ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology
An ISO 3297: 2007 Certified Organization Volume 9, Special Issue 1, March 2020
8th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '20)
18th-19th March 2020
Organized by
Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India

Design and Implementation of Single Precision FPM in FPU for Optimized Speed

Bhuvanapriya.R [1], Dr. Menakadevi.T. [2]

P.G. Student, Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India[1]

Associate Professor, Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India[2]

ABSTRACT: Currently, each CPU has one or additional Floating Point Units (FPUs) integrated inside it. It is usually utilized in math wide-ranging applications, such as digital signal processing. It is found in places be established in engineering, medical and military fields in adding along to in different fields requiring audio, image or video handling. A high-speed and energy-efficient floating point unit is naturally needed in the electronics diligence as an arithmetic unit in microprocessors. Most of the operations accounting 95% of conformist FPU are multiplication and addition. Many applications need the speedy execution of arithmetic operations. In the existing system, the FPM(Floating Point Multiplication) have more delay and fewer speed and fewer throughput. The demand for high speed and throughput intended to design the multiplier and adder blocks within the FPM (Floating point multiplication) in a format of 32 bit single precision floating point. This is designed with Verilog code using Xilinx ISE 14.5i software tool is employed to code and verify the ensuing waveforms of the designed code.

KEYWORDS: IEEE 754, FPU, FPM, Xilinx ISE.

I. INTRODUCTION

Floating point units (FPUs) are a mathematical coprocessor that's just for floating point numbers. It will handle arithmetic operations. The most newest processors through software library routines can also accomplish numerous transcendental functions like exponential or trigonometric calculation. In computer the real numbers can be expressed in numerous ways. Floating-point representation mainly follows the standard IEEE-754 format, which is the foremost common way of representing an approximate to real numbers in computers as it's well handled in most bulky processors. The real numbers represented in binary format are called floating point numbers. These are portrayed computers by a standards Committee which was formed in 1985 by IEEE [8] .The IEEE-754 floating point illustration for binary real numbers contains three parts is shown in figure 1.

SIGN	EXPONENT	MANTISSA

Fig.1: Floating point representation

A. *FLOATING POINT REPRESENTATION*

The floating point are often represented in four main sizes such as half (16 bits) ,single(32 bits),double(64 bits) and quadruple(128 bits) precision [9]. The IEEE single precision floating point format has a sign bit, 8 bits of exponent, 24 bits of mantissa and occupies 32 bit. The value of bias is 127, an exponent 0 implies that -127 is kept within the exponent field. The exponents -127 (all 0s) and +128 (all 1s) .The mantissa/significand is 24 bits long -23 bits of the mantissa that is stored in the memory and an implied '1' as the most significant '24^{th}' bit .Thus, a (32 bit) single precision floating point number representation in the IEEE standard is given by equations(1)&(2).

$$X=(-1^{S}) \times 2^{(E-Bias)} \times (1.M) \qquad (1)$$

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology
An ISO 3297: 2007 Certified Organization Volume 9, Special Issue 1, March 2020
8th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '20)
18th-19th March 2020
Organized by
Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India

$$\text{Value} = (-1^{\text{Sign bit}}) \times 2^{(\text{Exponent}-127)} \times (1.\text{Mantissa}) \quad (2)$$

PRECISION	SIGN	EXPONENT	Emax	Emin	MANTISSA	BIAS
Single Precision	1[31]	8[30-23]	+127	-126	23[22-0]	127

TABLE I: BIT RANGE

PRECISION	MAX	MIN
Single Precision	$3.40*10^{38}$	$1.8*10^{-38}$

TABLE II :IEEE 754 STANDARD RANGES

The bit range of single precision floating point is shown as table in Table I .The IEEE 754 Standard range of maximum and minimum values that the single (32 bit) precision that can accommodate is shown in Table II. If the values exceeds the maximum value then its overflow case and when the values is below the minimum range then its underflow case .

In FPM internal adders and multipliers constitute an essential element as it performs 95% of operations with extensive research is focused on improving its performance and delays. Amongst every arithmetic operation once the implementation of addition completed then it is easy to achieve multiplication.

This paper is ordered like: Section II describes work related to the implementation ,the system architecture is given in Section III, Section IV presents single precision floating point, Section V presents 32 bit FPM and followed by section VI presents implementation of 32 bit FPM and section VII presents simulation results and section VIII presents conclusion and section IX presents future scope .

