Devyani R. Kamale, 2021, 9:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

Design of Single Precision Floating Point Arithmetic Unit

Devyani R. Kamale, Lecturer Mrs. Pradnya J. Suryavanshi, Lecturer Ms. Sharda G. Mungale
Department of Electronics (VLSI) Engineering,
Priyadarshini College of Engineering,
Maharashtra, India,
devyanikamle12354@gmail.com, pradnyajs@rediffmail.com, shardamungale@gmail.com

Abstract- Most of the algorithms instigated in FPGAs used to be fixed-point. Floating-point operations are useful for calculations involving large dynamic range, but they require significantly more resources than integer operations. With the current trends in system necessities and available FPGAs, floating-point implementations are becoming more common and designers are increasingly taking benefit of FPGAs as a platform for floating-point implementations. The rapid development in Field- Programmable Gate Array (FPGA) technology makes such devices increasingly attractive for instigating floating-point arithmetic. Associated to Application Specific Integrated Circuits, FPGAs offer reduced development time and costs. Moreover, their suppleness enables field upgrade and variation of hardware to run-time conditions. A 32-bit floating-point arithmetic unit with IEEE 754 Standard has been premeditated using VHDL code and all operations of addition, subtraction, multiplication and division are verified on Xilinx.

***Keywords*- FPGAs Floating- point implementations , VHDL code etc.**

I. INTRODUCTION

The perception of floating-point representation over integer fixed-point numbers, which consist purely of significates that expanding it with the exponent component achieves superior range.

For example, in order, to denote large values, e.g. all 39 decimals are needed to be place down to femtometer-resolution in order to limit the distance between two galaxies. But if the best resolution is presumed in light years, only the 9 most significant decimal digits, where will be of importance as the remaining 30 digits purely carry noise, and thus can be securely avoided. The term floating point actually refers to the fact that a radix points of any number, (decimal point, or, more commonly in computers, binary point) can "float"; denotation that it can be tie up anywhere in relation to the

In current years, Floating-point numbers are widely espoused in many applications due to its high dynamic range and good robustness against quantization errors, capabilities. Floating-point representation is able to recollect its resolution and accuracy. IEEE specified standard for floating-point representation is acknowledged as IEEE 754 standard.

This IEEE 754 standard mandate interchange in arithmetic formats and approaches for binary and decimal floating-point arithmetic in computer programming environments. [IEEE 754- 2008] The main motive of execution of floating-point operation on reconfigurable hardware is to use fewer chip area with less combinational delay [Karan Gumber et.al, May 2012] which means less latency i.e. quicker

speed. A parameterizable floating point adder and multiplier instigated using the software-like language Handel-C, using the Xilinx XCV1000 FPGA, five stages pipelined multiplier accomplished [A. Jaenicke et. Al ,2001]. T

he hardware looked-for for the parallel 32-bit multiplier is approximately 3 times that of serial. A single precision floating point multiplier that does not facilitates rounding modes can be instigated using a digit-serial multiplier [L. Louca et. al, 1996]. The ALU is an important building block of the CPU of a computer, and even the modest microprocessors contain one for determinations such as maintaining timers. By using pipeline along with ALU design, ALU offers a high performance. With pipelining concept ALU accomplish multiple instructions at the same time.

In IEEE 754, the IEEE has uniform standardized the computer illustration for binary floating-point numbers. Almost all modern machines shadow this standard. Computer manufacturers are in scuffle to conform to a common representation and arithmetic convention for floating point data.

The standard outlines the Arithmetic presentations for binary and decimal floating-point data sets, which knows finite numbers (including signed zeros and subnormal numbers), infinities, and special "not a number" values Interchange formats encodings (bit strings) that may be employed to convert floating-point data in a improved and compact form.

- **Rounding Rules:** Gratification of certain properties requires during arithmetic and conversions to accomplish rounding of numbers.
- **Operations:** Arithmetic and few other operations on arithmetic formats.
- **Exception Handling:** Suggestions of conditions such as division by zero, overflow, etc.

A 32-bit word is essential for the IEEE single precision floating point standard representation involves whose bits may be represented as numbered from 0 to 31, left to right. Figure 1 displays the format of single precision floating point standard.

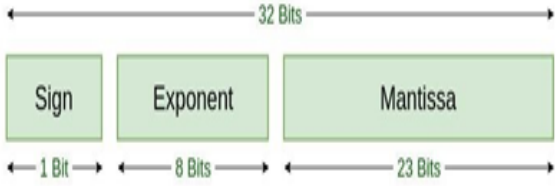

Fig 1. IEEE 754 single precision floating point standard.

Sign bit designate whether the number is positive or negative. If it shows '1' then the number is negative and if it shows '0' then the number is positive. "Exponent" is of 8 bit which offers the exponent range from E (min) =-126 to E (max) =127. The fractional part of a number is assumed by the Mantissa which is of 23 bits. The mantissa must not be muddled with the significand. The "1" in the significand is made implicit.

