Minimum Path Delay of SP and DP Floating Point Multiplier using Parallel Multiplier Technique

Sachindra Pathak and Prof. Manish Saxena
M. Tech. Scholar, Associate Professor
Dept. of Electronics and Communication Engineering
Bansal Institute Science and Technology, Bhopal

Abstract— **Due to advancement of new technology in the field of VLSI and Embedded system, there is an increasing demand of high speed and low power consumption processor. Speed of processor greatly depends on its multiplier as well as adder performance. In spite of complexity involved in floating point arithmetic, its implementation is increasing day by day. Due to which high speed adder architecture become important. Several adder architecture designs have been developed to increase the efficiency of the adder. In this paper, we introduce an architecture that performs high speed IEEE 754 floating point multiplier using carry select adder (CSA). Here we are introduced two carry select based design. These designs are implementation Xilinx Vertex device family.**

Keywords— ***IEEE754, Single Precision Floating Point (SP FP), Double Precision Floating Point (DP FP), Binary to Execess-1***

I. INTRODUCTION

The real numbers represented in binary format are known as floating point numbers. Based on IEEE-754 standard, floating point formats are classified into binary and decimal interchange formats. Floating point multipliers are very important in dsp applications. This paper focuses on double precision normalized binary interchange format. Figure 1 shows the IEEE-754 double precision binary format representation. Sign (s) is represented with one bit, exponent (e) and fraction (m or mantissa) are represented with eleven and fifty two bits respectively. For a number is said to be a normalized number, it must consist of'one' in the MSB of the significand and exponent is greater than zero and smaller than 1023. The real number is represented by equations (i) & (2).

$$Z = (-1^{s}) \times 2^{(E-Bias)} \times (1.M) \quad (1)$$

$$Value = (-1^{signbit}) \times 2^{(Exponent-1023)} \times (1.Mantissa) \quad (2)$$

Biasing makes the values of exponents within an unsigned range suitable for high speed comparison.

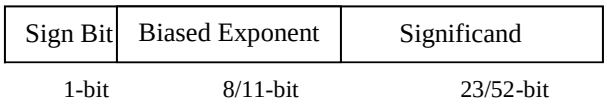

Figure 1: IEEE 754 Single Precision and Double Precision Floating Point Format

○ IEEE 754 STANDARD FLOATING POINT MULTIPICATION ALGORITHM

A brief overview of floating point multiplication has been explained below [5-6].

- Both sign bits S_1, S_2 are need to be Xoring together, then the result will be sign bit of the final product.
- Both the exponent bits E_1, E_2 are added together, then subtract bias value from it. So, we get exponent field of the final product.
- Significand bits Sig_1 and Sig_2 of both the operands are multiply including their hidden bits.
- Normalize the product found in step 3 and change the exponent accordingly. After normalization, the leading "1 "will become the hidden bit.

Above algorithm of multiplication algorithm is shown in Figure 2.

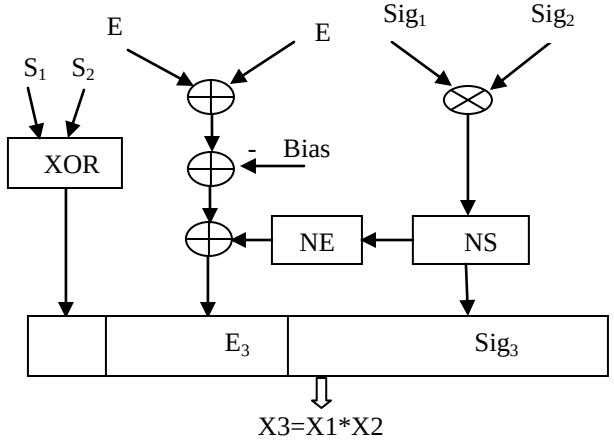

Figure 2: IEEE754 SP FP and DP FP Multiplier Structure, NE: Normalized exponent, NS: Normalized Significand

