

Design and Implementation of 64 bit High Speed Floating Point Multiplier for DSP Applications

¹Pooja Krishnamurthy Revankar ²Dr H C Hadimani ³Dr Udara Yedukondalu
Research Scholar, Department of E & C, G M Institute of Technology, Davangere, India
Professor, Department of E & C, G M Institute of Technology, Davangere, India
Professor, Department of E & C, SVEC (A), Tadepalligudem, India
pooja.k.revankar@gmail.com, hchadimani2017@gmail.com, drykudara@gmail.com

Article Info

Page Number: 767 – 773

Publication Issue:

Vol. 71 No. 3 (2022)

Abstract

Floating point number's multiplication is the most important process in the area of graph theory, multidimensional graphics, and digital signal processing, high performance computing etc. However, computers use binary numbers and it would like more precision however, it was found that binary numbers should be precise enough for most scientific and engineering calculations. So it was decided to double the amount of memory allocated. The Binary Floating point numbers are represented in Single and Double formats. The Single consist of 32 bits and the Double consist of 64 bits. The formats are composed of 3 fields; Sign, Exponent and Mantissa. The performance of Mantissa calculation Unit dominates overall performance of the Floating Point Multiplier. Many researchers have investigated the design of multiplier with different approaches. In this paper, we present the overview of work done by various researchers in their literature towards the design of Floating Point Multiplier. The creation of floating point units under a collection of area, latency and throughput constraint is an important ant consideration for system designers.

The code is written in Verilog and results shows that multiplier implemented using 64 bit floating point multiplication is efficient in terms of area, power and speed compared to its implementation using Array and Booth multiplier architectures.

Keywords: Floating point Multiplier, Genus, Innovus, cadence, GDSII file

Article History

Article Received: 12 January 2022

Revised: 25 February 2022

Accepted: 20 April 2022

Publication: 09 June 2022

1. Introduction

Floating-point numbers are widely adopted in many applications due their dynamic representation capabilities. Floating-point representation is able to retain its resolution and

accuracy compared to fixed-point representations. A large number of FP multiplications are carried out in various applications such as scientific calculation and computer graphics (CG). CG, in particular, requires enormous amount of FP multiplications to obtain high quality images required for multimedia systems. It is also of key importance to many modern applications such as 3D graphics accelerators, Digital Signal Processors (DSPs), High Performance Computing etc. These applications usually involve floating point calculations with double precision format. The growing computational demands of scientific applications shows that in many cases there is a need for increased precision in floating point calculations. Examples are the fields of computational physics, computational geometry, climate modeling etc., which require high precision calculations and great accuracy. Double precision binary floating-point is a commonly used format on PCs, due to its wider range over single precision floating point, even if at a performance and bandwidth cost. These applications usually require floating point calculations with double precision format, because this improves the accuracy of calculations and leads to more reliable results. For this reason, most Floating Point Units (FPUs) tend to provide support for executing double precision operations.

2. Literature survey

Y Sri Lakshmi [1] has built a 32-bit Vedic multiplier in which adder is Carry- select adder which is designed using a hybrid full adder. She compared Vedic multipliers that are built using a conventional full adder, compressor, and hybrid full adder in terms of Delay, Area, and Power. In this paper, she had simulated and verified this design using the DE2-115 FPGA kit.

Rohith S, Chandrashekara M N-[2] has designed an 8-bit Vedic multiplier with modified architecture which we have used in our design. They have done a comparative analysis of their design with Booth multiplier, Array multiplier, Wallace multiplier, and also they have obtained a delay of 14.219ns for their design using Spartan-6 device.

Shiksha Pandey-[3] presented 16x16 bit Vedic-multiplier architecture using Carry-Select adder. In this paper, a high speed of about 88ns is achieved. This is quite different from the Conventional method of multiplication like addition and shifting. She has shown simulations and design implementation and used software is XilinxISE9.2i.