III. FPU ARCHITECTURE

Figure 2 shows the architecture of floating-point unit. This is a modest single precision floating point unit. Two pre-normalization units regulate the fractions.One does it for add and subtract action, the other for multiply and divide operation.

Table 1. The FPU provisions the following mode.

R mode	Rounding Mode
0	Round to nearest even
1	Round to Zero
2	Round to +INF (UP)
3	Round to -INF (DOWN)

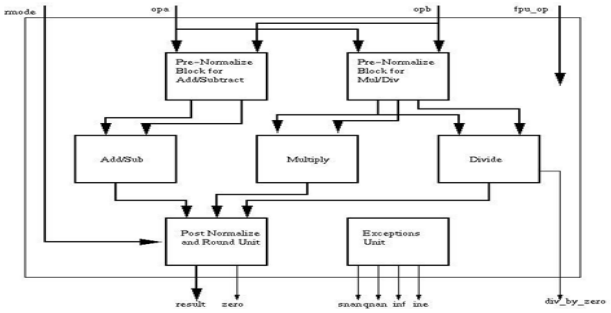

Fig 2. Architecture of floating-point unit.

IV. DESIGN METHODOLOGY

The FPU can accomplish a floating-point operation every cycle. It will handle the operation type, rounding mode and operands and carry a result fourcycles later. The FPU will never engender a SNAN output.

The output of SNAN is proclaimed at the time when one of the operands was a signaling NAN (output will be a quiet NAN). When carrying out a floating point to integer conversion, the output (representing an integer) can take on forms of a NAN or INF, which are impeccably legal integers.

1. Addition/ Subtraction:

While adding the two floating point numbers, two cases may get to your feet.

- **Case 1:** Both the numbers are of identical sign i.e. when both the numbers are any +ve or -ve. In such case most significant bit of both the numbers are 1 or 0.
- **Case 2:** Two numbers have diverse signs revenue two numbers check the sign of the two numbers. If the Sign of any of the two number is diverse then take the 2's complement of the respective number and then add the two numbers.

2. Multiplication:

Revenue the two normalized operands. Multiply the significands. Then add the exponents determining the sign. Normalize mantissa and update exponent. Find exception flags and govern special values for over flow and underflow.

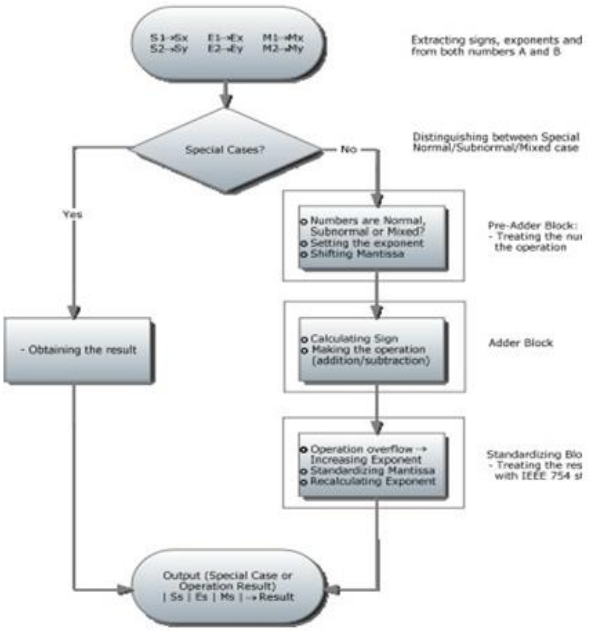

Fig 3. Flow of addition/subtraction.

3. Division:

Divide significands, subtract exponents, and determine their sign. Normalize mantissa and apprise exponent. Find exception flags and determine special values for over flow and underflow.

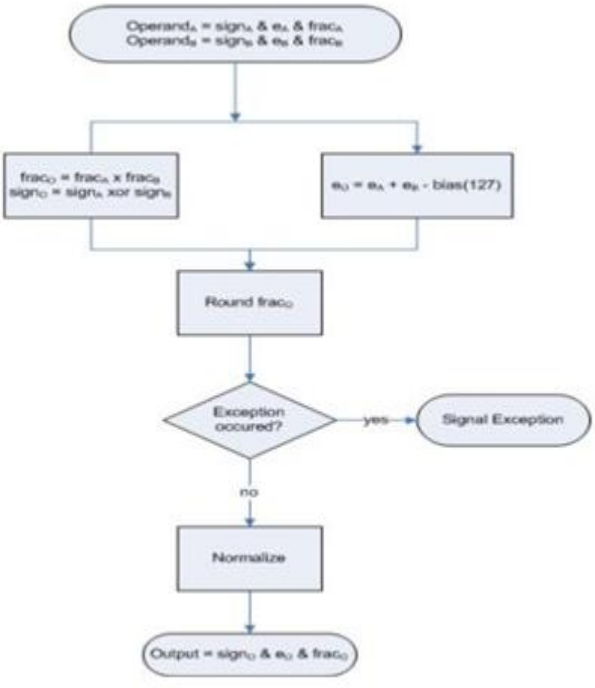

Fig 4. Flow of multiplication.