II. RELATED WORK

Several researchers have worked on implementing floating point arithmetic in their own way. The idea of floating point numbers given and three fields includes sign, fraction and an exponent field and extremely small to huge numbers with a varying level of precision [1]. The design of an IEEE single precision floating point addition and multiplication unit format given by IEEE std.754-1985 is characterized [2]. The IEEE-754 Standard defining numerous floating point number format and the sizes of the fields [3]. The well-known Carry-look ahead Adder (CLA) technique has been used to implement the phase accumulator in direct digital synthesizer [4]. This paper states with comparison of other adders CLA is better [7].

III. ARCHITECTURE OF THE SYSTEM

The floating point unit is an IC which is intended to govern all the arithmetic operations based on floating point numbers or fractions. Figure 2, shows the architecture of proposed FPU, this unit shows that there are pre-normaliser block for add and multiplication units and post normaliser block for both add and multiplication and the later exceptions unit is present to handle the expectations unit .The fpu_op helps in choosing the operation based on the inputs the operation is performed and the op_a and op_a are the operands and these are the floating point numbers will perform the appropriate functions based on the fpu_op operation . The fundamental microprocessor isn't capable of manipulating floating number quickly so a separate specialized floating point unit is designed as co-processor. This unit is meant to perform basically five operations like addition, subtraction, multiplication , division and square root. The proposed FPU corresponds to two major units in the unit which is represented in the top view of FPU is shown in Figure 3.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology
An ISO 3297: 2007 Certified Organization Volume 9, Special Issue 1, March 2020
8th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '20)
18th-19th March 2020
Organized by
Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India

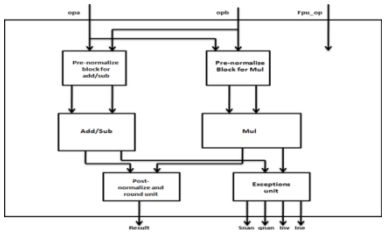

Fig 2: Architecture of proposed FPU

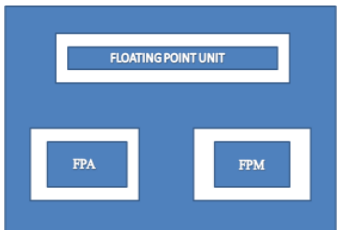

Fig 3: Top view of proposed FPU

IV. SINGLE PRECISION FLOATING POINT

The IEEE single precision floating point format has a sign bit is represented as "1/0" bit. "1" in sign bit indicates the negative and "0" indicates the positive. The total number of exponent is 255 here the ranges depends on the exponent value in accordance 8 bits of exponent is accommodated and it has a bias of 127 .There are totally 24 bits of mantissa but last 1 bit is hidden so it's always represented as 23 bit of mantissa as per IEEE-754 standard and occupies 32 bit.

V. FLOATING POINT MULTIPLICATION

Floating point multiplication (FPM) is represented by single precision format involve in estimation of sign bit , exponent and mantissa of the product and later normalizing and round off the resultant values to finally attain the result .The two operands are intended for XOR operation for the sign check . The exponents A and B are added to attain the resultant exponent is subtracted from 127 to achieve its biased exponent. The mantissa are multiplied based on the resultant results obtained .Finally , all are formed into the format declared by IEEE 754.The architecture of FPM is shown in figure 4.

ALGORITHM: FLOATING POINT MULTIPLICATION

ALGORITHM REPRESENTATION : $X3 = (-1)^{s1}(M1 \times 2^{E1}) * (-1)^{s2}(M2 \times 2^{E2})$

STEPS INVLOVED :

- Step 1. Obtain the sign; i.e. s1 XOR s2
- Step 2. Multiply the mantissa; i.e. (1.M1*1.M2)
- Step 3. Place the decimal point in the result
- Step 4. Add the exponents; i.e. (E1 + E2 – Bias)
- Step 5. Normalize the result; i.e. obtaining 1 at the MSB of the results mantissa
- Step 6. Round the result to fit in the available bits.
- Result X3 = X1 * X2