II. LITERATURE REVIEW

Implementation of Carry Select Adder based on Vedic Multiplier for Minimizing Path Delay and to Increase Efficiency processor Mr. M. Prasannakumar Dr. R. Thangavel 2021 [1], the Multiplier is a significant purposeful building block of all digital processors. Among all the arithmetic operations, the processor consumes most of its time and hardware resources in performing multiplication when compared to other operations like addition and subtraction. The high resource-consuming and iterative in nature are Multipliers, which minimizes the operational efficiency of the processing unit. As a result of this designing, a multiplier to minimize the combinational path delay and to increase the efficiency will be a challenging factor. In ancient maths, Vedic mathematics has an exclusive method of mental calculation with the support of basic rules and standards based on sutras. In designing the Multiplier the utilization of Vedic mathematics yield an extreme significance in the performance of processors. An efficient multiplier section can increase the overall productivity of the processor. In this paper, a new design is proposed using Carry Select Adder based Vedic Multiplier module to minimize the path delay.

Ms. Shilpa Shukla "VLSI Architecture for PPI-MO based Matrix Multiplication using Floating Point Multiplier" 2020 [2], due to advancement of new technology in the field of VLSI and Embedded system, there is an increasing demand of high speed and low power consumption processor. Speed of processor greatly depends on its multiplier as well as adder performance. Architecture that performs high speed IEEE 754 floating point multiplier using carry select adder (CSA). Here we are introduced two carry select based design. These designs are implementation Xilinx Vertex device family The major factors in the design or implementation of multipliers are chip area and speed of multiplication [1]. There is a high demand of high speed multiplications which requires less hardware. Various algorithms and the architectures have been proposed to design high-speed and low-power multipliers.

Design of High Speed, Low Power and Area Efficient 32-Bit Floating Point Multiplier " R Kalaimathi, R Senthil Ganesh 2017 [3], digital arithmetic operations are very important in the design of DSP and ASIC systems. Floating point computation has been widely used in graphics, digital signal processing, image processing and other applications. This complex application requires more time to process the data. Therefore, the high speed multiplication unit for floating point numbers is highly recommended to speed up multiplication process. The speed of floating point arithmetic unit is very crucial performance parameter which impinges the operation of the system. A fast and accurate operation of a digital system is greatly influenced by the performance of different adders and multipliers using different methodology. Emerging trends in low power made attention towards the search of low power, high speed and area efficient adders and multipliers architecture. In this project, single precision (32-bit) floating point multiplier is designed.

Table 1: Summary of Literature Review

Title, year of publication	Author Name	Methodology Used	Parameters
Implementation of Carry Select Adder based on Vedic Multiplier for Minimizing Path Delay and to Increase Efficiency processor 2021	Mr. M.Prasannakumar Dr. R. Thangavel	Carry Select Adder based on Vedic Multiplier	Delay for SPFPA = 63.01 ns, Delay for DPFPA = 83.169 ns
Architecture for PPI-MO based Matrix Multiplication using Floating Point Multiplier IJSPR 2020	Ms. Shilpa Shukla Prof. Amrita Khera	Architecture that performs high speed IEEE 754 floating point multiplier using carry select adder (CSA)	Delay for SPFPM =15.058 ns Delay in DPFPM = 20.071 ns
Design of High Speed, Low Power and Area Efficient 32-Bit Floating Point Multiplier 2017	Mr. R Kalaimathi, Mr. R Senthil Ganesh	Design a new floating point multiplier that occupies less area, low power dissipation and reduces computational time	Time Delay – 25.564 m

III. DIFFERENT TYPES OF ADDER

Parallel Adder:-

Parallel adder can add all bits in parallel manner i.e. simultaneously hence increased the addition speed. In this adder multiple full adders are used to add the two corresponding bits of two binary numbers and carry bit of the previous adder. It produces sum bits and carry bit for the next stage adder. In this adder multiple carry produced by multiple adders are rippled, i.e. carry bit produced from an adder works as one of the input for the adder in its succeeding stage. Hence sometimes it is also known as Ripple Carry Adder (RCA). Generalized diagram of parallel adder is shown in figure 3.

