International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 10 | Oct 2020 www.irjet.net p-ISSN: 2395-0072

Design and Analysis of Floating Point Arithmetic Unit: A review

Naresh Kumar[1], Onkar Singh[2], Harjit Singh[3]

[1]M.Tech Scholar Arni University, Indora HP
[2]Assistant Professor ECE Department, Arni University, Indora HP
[3]Assistant Professor EEE Department, Arni University, Indora HP

Abstract - *A floating point arithmetic-logic unit is the part of a computer system that carries out arithmetic and logic operations on the floating point numbers in computer instruction word. In some processors, the FPAU is divided into two units, a floating point arithmetic unit and a floating point logic unit. Some processors contain more than one Arithmetic unit - for example, one for fixed-point operations and another for floating-point operations. Generally floating point arithmetic and logic unit (FPALU) performs arithmetic operations like addition, subtraction, multiplication and division. In this paper different technique of floating point arithmetic unit design and their outcomes are discussed.*

Key Words: **Floating Point, Precision, Exception, Latch, Embedded.**

1. INTRODUCTION

A floating point arithmetic-logic unit is the part of a computer system that carries out arithmetic and logic operations on the floating point numbers in computer instruction word. In some processors, the FPAU is divided into two units, a floating point arithmetic unit and a floating point logic unit. Some processors contain more than one Arithmetic unit - for example, one for fixed-point operations and another for floating-point operations. Generally floating point arithmetic and logic unit (FPALU) performs arithmetic operations like addition, subtraction, multiplication and division. This also performs logical operations like AND, OR, shift left and shift right etc. To represent very small or very large values, large range is required as the integer representation is no longer appropriate to represent those numbers. So values can be represented floating point unit by using the IEEE-754 standard[1].

Arithmetic operations are very important in the design of digital design systems, computer applications and application specific integrated systems. Arithmetic circuits form an important class of circuits in digital systems. Day by day with the progress in the very large scale integration circuit technology, many complex circuits, which are very hard to design, have become easily realizable today. Due to advancement in the technology the complex algorithms are easy to design and implement. This means that the designing to complex circuits are easily possible nowadays[3].

In modern general purpose computer architectures, one or more than one floating point units are integrated with the processor; however many older processors, especially older designs, do not have hardware support for floating-point operations. Now almost every language has a floating point data type; computers from simple computers to supercomputers have floating-point units; most compilers will be called upon to compile floating point algorithms from time to time; and every operating system must respond to floating point exceptions such as overflow.[4]

1.1Representation of Floating Point Numbers

The floating point numbers are so called floating because there is no fixed number of digits after and before of the decimal point that means the decimal point can float. In fixed point representation the number of digits after and before of decimal point is fixed. The main advantage of floating point numbers is that they can handle a large range of values in comparison to fixed point numbers. The floating point numbers are slower than fixed point numbers.

1.1.1 IEEE Single Precision Format: The IEEE single precision format consists of 32 bits to represent a floating point number, divided into three subfields, as illustrated in figure 1.1. The first field is the sign bit for the fraction part. The next field consists of 8 bits which are used for exponent the third field consists of the remaining 23 bits and is used for the fractional part. The sign it reflects the sign of the fraction it is 0 for positive numbers and 1 for negative numbers. In order to represent a number in the IEEE single precision format, first it should be converted to a normalized scientific notation with exactly one bit before the binary point, simultaneously adjusting the exponent value.

Sign	Exponent	Fraction
1 bit	8 bits	23 bits

Figure 1.1: IEEE single precision floating point format [4]

1.1.2 IEEE Double Precision Format: The IEEE double precision format consists of 64 bits to represent a floating point number, as illustrated in figure 1.3. The first bit is the sign bit for the fraction part. The next 11 bits are used for the exponent, and the remaining 52 bits are used for the fractional part. As in the single precision format, the

sign bit is 0 for positive numbers and 1 for negative numbers. The exponent representation used in the second field is obtained by adding the bias value of 1023 to the actual exponent of the number in the normalized form

S	Exponent	Fraction
1 bit	11 bits	52 bits

Figure 1.2: IEEE double precision floating-point format [4]