V. PROPOSED DESIGN

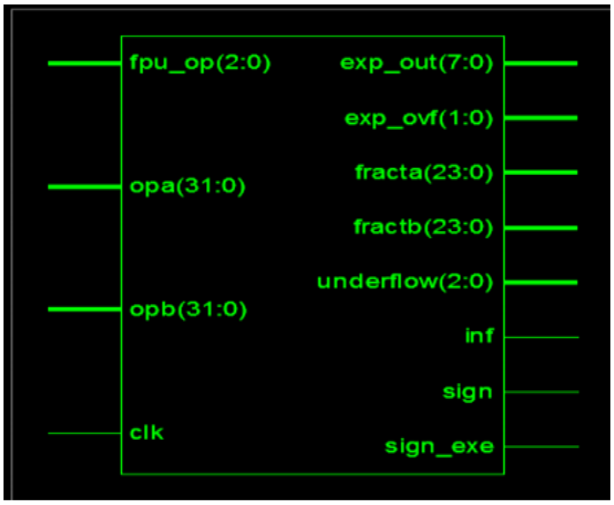

Fig 5. Top level entity of proposed FPU.

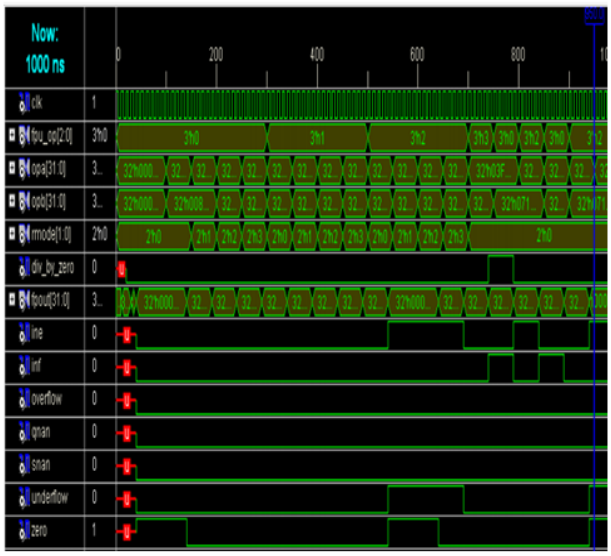

Fig 6. Waveform of FPU 32 Bit.

VI. CONCLUSION AND FUTURE WORK

The Floating-point Arithmetic unit have been deliberated and suitable algorithm has been established to perform operation such as addition, subtraction, multiplication and division. The algorithm can be instigated in pipelined way to diminish the delay and upsurge the computation time for operation. IEEE 754 standard based floating- point depiction also can be cast-off to operation like square root. A double precision Floating point Multiplier can be premeditated for multiplication intensive applications, such as DSP or graphics, could benefit several high-performance multipliers on same chip. A high output multiplier or several multipliers waged on same chip can be used for single chip video signal processing.

REFERENCES

[1] Mr. Ankit Trivedi, Mr. Apoorv, Amana Yadav Axis Institute of Technology & Management Kanpur, up, India "A Review on Single Precision Floating Point Arithmetic Unit of 32-bit Number" International Research Journal of Engineering and Technology (IRJET)e-ISSN: 2395-0056 Volume: 06 Issue: 04 | Apr 2019.

[2] Nikhila S, B Yamuna and Karthi Balasubramanian Deepak Mishra Digital Communication Division (DCD), Space Application Center (SAC), ISRO, Ahmedabad, "FPGA Based Implementation of aloating Point Multiplier and its Hardware Models" 2019 IEEE 16th India Council International Conference (INDICON) Electronic ISSN: 2325-9418©2019 IEEE.

[3] Mr. Anand Burud ,Dr. Pradip Bhaskar Department of technology, Shivaji university, Kolhapur, India "Design and Implementation of FPGA Based 32 Bit ating Point Processor for DSP Application" Fourth International Conference on Computing mmunication Control and Automation (ICCUBEA)-1-5386-5257-2©2018 IEEE.

[4] Amana Yadav, Ila Chaudhary, Manav Rachna, International University, Faridabad "Design of 32-bit Floating Point Unit for Advanced Processors "Int. Journal of Engineering Research and Application www.ijera.com ISSN: 2248-9622, Vol. 7, Issue 6, (Part-5) June 2017, pp.39-46.