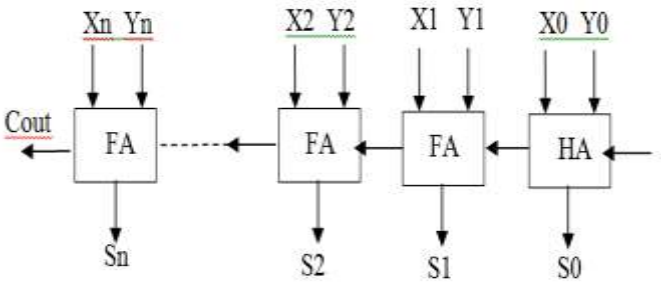

Figure 3: Parallel Adder (n=7 for SPFP and n=10 for DPFP)

An n-bit parallel adder has one half adder and n-1full adders if the last carry bit required. But in 754 multiplier's exponent adder, last carry out does not required so we can use XOR Gate instead of using the last full adder. It not only reduces the area occupied by the circuit but also reduces the delay involved in calculation. For SPFP and DPFP multiplier's exponent adder, here we Simulate 8 bit and 11 bit parallel adders respectively as show in figure 4.

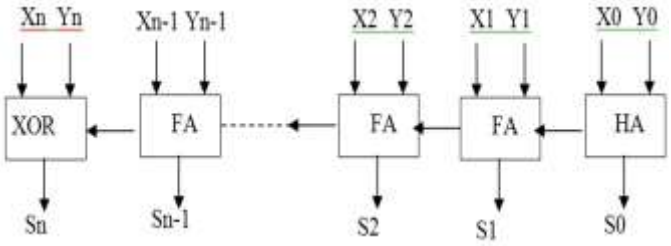

Figure 4: Modified Parallel Adder (n=7 for SPFP and n=10 for DPFP)

Carry Skip Adder:-

This adder gives the advantage of less delay over Ripple carry adder. It uses the logic of carry skip, i.e. any desired carry can skip any number of adder stages. Here carry skip logic circuitry uses two gates namely "and gate" and "or gate". Due to this fact that carry need not to ripple through each stage. It gives improved delay parameter. It is also known as Carry bypass adder. Generalized figure of Carry Skip Adder is shown in figure 5.

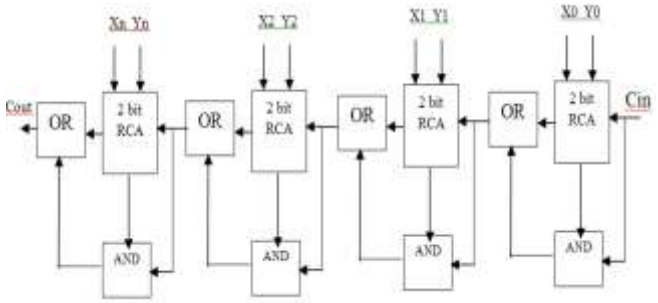

Figure 5: Carry Skip Adder

Carry Select Adder:-

Carry select adder uses multiplexer along with RCAs in which the carry is used as a select input to choose the correct output sum bits as well as carry bit. Due to this, it is called Carry select adder. In this adder two RCAs are used to calculate the sum bits simultaneously for the same bits assuming two different carry inputs i.e. '1' and '0'. It is the responsibility of multiplexer to choose correct output bits out of the two, once the correct carry input is known to it. Multiplexer delay is included in this adder. Generalized figure of Carry select adder is shown in figure 3.9. Adders are the basic building blocks of most of the ALUs (Arithmetic logic units) used in Digital signal processing and various other applications**.** Many types of adders are available in today's scenario and many more are developing day by day. Half adder and Full adder are the two basic types of adders. Almost all other adders are made with the different arrangements of these two basic adders only. Half adder is used to add two bits and produce sum and carry bits whereas full adder can add three bits simultaneously and produces sum and carry bits.