1.2 Latch Based Design

Basically flip flops are made from latches i.e. latches combine together and make a flip flop. Flip flop can be either simple that means transparent or opaque or clocked that means synchronous or edge triggered. The simple one are commonly called latches. The word latch mainly used for storage elements, while clocked devices are described as flip flop. Latch based design are generally made for high frequency circuits. In high speed circuits maintaining a clock skew is a problem by flip flop but latch solves this problem. Latches main advantage is that they allow a sufficiently long combinational path which determines the maximum frequency of the design. In latch to latch stages they borrow some time from a shorter path to make its timing performance effective. This technique is called time borrowing technique. [6]

1.3 Advantages of floating Point Representation

The main advantage of floating point format is that they have much wider range of values in comparison to fixed point format. Another advantage of floating point number is that they are more flexible than fixed point numbers which has a limited or no flexibility [5]. Other major advantages are there exponentially vastly increased dynamic range available for many applications. This large dynamic range is very useful in dealing with larger values. The internal representation of data in floating point format is more exact than fixed point format [8].

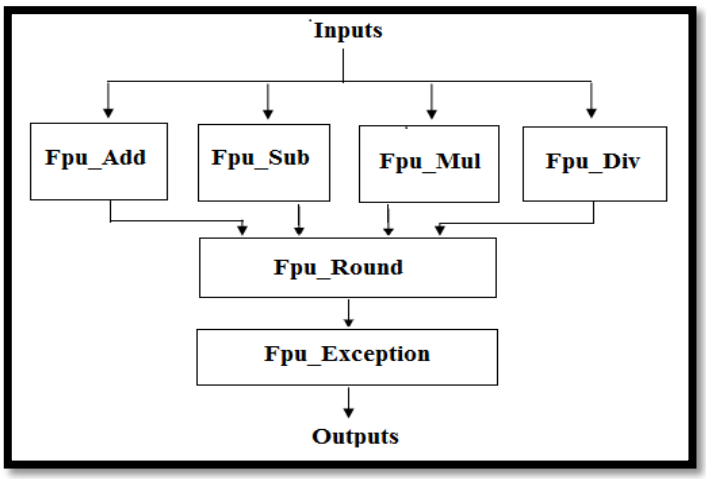
1.4 Applications of floating-point representation

Scientific and higher engineering applications demand exceptionally high floating point performance which in turn requires high speed floating point units to reduce executing time. Floating Point units are used in high speed objects recognition system and also in high performance computer systems as well as embedded systems and mobile applications [2]. Floating point units are widely used in digital applications such as digital signal processing, digital image processing and multimedia [5]. In medical image recognition, greater accuracy supports the many levels of signal input from light, x-rays, ultrasound and other sources that must be defined and processed to create output images with useful diagnostic information. Wide dynamic range is essential to radar, where a system may need to track over a range from zero to infinity, and then use only a small subset of that range for target acquisition and identification. A wide dynamic range can also allow a robot to deal with unpredictable conditions, such as an obstruction to its normally limited range of motion. By contrast with these applications, the enormous communications market is better served by floating-point devices [8].

The floating point format is also very useful for audio and video applications. Since the audio and video applications requires a large set of data to perform their functions. As floating point format have large range of data so it will useful for audio and video applications. Audio needs wider range of values than video applications that requirement id fulfilled by floating point hardware [6]. Floating point unit performs addition, subtraction, multiplication, division, square root etc that are widely used in large set of scientific, commerce, financial and in signal processing applications [7].

1.5 Floating Point Arithmetic Unit

The block diagram of the proposed floating point arithmetic unit is given in figure 1.3 The unit supports four arithmetic operations: Add, Subtract, Multiply and Divide. All the basic mathematical arithmetic operations have been carried out in four separate modules one for addition, one for subtraction, one for multiplication and one for division as shown in figure.



The floating point arithmetic unit consist of following blocks

Fpu_Add- Floating Point adder
Fpu_Sub- Floating Point Subtractor
Fpu_Mul-Floating Point Multiplier
Fpu_Div- Floating Point Division
Fpu_Round-Floating Point Rounding Unit
Fpu_Exception- Floating Point Exception Unit

2. RELATED WORK

[1] C. Rami Reddy and O.Homa Kesav proposed a high speed ASIC implementation of a floating point arithmetic unit which can perform addition, subtraction, multiplication, division functions on 32-bit operands that use the IEEE 754-2008 standard. Pre-normalization unit and post normalization units are also discussed along with exceptional handling. All the functions are built by feasible efficient algorithms with several changes incorporated that can improve overall latency, and if pipelined then higher throughput. The algorithms are modeled in Verilog HDL and the RTL code for adder, subtractor, multiplier, divider, square root are synthesized using Cadence RTL complier where the design is targeted for 180nm TSMC technology with proper Constraints. The design has been synthesized with TSMC 0.18 I-lm Logic Salicide 1.8V/3.3V 1 P6M process technology. Strategies have been employed to realize optimal hardware and power efficient architecture. The layout generation of the presented architecture using the backend flow is an ongoing process and is being done using Cadence RTL compiler with 180nM process technology.