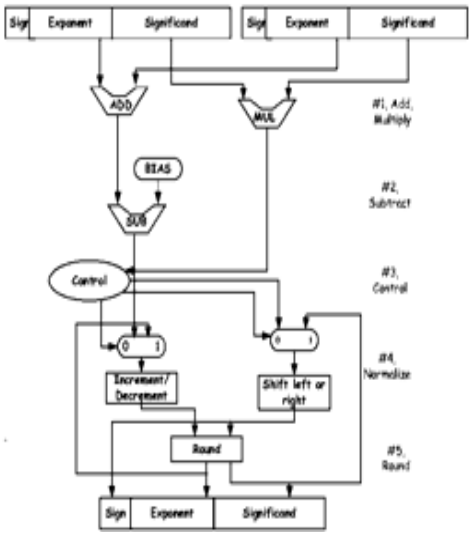

Fig 4 :Floating Point Multiplication Architecture

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology
An ISO 3297: 2007 Certified Organization Volume 9, Special Issue 1, March 2020
8th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '20)
18th-19th March 2020
Organized by
Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India

VI. IMPLEMENTATION OF FLOATING POINT MULTIPLICATION

The proposed floating point multiplication block is shown figure 5, the block diagram of multiplier unit comprises of a XOR operation, normalizer block and bias operation. Consider two operands A and B , the XOR involves in SIGN CALCULATION : if its positive "0" or "1".

EXPONENT CALCULATION : In this design exponent addition is calculated with two 8 bit exponent and this block is proposed with 8 bit carry look ahead adder (CLA) as shown in figure 6 , it enhances the speed by dropping the time essential to corroborate carry bits[10]. It computes one or added carry bits former to the sum, the huge-value bits of the adder to conclude the outcome has less wait time The end result is a condensed carry propagation time. It subsets of propagate, sum and carry generator. It adds the exponents of the two operands and then it bias 127 is subtracted from results obtained from the resultant result of exponent i.e.EA+EB-bias .

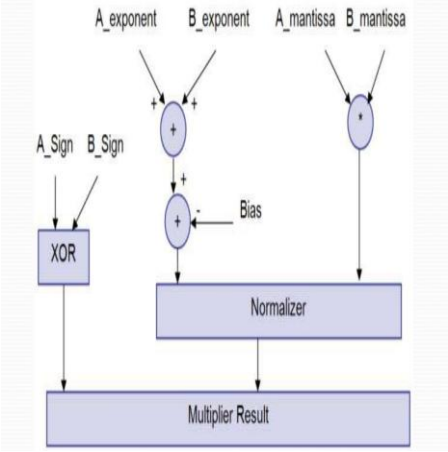

Fig 5: Block Diagram Of Multiplication Unit

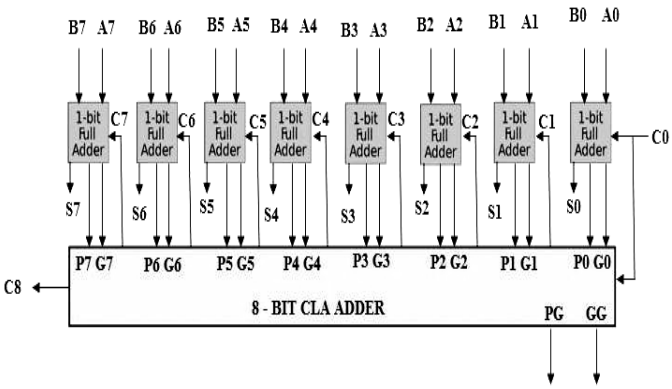

Fig 6 : 8 Bit Carry Look Ahead For Exponent

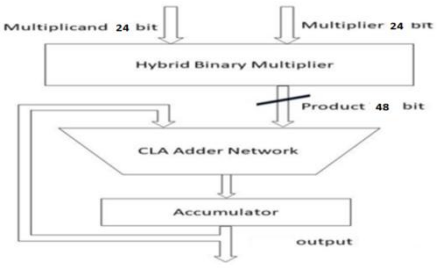

Fig 7 : 24 Bit hybrid multiplier for Mantissa

MANTISSA CALCULATION: In this design the mantissa/significand of two operands A and B are to be multiplied. As performance of mantissa multiplier dominates overall performance of the floating point multiplication, proposed 24 bit hybrid multiplier is designed as shown in figure. Here ,24 bit operands A and B is given into hybrid multiplier the partial products are given to CLA adder network then those values gets stored in the accumulator then final result of the calculated values is results as the output of 48 bits .Finally, all the resultants of the values of the sign calculation and mantissa calculation and exponent calculation are all formed into a standard IEEE-754 format representation.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology
An ISO 3297: 2007 Certified Organization Volume 9, Special Issue 1, March 2020
8th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '20)
18th-19th March 2020
Organized by
Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India

VII.SIMULATION RESULTS

The simulation of proposed single precision FPM module of internal adder and multiplier has been done to calculate the high throughput. This section includes comparative results of hybrid multiplier and previously designed multipliers. The table III shows the comparison of multipliers. Table IV shows the comparison of adders and values given were converted using tool [5][6] .Table V shows device summary of proposed single precision of FPM.Table VI shows performance analysis of proposed single precision of FPM is implemented by using Verilog language and its simulated in Xilinx ISE 14.5i on Spartan 6 .