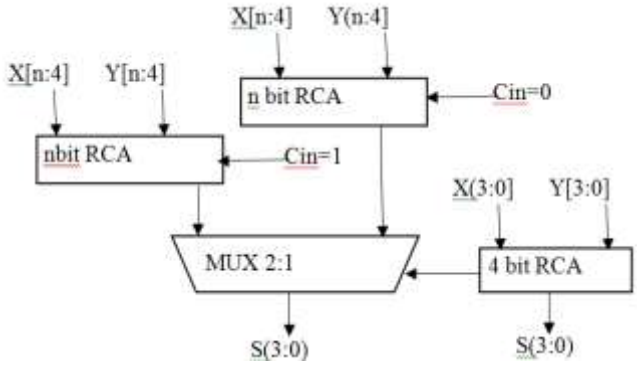

Figure 6: Carry Select Adder

IV. PROPOSED METHODOLOGY

In IEEE754 standard floating point representation, 8 bit Exponent field in single precision floating point (SP FP) representation and 11 bit in double precision floating point (DP FP) representation are need to add with another 8 bit exponent and 11 bit exponent respectively, in order to multiply floating point numbers represented in IEEE 754 standard as explained earlier. Ragini et al. [10] has used parallel adder for adding exponent bits in floating point multiplication algorithm. We proposed the use of 8-bit modified CSA with dual RCA and 8-bit modified CSA with RCA and BEC for adding the exponent bits. We have found the improved area of 8-bit modified Carry select adder with RCA and BEC over the 8-bit modified CSA with dual RCA.

- Sign bit calculation

To calculate the sign bit of the resultant product for SP FP and DP FP multiplier, the same strategy will work. We just need to XOR together the sign bits of both the operands. If the resultant bit is '1', then the final product will be a negative number. If the resultant bit is '0', then the final product will be a positive number.

- Exponent bit calculation

Add the exponent bits of both the operands together, and then the bias value (127 for SPFP and 1023 for DPFP) is subtracted from the result of addition. This result may not be the exponent bits of the final product. After the significand multiplication, normalization has to be done for it. According to the normalized value, exponents need to be adjusted. The adjusted exponent will be the exponent bits of the final product.

- Significand bit calculation

Significand bits including the one hidden bit are need to be multiply, but the problem is the length of the operands. Number of bits of the operand will become 24 bits in case of SP FP representation and it will be 53 bits in case of DP FP representation, which will result the 48 bits and 106 bits product value respectively. In this paper we use the technique of break up the operands into different groups then multiply them. We get many product terms, add them together carefully by shifting them according to which part of one operand is

multiplied by which part of the other operand. We have decomposed the significand bits of both the operands ain four groups. Multiply each group of one operand by each group of second operand. We get 16 product terms. Then we add all of them together very carefully by shifting the term to the left according to which groups of the operands are involved in the product term.

V. SIMULATION RESULT

All the designing and experiment regarding algorithm that we have mentioned in this paper is being developed on Xilinx 6.2i updated version. Xilinx 6.2i has couple of the striking features such as low memory requirement, fast debugging, and low cost. The latest release of ISE™ (Integrated Software Environment) design tool provides the low memory requirement approximate 27 percentage low. ISE 6.2i that provides advanced tools like smart compile technology with better usage of their computing hardware provides faster timing closure and higher quality of results for a better time to designing solution.

These designs were compared with IEEE-754 floating point multiplier architecture proposed by Soumya Havaldar et al. [2] to show for the improvements obtained.

So Soumya Havaldar et al. [1] architecture is best in all these architectures. Implementing the Soumya Havaldar et al. [1], proposed architecture IEEE-754 floating point design has been captured by VHDL and the functionality is verified by RTL and gate level simulation. To estimate the number of slice, number of 4-i/p LUTs and maximum combinational path delay (MCPD).

Table IV: Comparison Result

Parameter	Previous SPFP Algorithm	Implemented SPFP Multiplier using Partition Method	Previous DPFP Algorithm	Implemented DPFP Multiplier using Partition Method
Number of Slice LUTs	154	55	146	39
Number of Input Output Bonded	74	38	96	48
Maximum Combinational Path Delay	15.058	9.128	20.148	11.543

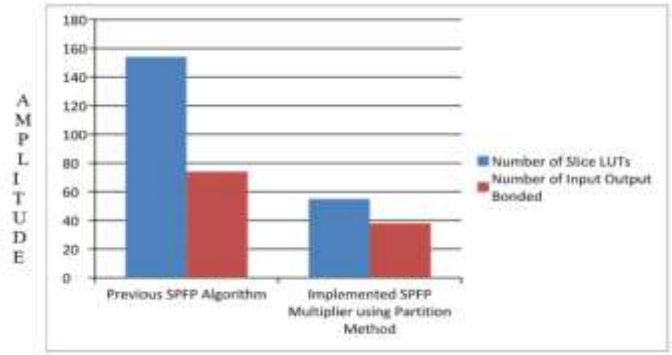

Figure 7: Comparison Result for Previous and Implemented SPFP Algorithm for different Device Family