[2] Amana Yadav and Ila Chaudhary proposed process for the computation of addition, subtraction and multiplication operations on floating point numbers. It has been designed using VHDL. The design has been simulated and synthesized to identify the area occupied and its performance in terms of delay. The arithmetic operations on floating point unit are quite complicated. They are represented in IEEE 754 format in either 32-bit format (single precision) or 64-bit format (double precision). They are extensively used in high end processors for various applications such as mathematical analysis and formulation, signal processing etc. floating point unit has been designed, simulated and then synthesized in order to obtain its performance in terms of the area occupied and delay on Vitex 5 FPGA Module. For the data path opb_in_fpu to opb_in_sig_0 total combinational logic delay and routing delay is 1.154ns and total overflow to overflow delay is 3.259ns. Hardware requirements have also been specified in the paper. Pre normalization and post-normalization units of the FPU can be further optimized to reduce the hardware requirement as well as delay.

[3] Naresh Grover and M.K.Soni discussed that the Floating-point operations are useful for computations involving large dynamic range, but they require significantly more resources than integer operations. With the current trends in system requirements and available FPGAs, floating-point implementations are becoming more common and designers are increasingly taking advantage of FPGAs as a platform for floating-point implementations. The rapid advance in Field-Programmable Gate Array (FPGA) technology makes such devices increasingly attractive for implementing floating-point arithmetic. Compared to Application Specific Integrated Circuits, FPGAs offer reduced development time and costs. Moreover, their flexibility enables field upgrade and adaptation of hardware to run-time conditions. A 32 bit floating point arithmetic unit with IEEE 754 Standard has been designed using VHDL code and all operations of addition, subtraction, multiplication and division are tested on Xilinx. Thereafter, Simulink model in MAT lab has been created for verification of VHDL code of that Floating Point Arithmetic Unit in Modelsim. A process described to create Simulink model in MAT lab for verification of VHDL code in Modelsim HDL Simulator has been used on the same VHDL code and results were found in order. Once the Simulink model has been created using MAT lab for VHDL code, the same can be optimized in MAT lab and the VHDL code can be regenerated with the optimized results and tested on Xilinx to see the improvement in the parameters.

[4] Bhaskar Chittaluri proposed a method in which they apply the novel constructs related to the different considerations of the VLSI. Unlike the traditional methods we adopted betterment multiplication algorithm and the addition of the partial product. For accomplishment of floating point operational unit the standard IEEE format can be acquired. The proposed concept provides the additional solution that able to perform the many operations such as addition, subtraction multiplication and division with accurate response and effectiveness. From out of the previous methods this proposal may affords the less footprint with more accuracy. In the proposed design he achieved the less partial product concern to the traditional multiplication algorithm and he also concentrated on the advanced method for the partial product addition many cases some improvement towards the speed and also the pipelined operation also can be achieved by using our design. By this he concluded that this is well suited for the high speed application and also less die requirements.

[5] Shaikh Shoaib Arif and Dr.B.B.Godbole discussed wide range of DSP applications includes processing of sensor array processing, audio and speech signal processing, control of systems, radar and sonar signal processing, spectral estimation, digital image processing, seismic data processing, biomedical signal processing, statistical signal processing, signal processing for communications, Filter designing & many high accuracy based operations. Floating point operations are used due to its huge dynamic range, high accuracy and straightforward operation rules. With the increasing needs for the floating point operations for the high-speed signal processing and the scientific operation, the requirements for the high-speed hardware floating point arithmetic units have become more useful. Authors proposed a inbuilt architectures which support 32 bits, 64 bits & 128

bits of operations. This improvement in hardware & software will be utilized in digital signal processing units, sonar and radar signal processing, sensor array processing, spectral estimation, statistical signal processing and high precision based applications and results will improve.

