TECH-CHRONICLE
AN INTERNATIONAL E-JOURNAL ON
EMERGING TRENDS IN SCIENCE, TECHNOLOGY AND MANAGEMENT

"Design and Implementation of single precision floating point multiplier with the help Xilinx on FPGA."

Miss Sneha S. Jumle (Author)
Department of Electronics,
Priyadarshani College of Engg. ,
Nagpur, Maharashtra, India
snehajumle99@gmail.com

Mrs. M.V. Vyawahare (Author)
Department of Electronics,
Priyadarshani College of Engg. ,

ABSTRACT- : Floating point arithmetic is widely used in many areas. Multipliers play an important role in today's digital signal processing and various other applications. A system's performance is generally determined by the performance of the multiplier, This paper presents an efficient FPGA implementation of Single precision floating point multiplier using VHDL. The aim of project is design and simulation of a floating point multiplier that supports the IEEE 754-2008 binary interchange format, the proposed multiplier doesn't implement rounding and presents the significant multiplier.

Keywords- *SIGNIFICANT MULTIPLIER, VHDL SIMULATION*.

I. INTRODUCTION

Floating point numbers are one possible way of representing real numbers in binary format, the IEEE 754[1] standard presents two different floating point formats, Binary interchange format and Decimal interchange format.

The aim of project is design and simulation of a floating point multiplier that supports the IEEE 754-2008 binary interchange format, the proposed multiplier doesn't implement rounding and presents the significant multiplication result as is (48 bits).

Multiplying two numbers in floating point format is done by adding the exponent of the two numbers then subtracting the bias from their result, and multiplying the significant of the two numbers, and calculating the sign by XORing the sign of the two numbers.The way floating point operations are executed depends onthe data format of the operands. IEEE standards specify aset of floating point data formats, single precision and double precision. The Single precision consists of 32 bits and the Double precision consists of 64 bits.

Figure 1

I. shows the IEEE single and double precision data formats

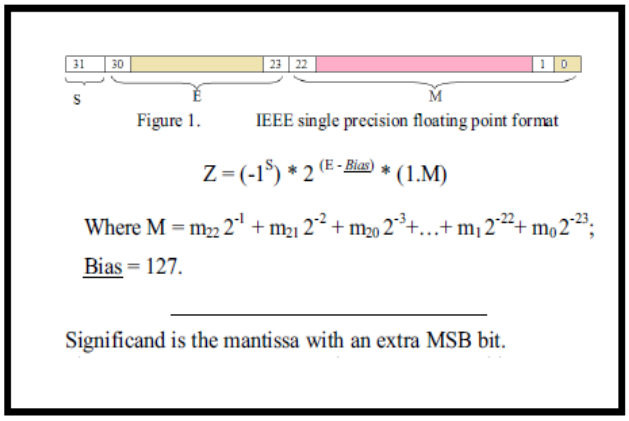


Figure 1. IEEE single precision floating point format

$$Z = (-1^{S}) * 2^{(E - \underline{Bias})} * (1.M)$$

Where $M = m_{22}2^{-1} + m_{21}2^{-2} + m_{20}2^{-3} + \ldots + m_{1}2^{-22} + m_{0}2^{-23}$;

Bias = 127.

Significand is the mantissa with an extra MSB bit.

II. Floating point multiplication algorithm

1. Add exponents
2. Multiply fractions
3. If product is 0, adjust for proper 0
4. Normalize product fraction
5. Check for exponent overflow or underflow
6. Round product fraction

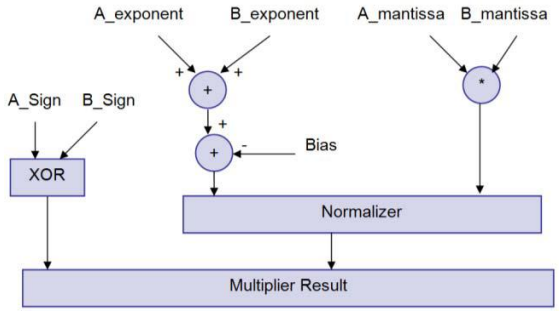
- Exponents 00000000 and 11111111 reserved
- Smallest value

- Exponent: 00000001
  Þ actual exponent = 1 – 127 = –126
- Fraction: 000…00 Þ significand = 1.0
- $\pm 1.0 \times 2^{-126} \approx \pm 1.2 \times 10^{-38}$

• Largest value
- exponent: 11111110
  Þ actual exponent = 254 – 127 = +127
- Fraction: 111…11 Þ significand ≈ 2.0