A. *SINGLE PRECISION OF FPM*

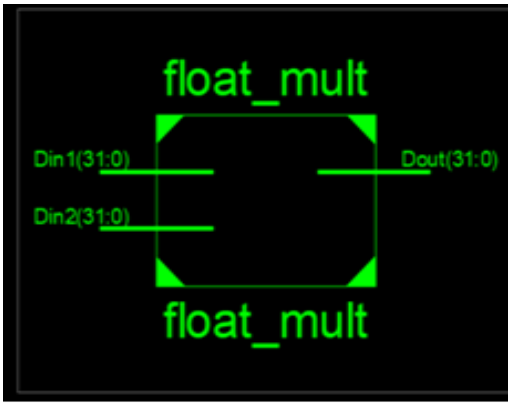

Fig 8 : RTL design of single precision FPM

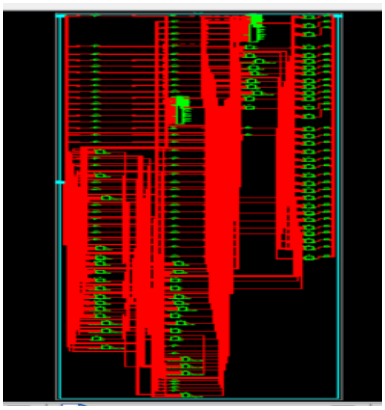

Fig 9: Technology schematic of single precision FPM

The proposed single precision FPM is designed . The proposed FPM is integrated internally with the proposed adder and hybrid multiplier and RTL design and technology schematic view is shown in figure 8,9. The simulation results of the calculated values is shown in figure 10.

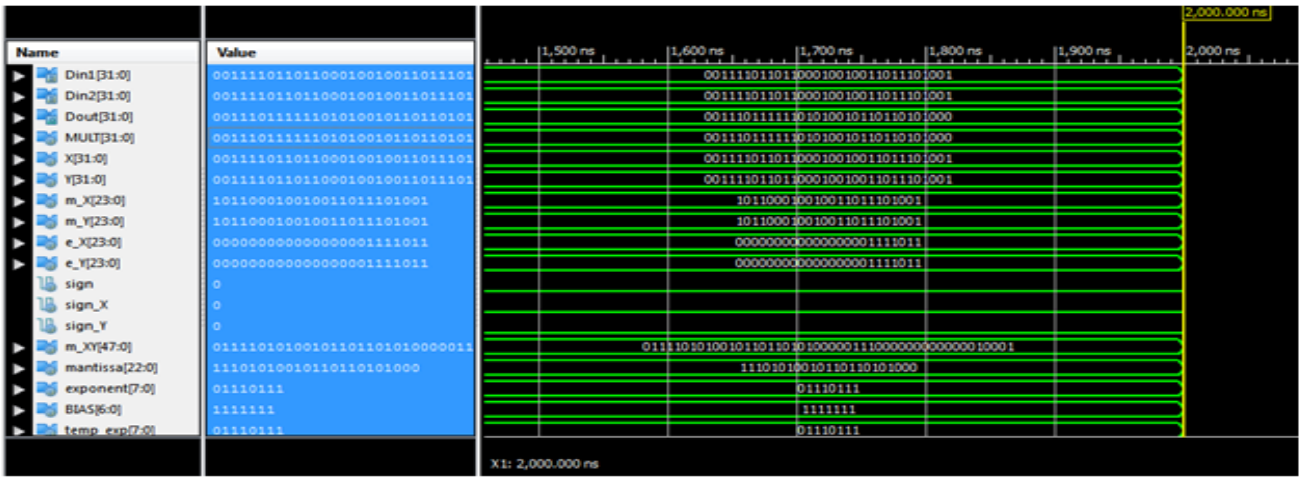

Fig 10 : Simulation results of single precision FPM

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology
An ISO 3297: 2007 Certified Organization Volume 9, Special Issue 1, March 2020
8th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '20)
18th-19th March 2020
Organized by
Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India