VI. CONCLUSION

IEEE754 standardize two basic formats for representing floating point numbers namely, single precision floating point and double precision floating point. Floating point arithmetic has vast applications in many areas like robotics and DSP. Delay provided and area required by hardware are the two key factors which are need to be consider Here we present single precision floating point multiplier by using two different adders namely modified CSA with dual RCA and modified CSA with RCA and BEC.

Among all two adders, modified CSA with RCA and BEC is the least amount of Maximum combinational path delay (MCDP). Also, it takes least number of slices i.e. occupy least area among all two adders.

REFRENCES

[1] Implementation of Carry Select Adder based on Vedic Multiplier for Minimizing Path Delay and to Increase Efficiency processor Mr. M. Prasannakumar Dr. R. Thangavel 2021.

[2] Ms. Shilpa Shukla “VLSI Architecture for PPI-MO based Matrix Multiplication using Floating Point Multiplier” 2020.

[3] Design of High Speed, Low Power and Area Efficient 32-Bit Floating Point Multiplier “ R Kalaimathi, R Senthil Ganesh 2017.

[4] Soumya Havaldar, K S Gurumurthy, “Design of Vedic IEEE 754 Floating Point Multiplier”, IEEE International Conference On Recent Trends In Electronics Information Communication Technology, May 20-21, 2016, India.

[5] Ragini Parte and Jitendra Jain, “Analysis of Effects of using Exponent Adders in IEEE- 754 Multiplier by VHDL”, 2015 International Conference on Circuit, Power and Computing Technologies [ICCPCT] 978-1-4799-7074-2/15/$31.00 ©2015 IEEE.

[6] Ross Thompson and James E. Stine, “An IEEE 754 Double-Precision Floating-Point Multiplier for Denormalized and Normalized Floating-Point Numbers”, International conference on IEEE 2015.

[7] M. K. Jaiswal and R. C. C. Cheung, “High Performance FPGA Implementation of Double Precision Floating Point Adder/Subtractor”, in International Journal of Hybrid Information Technology, vol. 4, no. 4, (2011) October.

[8] B. Fagin and C. Renard, "Field Programmable Gate Arrays and Floating Point Arithmetic," IEEE Transactions on VLS1, vol. 2, no. 3, pp. 365-367, 1994.

[9] N. Shirazi, A. Walters, and P. Athanas, "Quantitative Analysis of Floating Point Arithmetic on FPGA Based Custom Computing Machines," Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines (FCCM"95), pp.155-162, 1995.

[10] Malik and S. -B. Ko, “A Study on the Floating-Point Adder in FPGAs”, in Canadian Conference on Electrical and Computer Engineering (CCECE-06), (2006) May, pp. 86–89.

[11] D. Sangwan and M. K. Yadav, “Design and Implementation of Adder/Subtractor and Multiplication Units for Floating-Point Arithmetic”, in International Journal of Electronics Engineering, (2010), pp. 197-203.
[12] L. Louca, T. A. Cook and W. H. Johnson, “Implementation of IEEE Single Precision Floating Point Addition and Multiplication on FPGAs”, Proceedings of 83rd IEEE Symposium on FPGAs for Custom Computing Machines (FCCM‟96), (1996), pp. 107–116.
[13] Jaenicke and W. Luk, "Parameterized Floating-Point Arithmetic on FPGAs", Proc. of IEEE ICASSP, vol. 2, (2001), pp. 897-900.
[14] Lee and N. Burgess, “Parameterisable Floating-point Operations on FPGA”, Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems, and Computers, (2002).
[15] M. Al-Ashrafy, A. Salem, W. Anis, “An Efficient Implementation of Floating Point Multiplier”, Saudi International Electronics, Communications and Photonics Conference (SIECPC), (2011) April 24-26, pp. 1-5.