M. Akila, C. Gowribala, S. Maflin Shaby-[4] has designed a 16-bit Vedic multiplier in which their focus is on modification of adder. In their design adder is designed using structures of Carry-Skip Adder and Carry Select Adder. They have obtained better speed which is about 10.730ns compared to that of other Vedic multiplier architectures.

Ms. Ayushi Sharma, Mr. Ajit Singh-[5] has designed a 32-bit Vedic multiplier using Brentkung adder and their design is synthesized and implemented on Spartan-6 XC6SLX4 FPGA board. They obtained a delay of 30.001ns.

V.Anil Kumar, S.Tamilselvan, CH.V.M.S.N.Pavan Kumar, V. Kamalkannan-[6] have designed a 16-bit Vedic multiplier and used this multiplier for the radix-2 FFT algo- rithm. They have done analysis on Carry-Skip Adder, Carry-Select Adder, Ripple-Carry Adder, and Carry Look-ahead Adder and found Carry-Select Adder gives less delay, so they have used this adder in their design of Vedic multiplier.

Shivaraj Kumar Patil, Poornima M, Shivakumar, Shridhar K P, Sanjay H-[7] presented a study on 8-bit Vedic multiplier architecture. Further, they have designed and implemented 2-,4-,8-bit Vedic multiplier. By using the Xilinx Synthesis tool, they have implemented the Verilog code for an 8-bit Vedic multiplier on the Spartan 3 kit and have obtained a delay of 28.27ns.

V. Charishma, G. Ganesh Kumar-[8] has implemented and designed a 32-bit Vedic multiplier and implemented on Spartan XC3S500-5-FG320 board and obtained a delay of about 31.526ns. They also used the Urdhva Tiryakbhyam sutra for their design.

3. Methodology

A. Methods for multiplication

Use of numerical methods is prevalent in most software algorithms. Computational physics, computational geometry, climate modeling etc., which require high precision calculations and great accuracy. Such applications demand an efficient code for basic mathematical operations i.e. multiplication. Real Time Systems demand instantaneous response to environmental variables and quick execution of taken decision. This motivated for an increased precision (64 bits) using 'time efficient' method for 'multiplication' (Vedic multiplication technique) to improve processor throughput. Proposed method for designing of a 64 bits double precision

Floating point multiplier of floating point numbers represented in IEEE 754 format is as follows. Initially, two operands will be checked to determine whether they contain a zero. If one of the operands is zero. The output results zero. If neither of them will zero, then the inputs with IEEE754 format will be unpacked and will be assigned to the check sign, add exponent and multiply mantissa. The product is positive when the two operands have the same sign; otherwise it is negative. Sign of the result is calculated by XORing sign bits of both the operands A and B Exponents of two multiplying numbers will be added to get the resultant exponent. Addition of exponent will be done using 16 bits adder. Exponents will be expressed in excess 1023 bit. The Mantissa Calculation Unit requires a 53 bit multiple.

This unit requires unsigned multiplier for multiplication of 53*53 BITS. The Vedic Multiplication technique is chosen for the implementation of this unit. This technique gives promising result in terms of speed and power. The Vedic multiplication system is based on 16 Vedic sutras, which describes natural ways of solving a whole range of mathematical problems. Out of these 16 Vedic Sutras the Urdhva triyakbhyam sutra or Nikhilam Sutra will be suitable for this purpose.

4. Simulation results

Above Fig. 4 shows the simulation result of High Speed and Low Power three operand adder (HS and LP 3 operand adder).