SINGLE PRECISION FPM: - VALUE APPLIED :0.0865* 0.0865= 0.00748225
Din1-0. 0865 =0_0111101 1_0110001 00100110 11101001
Din2- 0.0865 = 0_0111101 1_0110001 00100110 11101001
Dout- 0.00748225=0_01110111_ 1110_1010_1001_0110_1101_0100

B. *INTERNAL ADDER IN FPM*

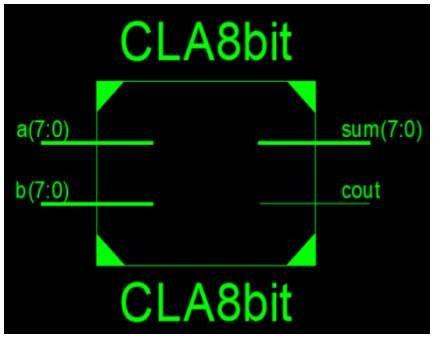

Fig 11 : RTL design 8 bit CLA adder for exponent

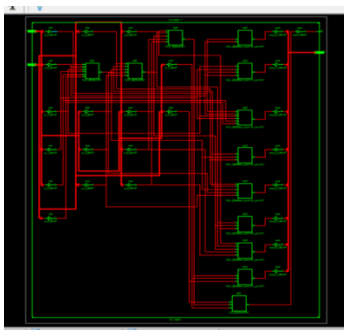

Fig 12 : Technology schematic 8 bit CLA adder for exponent

The proposed single precision FPM is internally designed with CLA adder of 8 bit for exponent part calculation . The RTL design and technology schematic view of the proposed CLA 8 bit is shown in figure 11,12. The simulation results of the calculated values is shown in figure 13, here value applied in input a is 56 and input b is 34 is added and output sum value is 90.

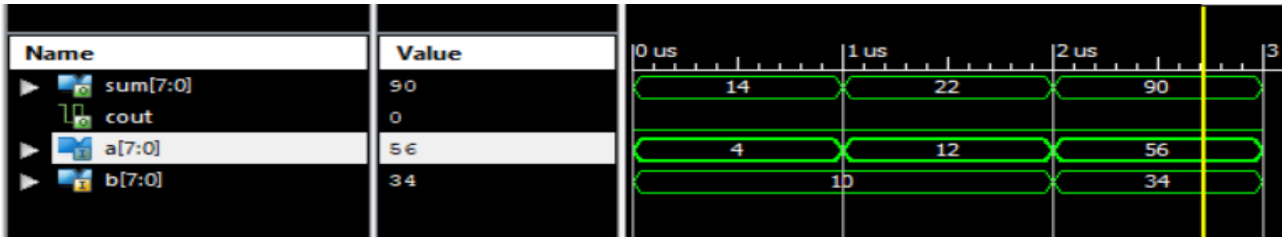

Fig 13: Simulation results of 8 bit CLA adder for exponent

C. *INTERNAL MULTIPLIER IN FPM*

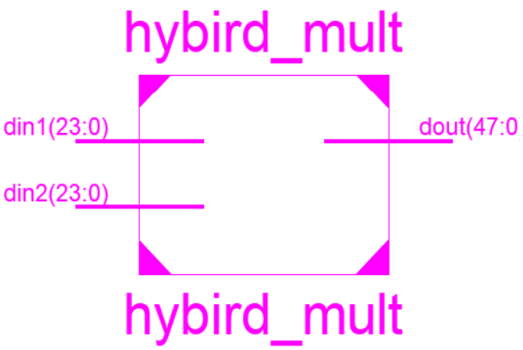

Fig 14 : RTL Design 24 bit Hybrid multiplier for mantissa

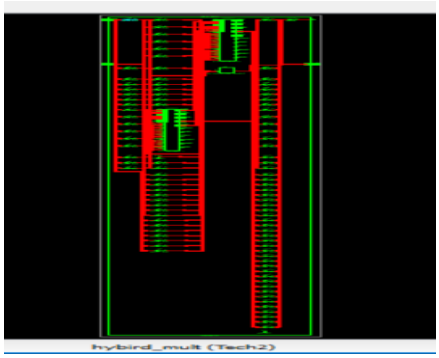

Fig 15: Technology schematic 24 bit Hybrid multiplier for mantissa

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology
An ISO 3297: 2007 Certified Organization Volume 9, Special Issue 1, March 2020
8th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '20)
18th-19th March 2020
Organized by
Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India

The proposed single precision FPM is internally designed with hybrid multiplier of 24 bit for mantissa part calculation . The RTL design and technology schematic view of the proposed hybrid multiplier of 24 bit is shown in figure 14,15. The simulation results of the calculated values is shown in figure 16, here value applied in input a is 1098 and input b is 1098 is multiplied and the output dout value is 474336.

