IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 6, June 2015.
www.ijiset.com
ISSN 2348 – 7968

Design And Implementation Of Double Precision Floating Point Multiplier Using VHDL

Sadiya Fatima Sufi
Dept. of electronics and telecom Engg. NUVA college of engineering and technology, Nagpur, India
Pooja Thakre
Dept. Of Electronics And Telecom. Engg. NUVA college of engineering and technology, Nagpur, India

1. ABSTRACT

This paper give the brief description of double precision floating point multiplier. We present a comparative study of double precision floating point multiplier. Floating point number system is a common device for many scientific computations due to its wide range feature.

KEYWORDS:-
HDL, Floating point multiplication, simulation, synthesis.

2. INTRODUCTION:-

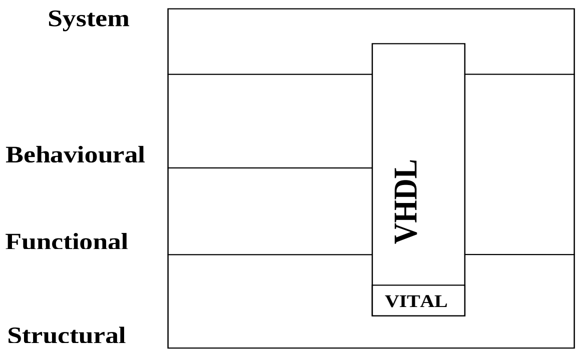
Floating numbers consists of three fields:-

Signs (S):- It is used to denote the sign of the number i.e 0 represent positive number and 1 represent negative number.

Mantissa (M):- Mantissa is a part of a floating point number which represent the magnitude of the number.

Exponent (E):- It is a part of the floating point number that represent the numbers of places that the decimal point is to be moved.

VHDL was chosen for this implementation because it is better suited for design reusability. The floating point multiplier is a subsystem that may be employed in larger floating point arithmetic unit or a digital filter implementation.



Floating point number are frequently used for numerical calculations in computing system. The term floating point is derived from the fact that there is no fixed number of digits before and after the decimal point that is the decimal point or binary can float.

The main advantage of floating point arithmetic is that wide dynamic range virtually eliminates overflow and under flow in most application.

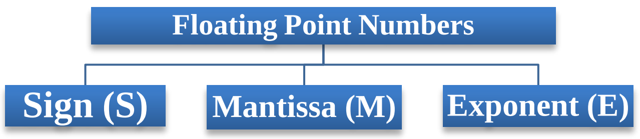


Fig. 2.

- Good design reusability
- Good design management
- Support structure predication

3. REPRESENTATION OF FLOATING POINT NUMBERS

3.1.Simple representation of floating number F th value

$$F = (-1)^s M \beta^{\in}$$

Where, S = Sign
M = Mantissa
β = Base
∈ = Exponent

3.2. The IEEE 754 standards fig below shows the IEEE single precision and double precision data format

Sign (S)	**8 Bit – Biased Exponent**	**23 bits – unsigned fraction**

Fig. 3 single precision data format

The IEEE 754 single precision (32 bits) floating point format which is widely implemented, has an 8 bit biased integer exponent which ranges between 0 to 255. The range of effective values of exponent is -127 to 128 corresponding stored values of 0 to 255 respectively.

The left most bit is the sign bit 0 for positive and bit 1 for negative numbers. There is an 8 bit exponent field E and a 23 bit Mantissa filed M. the exponent is with respect to radix 2. The extreme values of E = 0 and E = 255 are used to denote exact zero and infinity respectively.

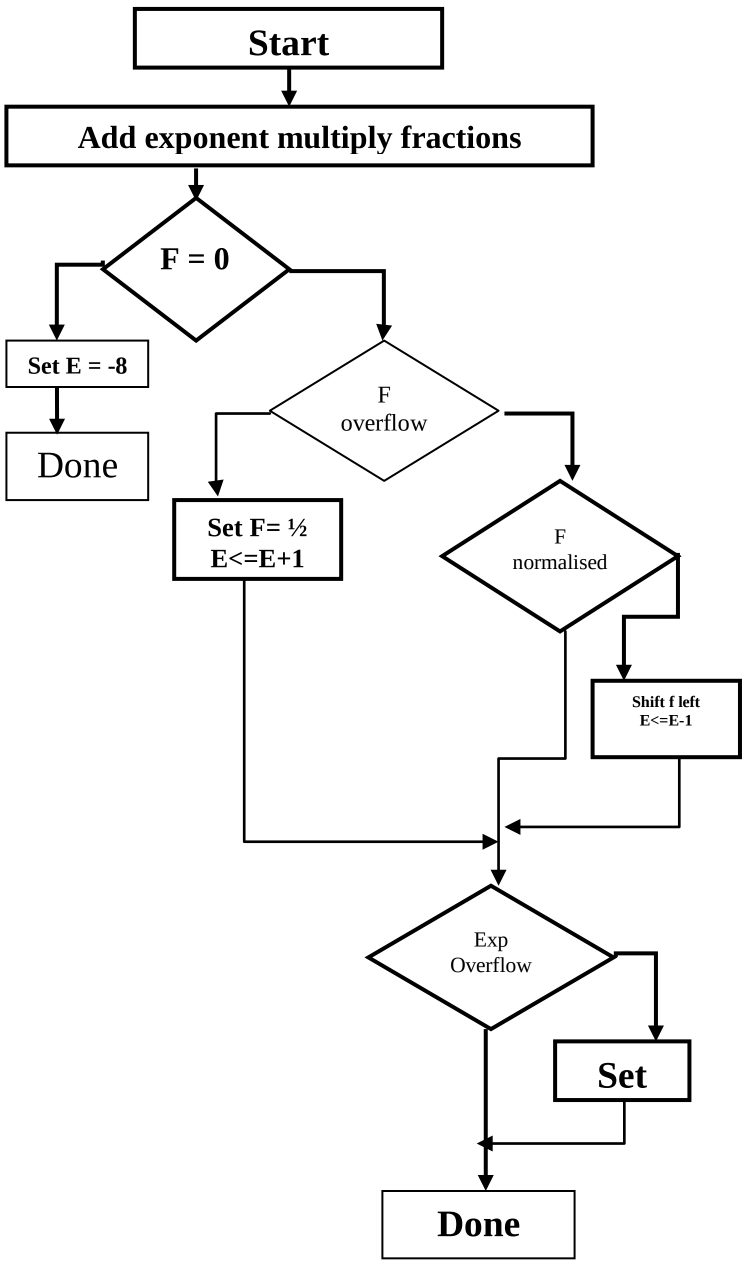
Sign (S)	**11 Bit – Biased Exponent**	**52 bits – unsigned fraction**