$\pm 2.0 \times 2^{+127} \approx \pm 3.4 \times$

II. PROPOSED PLAN



ROUNDING

e = -1	e = 0	e = 1
1.00 X 2^(-1) = 1/2	1.00 X 2^0 = 1	1.00 X 2^1 = 2
1.01 X 2^(-1) = 5/8	1.01 X 2^0 = 5/4	1.01 X 2^1 = 5/2
1.10 X 2^(-1) = 3/4	1.10 X 2^0 = 3/2	1.10 X 2^1= 3
1.11 X 2^(-1) = 7/8	1.11 X 2^0 = 7/4	1.11 X 2^1 = 7/2

III. SYSTEM REQUIREMENT

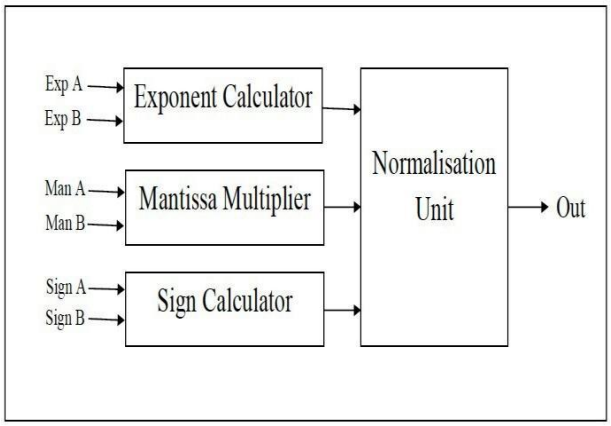
- ***Software Requirement***

Operating System: Windows XP/7
Language/Technology: VHDL
Tools: Xilinx ISE 14.5

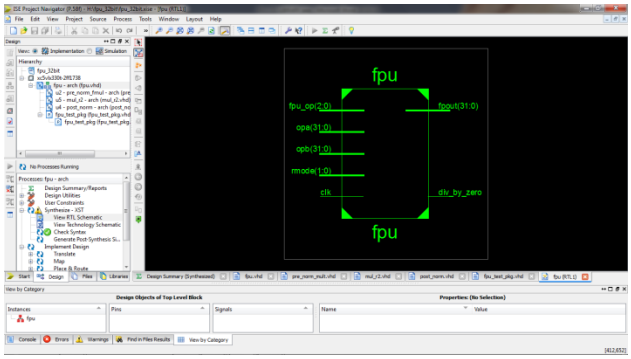
- ***Hardware Requirement***

Spartan 3 FPGA

IV. SYSTEM ARCHITECTURE



V. RTL schematic



VI. CONCLUSION

This paper presents design and simulation of a floating point multiplier that supports the IEEE 754-2008 binary interchange format, The proposed multiplier doesn't implement rounding and presents the significand multiplication result as is (48 BITS), This gives bitter precision if the whole 48 bits are utilized in another unit; i.e with a floating point adder to form a MAC Unit. But The floating point multiplier core generated by Xilinx core generated by Xilinx core generator does not indicates the 48 bits of mantissa due to rounding and is not beneficial in case DSP application of large dynamic range.

VII. *REFERENCES*

1. IEEE 754-2008, IEEE Standard for Floating-Point Arithmetic, 2008.

2. B. Fagin and C. Renard, "Field Programmable Gate Arrays and Floating Point Arithmetic," *IEEE Transactions on VLSI, vol. 2, no. 3, pp. 365–367, 1994.*

3. 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.*

4. L. Louca, T. A. Cook, and W. H. Johnson, "Implementation of IEEE Single Precision Floating Point Addition and Multiplication on FPGAs," *Proceedings of 83 the IEEE Symposium on FPGAs for Custom Computing Machines (FCCM'96), pp. 107–116, 1996.*

5. A. Jaenicke and W. Luk, "Parameterized Floating-Point Arithmetic on FPGAs", *Proc. of IEEE ICASSP, 2001, vol. 2, pp.897-900.*

6. B. Lee and N. Burgess, "Parameterisable Floating-point Operations on FPGA," *Conference Record of the Thirty-Sixth Asilomar Conference onSignals, Systems, and Computers, 2002.*

7. Mohamed Al-Ashrafy, Ashraf Salem and Wagdy Anis" An Efficient Implementation of Floating Point Multiplier" *Electronics, Communications and Photonics Conference (SIECPC), 2011 Saudi International.*