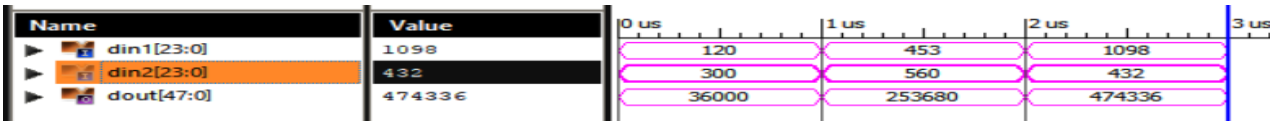

Fig 16: Simulation results of 24 bit hybrid multiplier for mantissa

MULTIPLIERS	DELAY
ARRAY MULTIPLIER	12.150ns
WALLACE MULTIPLIER	7.815ns
BOOTHS	7.583ns
HYBRID MULTIPLIER	7.045ns

TABLE III : Comparison of multipliers

ADDERS	DELAY
CARRY SELECT ADDER(CSLA)	3.504ns
CARRY SKIP ADDER(CSA)	2.925ns
PROPOSED - CARRY LOOK AHEAD ADDER(CLA)	2.527ns

TABLE IV: Comparison of adders

PRECISION	TARGET FPGA	NO OF SLICES	NO OF FLIP FLOPS	NO OF INPUT LUTS
SINGLE PRECISION	SPARTAN 6	190	308	600

TABLE V: Device summary of proposed single precision FPM

PRECISION	TARGET FPGA	FREQUENCY	DELAY
SINGLE PRECISION	SPARTAN 6	70.8 MHz	15.6ns

TABLE VI: Performance analysis of proposed single precision FPM

VIII.CONCLUSION

The demand for high performance and throughput and less delay Floating Point Multiplier (FPM) unit has been on the rise during the recent years. So the study and assessment of different adders and multipliers were considered. To overcome the challenges such as more delay and less throughput, the proposed with hybrid multiplier and adder are designed and integrated into internal modules of Floating Point multiplier(FPM) of FPU and coded in Verilog HDL using Xilinx ISE 14.5i.

IX. FUTURE SCOPE

The proposed FPM module of FPU is designed for single precision floating point numbers . It can be further extended for 64 bit, 128 bit and can also be validated in additional modules respectively. Even it can be

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology
An ISO 3297: 2007 Certified Organization Volume 9, Special Issue 1, March 2020
8th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '20)
18th-19th March 2020
Organized by
Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India

designed and tried in a hardware for even better analysis. In fact by combining hybrid multipliers and adders and additional blocks that can achieve less delay and high throughput

REFERENCES

[1] D.Goldberg,*"What every computer scientist should know about floating-point arithmetic",*ACM Computing Surveys Vol.23-1,pp.5-48,1991.
[2] L.Louca,T.A.Cook and W.H.Johnson,"*Implementation of IEEE Single Precision Floating point Addition and Multiplication On FPFAs*",Proceedings of 83rd IEEE 754 Symposium on FPGAs for Custom Computing Machines (FCCM"96),(1996),pp.107-116.
[3] B.Parhami,"*Computer Arithmetic Algorithm :Algorithms and Hardware Designs*",Oxford University Press,2000.
[4] T.Menakadevi, M.Madheswaran," Direct Digital Synthesizer using Pipelined CORDIC Algorithm for Software Defined Radio", International Journal of Science and Technology, Volume 2 No.6, June 2012,pp.372-378.
[5] Conversion Tool https://babbage.cs.qc.cuny.edu/IEEE-754.old/64bit.html
[6] Conversion Tool http://www.binaryconvert.com/convert_float.html
[7] Rashmi Rahul Kulkarni," Comparison among Different Adders", IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) Volume 5, Issue 6, Ver. I (Nov -Dec. 2015), PP 01-06
[8] https://www.johndcook.com/blog/2009/04/06/anatomy-of-a-floating-point-number/
[9] http://www.applied-mathematics.net/miniSSEL1BLAS/float-ieee754.pdf
[10] Yu-Kumg Chen ; "Fastest Carry Lookahead Adder" Proceedings of the Second IEEE International Workshop on Electronic Design, Test and Applications (DELTA'04)0-7695-2081-2/04 $ 20.00 © 2004 IEEE