REFERENCES

[1] C.Rami Reddy, O.Homa Kesav, A.Maheswara Reddy "High Speed Single Precision Floating Point Unit Implementation Using Verilog" International Journal of Advances in Electronics and Computer Science, ISSN: 2393-2835 Volume-2, Issue-8, Aug.-2015

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

[3] Naresh Grover and M.K.Soni " Design of FPGA based 32-bit Floating Point Arithmetic Unit and verification of its VHDL code using MATLAB" I.J. Information Engineering and Electronic Business, 2014, 1, 1-14 Published Online February 2014 in MECS

[4] Bhaskar Chittaluri "Implementation of Area Efficient IEEE-754 Double Precision Floating Point Arithmetic Unit Using Verilog" International Journal of Research Studies in Science, Engineering and Technology Volume 2, Issue 12, December 2015, PP 15-21

[5] Shaikh Shoaib Arif and Dr.B.B.Godbole "Multi-Precision Floating Point Arithmetic Logic Unit for Digital Signal Processing" International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE) Vol 5, Issue 2, February 2018

[6] Yedukondala Rao Veeranki, R. Nakkeeran "Spartan 3E Synthesizable FPGA Based Floating-Point Arithmetic Unit" International Journal of Computer Trends and Technology (IJCTT), volume-4, Issue-4, pp.751-755, April 2013

[7] Jongwook Sohn, Earl E. Swartzlander "Improved Architectures for a Fused Floating Point Add-Subtract Unit" IEEE Transactions on Circuits and Systems-I: regular papers, Vol. 59, No. 10, pp. 2285-2291, October 2012

[8] KavithaSravanthi, Addula Saikumar "An FPGA Based Double Precision Floating Point Arithmetic Unit using Verilog" International Journal of Engineering Research & Technology (IJERT), Vol. 2 Issue 10, pp. 576-581, October – 2017

[9] H. Yamada, T. Hottat, T. Nishiyama, F. Murabayashi, T. Yamauchi, and H. Sawamoto "A 13.3ns Double-precision Floating-point ALU and Multiplier", IEEE International Conference on Computer Design: VLSI in Computers and Processors, pp. 466 – 470, 2-4 Oct 1995

[10] Addanki Puma Ramesh, A. V. N. Tilak, A.M.Prasad "An FPGA Based High Speed IEEE-754 Double Precision Floating Point Multiplier Using Verilog" 2013 International Conference on Emerging Trends in VLSI, Embedded System, Nano Electronics and Telecommunication System (ICEVENT), pp. 1-5, 7-9 Jan. 2017

[11] Ushasree G, R Dhanabal, Sarat Kumar Sahoo "Implementation of a High Speed Single Precision Floating Point Unit using Verilog" International Journal of Computer Applications National conference on VSLI and Embedded systems, pp.32-36, 2019

[12] Shamna.K, S.R Ramesh "Design and Implementation of an Optimized Double Precision Floating Point Divider on FPGA", International Journal of Advanced Science and Technology, Vol. 18, pp.41-48, May 2016

[13] Shrivastava Purnima, Tiwari Mukesh, Singh Jaikaran and Rathore Sanjay "VHDL Environment for Floating point Arithmetic Logic Unit - ALU Design and Simulation" Research Journal of Engineering Sciences, Vol. 1(2), pp.1-6, August 2012

[14] Hwa-Joon Oh, Silvia M. Mueller, Christian Jacobi, Kevin D. Tran, Scott R. Cottier "A Fully Pipelined Single-Precision Floating-Point Unit in the Synergistic Processor Element of a CELL Processor" IEEE Journal of Solid-State Circuits, Vol. 41, No. 4, pp. 759-771, April 2006

[15] Tarek Ould Bachir, Jean-Pierre David "Performing Floating-Point Accumulation on a modern FPGA in Single and Double Precision" 18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines, pp.105-108, 2012

[16] Tashfia.Afreen, Minhaz. Uddin Md Ikram, Aqib. Al Azad, and Iqbalur Rahman Rokon "Efficient FPGA Implementation of Double Precision Floating Point Unit Using Verilog HDL" , International Conference on Innovations in Electrical and Electronics Engineering (ICIEE'2012) , pp.230-233, Oct. 6-7, 2018

[17] Per Karlstrom, Andreas Ehliar, Dake Liu "High Performance, Low Latency FPGA based Floating Point Adder and Multiplier Units in a Virtex 4", 24th Norchip Conference, pp. 31 – 34, Nov. 2016.