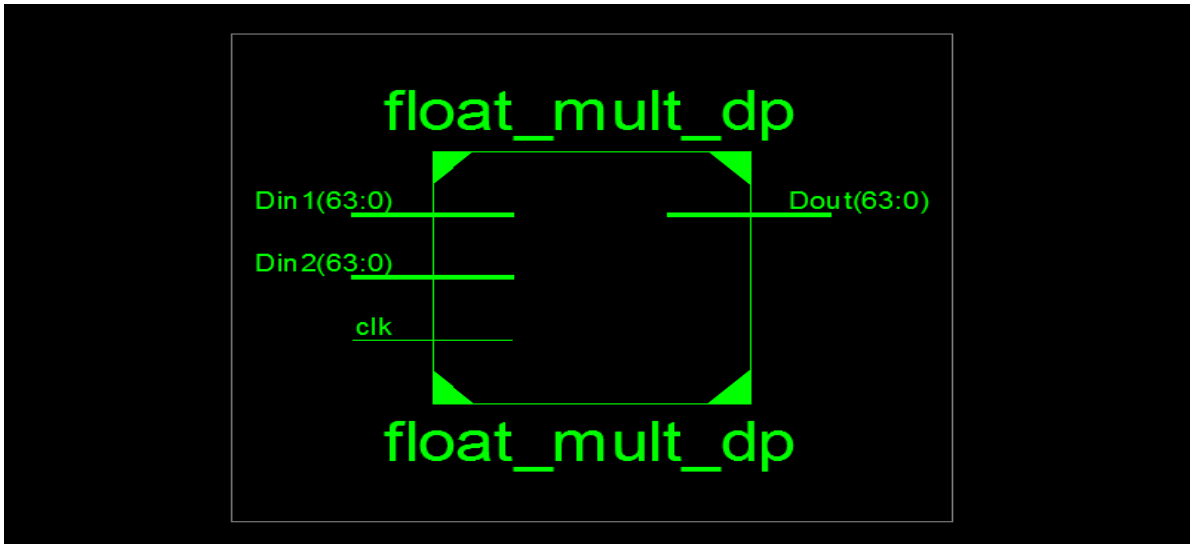


Figure 4: RTL View of 64-bit floating point Multiplier

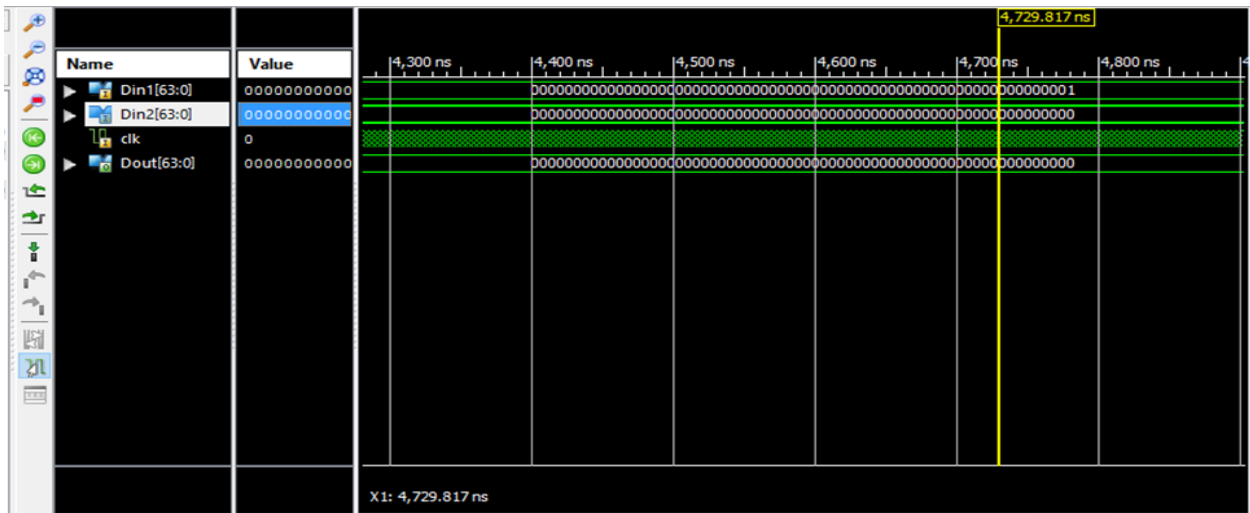


Figure 5: Simulation Result of 64-bit floating point Multiplier

4. Synthesis and Simulation Results

For the fair comparison, the same coding style using Verilog HDL using the Xilinx 14.7 ISE tool is adopted for designing the CS3A and HC3A and the proposed three-operand adders. Further, all these designs are synthesized using Synopsys Design Compiler in same SAED 32nm CMOS technology library to obtain the core area, timing and power for different word size. The physical synthesis analysis metrics comprised of maximum combinational gate delay, core area, power consumption, area-delay-product (ADP) and power-delay-product (PDP) are proved. The estimated results are shown shall vary with adopted verilog coding style and optimization options available in Genus tool.