Fig 4. Double precision data format

The double precision format is the extension of the single precision format and uses 64 bits as shown in fig 4. Both the exponent and mantissa. Fields are larger thus allowing for greater precision and range. The exponent field has 11 bits and it is specified in the excess 1023 format where exponent = E- 1023. The normal range of the exponent is from -1022 to 1023which is represented by values of E from 1 to 2046.

4. ALGORITHM

The algorithm for floating point. Point multiplication forms the product of the operand. Significant and the sum of the operand exponents. The flow chart for the floating point multiplication algorithm is given below



5. IMPLEMENTATION

The floating point multiplier is made up of a FPU - MUL. Block and RTL view of FPU – MUL is given below :-

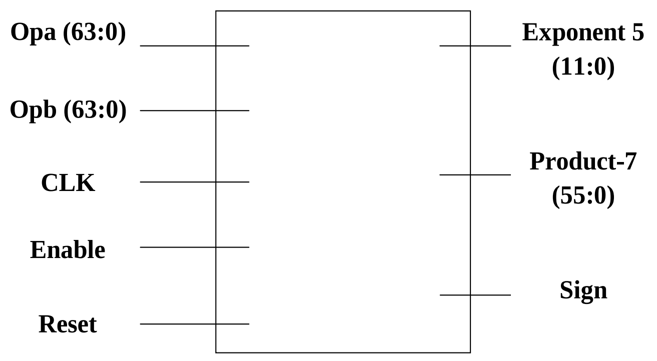


Fig.6 FPU - MUL

It consist of 5 input signals i.e operand a operand b CLK enable reset and 3 output signals exponent product and sign bit as shown in fig. 6.

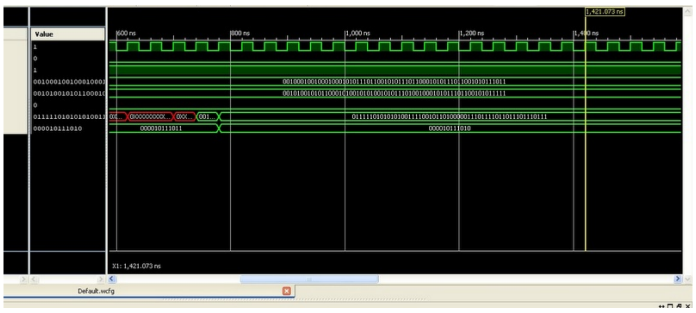


Fig. 7. Waveforms

6. SIMULATION AND SYNTHESIS

The floating point multiplier has been implemented on a Xilinx Spartan 2 FPGA simulation were carried out on the Xilinx ISE. The simulation waveforms are shown in fig.7. the synthesis result are shown in table :-

FPU – MUL Project Status			
Project file	DPMUL-Xise	Parser errors:	No errors
Module name	Fpu-mul	Implementation	Synthesized
Target device	XC6VLX 75TL-LFF484	Errors	No errors
Product versions	ISE13.1	Warnings:	5 warnings

Device Utilization Summary			
Logic utilization	used	available	utilization
Number of slice Register	952	93120	1%
Number of slice LUTS	1758	46560	3%
Number of fully used LUT-FF pairs	734	1976	37%

7. REFERENCES

1. Koren. Computer Arithmetic Algorithms, Brookside Court Publishers, Amherst, MA,1998.
2. P.Belanovi´c and M. Leeser. A Library of Parameterized Floating Point Modules and Their Use. In Proceedings, International Conference on Field Programmable Logic and Applications, Montpelier, France, Aug.2002.
3. B. Fagin and C. Renard. Field programmable gate arrays and floating point arithmetic. IEEE Transactions on VLSI Systems,2(3):365–367, Sept.1994.
4. A.A. Gaffar, W. Luk, P. Y. Cheung, N. Shirazi, and J.Hwang. Automating Customization of Floating-point Designs. In Proceedings, International Conference on Field- Programmable Logic and Applications, Montpelier, France, Aug.2002.
5. W. B. Ligon, S. McMillan, G. Monn, F. Stivers, and K.D. Underwood. A Re-evaluation of the Practicality of Floating-Point Operations on FPGAs. In Proceedings, IEEE Symposium on Field-Programmable Custom Computing Machines, pages206–215, Napa, CA, Apr.1998.