes of suggested multiplier were differentiated with traditional booth and array multipliers. So that, we can conclude the suggested methodology has less delay, low power in contrast with previous multiplication techniques. But we can observe clearly that to get reduced power and delay in the suggested methodology we went to trade off for the area. It means that area got increased than older multiplication techniques.

ACKNOWLEDGEMENTS

The assistance given by VIT university was highly admitted by writers where the consequences were followed up.

REFERENCES

1. H.D. Tiwari, G Gankhuya, C M Kim, and Y.B Cho, "Multiplier design-based on ancient Indian Vedic mathematics, "Proc. IEEE ISOCC'08, Vol.2, pp.65-68, 2008.
2. P. Saha, A. Banerjee, P. Bhattacharyya, and A. Dandapat, "High speed ASIC design of complex multiplier using Vedic mathematics," Proc. IEEE Students' Technology Symposium (TechSym), pp.237-241, Jan 2011.
3. K. Deergha Rao, Ch. Gangadhar and P. K. Korrai, "FPGA implementation of complex multiplier using minimum delay Vedic real multiplier architecture," Proc. IEEE UPCON 2016
4. Anjana R, Abishna B, Harshitha MS, Abhishek E, Ravichandra V, Suma MS (2014) Implementation of Vedic multiplier using Kogge stone adder. In: IEEE international conference on embedded systems (ICES), pp 28–31
5. Devpura P, Paliwal A (2016) High throughput Vedic multiplier using binary to excess-1 code converter. Int J Adv Res Electron Communication Engineering.
6. Gupta T, Sharma JB (2017) A CSA based architecture of Vedic multiplier for complex multiplication. Springer Ambient Communications and Computer Systems (RACCCS)
7. Han T, Carlson DA (1987) Fast area efficient VLSI adders. In: IEEE 8th symposium on computer arithmetic, pp 49–56
8. Jais A, Palsodkar P; SM-IEEE (2016) Design and implementation of 64bit multiplier using Vedic algorithm. In: IEEE international conference on communication and signal processing (ICCSP)

AUTHORS PROFILE

N. Janardan, student, is currently pursuing his M Tech VLSI, school of electronics, VIT Chennai. He has completed his B. Tech in ECE from Laki reddy Bali reddy college of Engineering. He already published a paper in IJR. His research areas are Digital IC Design, VLSI and signal processing.

T. Lakshman Sai Kumar, student, is currently pursuing his M Tech VLSI, school of electronics, VIT Chennai. He has completed his B. Tech in ECE from P V P Siddhartha institute of technology. His research areas are Digital IC Design, VLSI and Digital Design Using FPGA.

Dr. G. Velmathi, is the Professor in School of Electronics Engineering, at VIT Chennai. She received her UG from Thiagarajan college of engineering, PG from Madras Institute of Technology and obtained her doctoral degree from Indian Institute of Science (IISc) Bangalore. She has 25 years of teaching and research experience and also has 4 patents in her credit. She received several awards like best faculty award, best paper award etc. Dr. Velmathi is also the editorial board member of IEEE-Industrial Electronics Technology Transfer News. Her areas of research include VLSI, MEMS, Nano materials and Biomedical instrumentation applications.