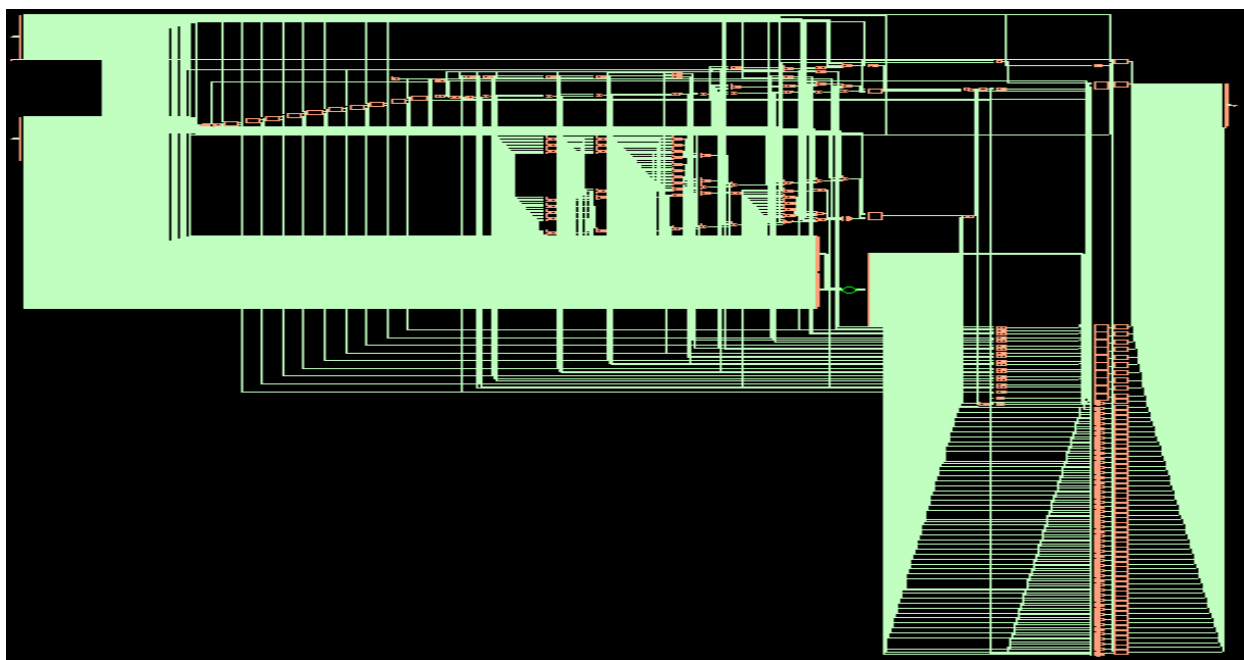


Figure 6: Synthesis Layout of floating point multiplier

```
legacy_genus:/> report power
```

Generated by: Genus(TM) Synthesis Solution 16.21-s018_1
 Generated on: May 02 2022 04:03:26 pm
 Module: float_mult_dp
 Technology library: fast_vddlv0 1.0
 Operating conditions: PVT_1P1V_0C (balanced_tree)
 Wireload mode: enclosed
 Area mode: timing library

Instance	Cells	Leakage Power(nW)	Dynamic Power(nW)	Total Power(nW)
float_mult_dp	4518	788.989	1383164.683	1383953.673
MULT_dp_7_6:mul_41_13	4273	750.659	1325223.371	1325974.030

Figure 7: detailed report of power

```
legacy_genus:/> report area
```

Generated by: Genus(TM) Synthesis Solution 16.21-s018_1
 Generated on: May 02 2022 04:03:47 pm
 Module: float_mult_dp
 Technology library: fast_vddlv0 1.0
 Operating conditions: PVT_1P1V_0C (balanced_tree)
 Wireload mode: enclosed
 Area mode: timing library

Instance	Module	Cells	Cell Area	Net Area	Total Area	Wireload
float_mult_dp		4518	12770	0	12770	<none> (D)
MULT_dp_7_6:mul_41_13	mult_unsigned	4273	12034	0	12034	<none> (D)

Figure 8: Detailed report of area

5. Physical layout

64 bit length input sequence is taken for the implementation of the described High Speed and Low Power floating point Multiplication and it is clear that described method acquires less power, less area and less delay which automatically increases the speed, which is suitable for

DSP applications. ADP and PDP plots are represented for 64 bit input sequence in Fig. 7 and Fig. 8 respectively. RTL View of 64-Bit Vedic multiplier is shown in below Fig. 9.

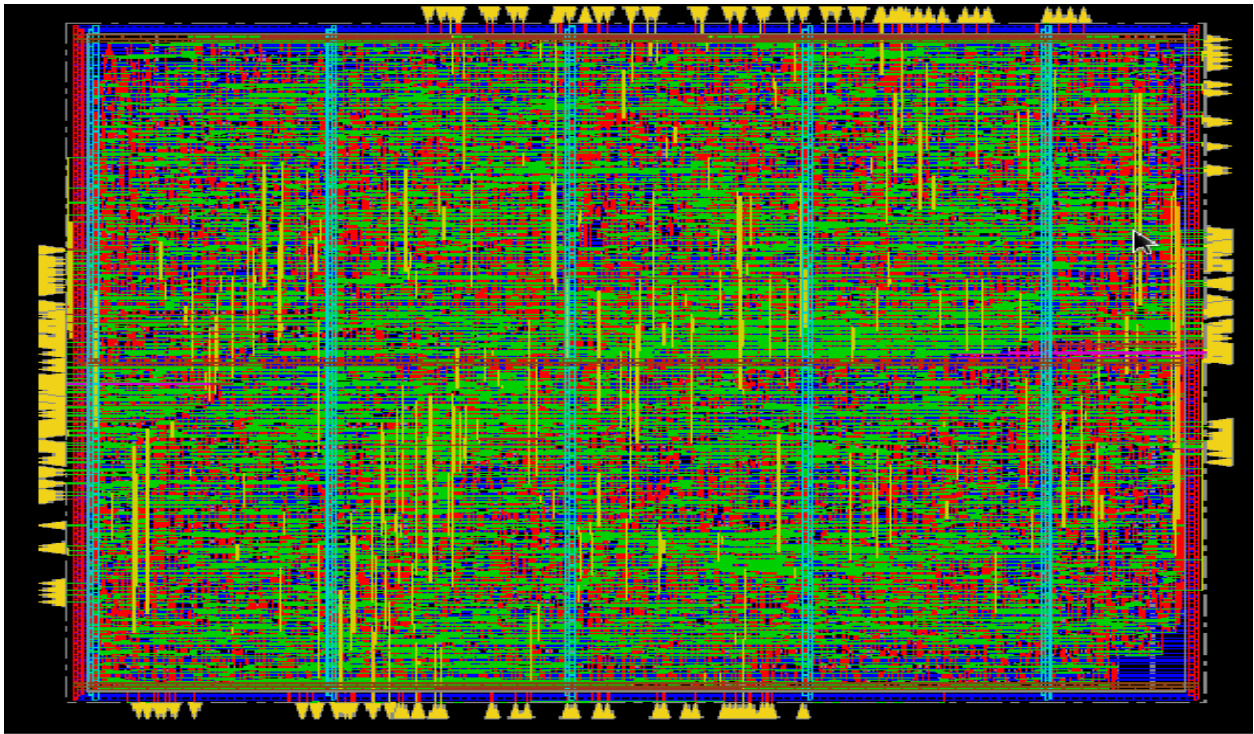


Figure 9: Physical Implementation of 64 bit floating point Multiplier

6. Conclusion

In this paper, 64 bit of floating point multiplier of High Speed and Low Power VLSI Architecture is implemented. The proposed Vedic technique is a parallel prefix adder that uses different stages structures to compute the multiplication of input operands. For the fair comparison, the same coding style using Verilog HDL using the Xilinx 14.7 ISE tool is adopted for designing the logic blocks and the proposed floating point Multiplication. The novelty of this proposed architecture is the reduction of power, delay and area in the prefix computation stages in PG logic and bit-addition logic that leads to an overall reduction in area-delay product (ADP) and power-delay product (PDP). From the physical synthesis results, this is clear that the proposed floating point multiplier architecture is 5 to 20 times faster than the corresponding booth architecture. Concluding that, our floating point Multiplication was comparatively better than other existing multiplication in terms of the power, area and delay.

References

- 1) S. S. Kerur, Prakash Narchi, Jayashree C N, Harish M Kittur, Girish V A, Implementation of Vedic Multiplier for Digital Signal Processing,” International Journal of Computer Applications (IJCA) 2011.
- 2) Al-Ashrafy, M.; Salem, A.; Anis, “An efficient implementation of floating point multiplier,” Electronics Communications and Pho-tonics Conference (SIECPC), 2011

- 3) Aniruddha Kanhe, Shishir Kumar Das, Ankit Kumar Singh, “De-sign and Implementation of Floating Point Multiplier based on Vedic Multiplication Technique” 2012 International Conference on Communication, Information & Computing Technology (ICCICT), Oct. 19-20, Mumbai, Indi.
- 4) Kavita Khare, R.P .Singh, Nilay Khare,”Comparison of pipelined IEEE-754 standard floating point multiplier with un pipelined multiplier” Journal of Scientific & Industrial Research Vol.65, pages 900-904 November 2006.
- 5) Manish Kumar Jaiswal, Nitin Chandrachood an “Efficient Implementation of IEEE Double Precision Floating-Point Multiplier on FPGA” 2008 IEEE Region 10 Colloquium and the Third ICIIS, Kharagpur, INDIA. December 8-10.
- 6) B. Lee and N. Burgess, “Parameter sable Floating-point Operations on FPGA,” Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems, and Computers, 2002.
- 7) Xilinx Floating-Point v2.0. [Online]. Available: <http://www.xilinx.com>
- 8) Gokul Govindu, L. Zhuo, S. Choi, V. Prasanna, “ Analysis of High performance Floating-point Arithmetic on FPGAs”, Proceedings of 18th International Parallel and Distributed Processing Symposium (IPDPS '04), pages 149-156, April-2004
- 9) P.V.Krishna Mohan Gupta, Ch.S.V.Maruthi Rao, G.R. Padmini, “An Efficient Implementation of High Speed Modified Booth En-coder for Floating Point Signed & Unsigned Numbers”. International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 8, August - 2013
- 10) G.Vaithiyanathan, K.Venkatesan, S.Sivaramakrishnan, S.Sivaand S. Jayakumar “Simulation And Implementation Of Vedic Multiplier Using Vhdl Code” International Journal of Scientific & Engineering Research Volume 4, Issue 1, January-2013 ISSN 2229-5518.
- 11) Himanshu Thapliyal, “Modified Montgomery Modular Multiplication using 4:2 Compressor and CSA Adder”, Proceedings of the third IEEE international workshop on electronic design, test and applications (DELTA 06), Jan 2005. IJSER International Journal of Scientific & Engineering Research, Volume 5, Issue 2, February-2014-238 ISSN 2229-5518 IJSER © 2014 <http://www.ijirest.org>
- 12) E.M.Saad, M.Taher, “High speed area efficient FPGA based floating point arithmetic modules”, National conference on radio science (NRSC 2007), March-2007, pp 1-8. doi:10.1109/DELTA.2008.19