Floating Point Addition, Subtraction and Multiplication on FPGA

[1]G. Mohana Durga,[2]D. Bhavani

[1]Assistant Professor, [2]Assistant Professor
[1]Electronics and Communication Engineering,
[1]Sagi Rama Krishnam Raju Engineering College, Chinnamiram,India

Abstract- **Floating point operations are widely used in many applications and in different areas to perform mathematical operations very accurately, Digital Signal Processing and Digital Image Processing algorithms. These operations are hard to implement on Field Programmable Gate Arrays (FPGAs) because of the complexity of their algorithms but many scientific problems require floating point arithmetic [3] with high levels of accuracy in their calculations. Floating point arithmetic architecture is designed and implemented on FPGA become easier by using high level language such as Verilog HDL. This paper explores FPGA implementations of addition, subtraction and multiplication for IEEE-754 single precision floating point numbers. Here 24 bit multiplier is designed with small 4 bit multipliers. The implementation is performed using Xilinx's Spartan 3 FPGAs.**

Index Terms: **Field Programmable Gate Array, Hardware Implementation, Float Point Arithmetic, Xilinx, Verilog HDL.**

I.INTRODUCTION

Field programmable gate arrays (FPGAs) it is feasible to provide custom hardware [2] for application specific computation design. Floating point implementation on FPGAs is a challenging problem because floating point numbers require more fields than fixed point numbers and availability of physical resources on FPGAs (memory, gates, etc.) is limited. Image and digital signal processing applications require high floating point calculations through put and FPGA is used for performing these Digital Signal Processing (DSP) operations. Floating point operations are hard to implement on FPGAs as their algorithms are quite complex. The floating point implementations on FPGAs require bit-width variation as a means to control precision. Implementing floating point adders and multipliers [4] on FPGAs, which meet IEEE 754 floating point format, here we study implementation of various floating point arithmetic operations such as addition, subtraction and multiplication [1] Xilinx's Spartan 3 FPGAs. This is invaluable tools in the implementation of high performance systems, combining the reprogrammability advantage of general purpose processors with the speed and parallel processing. More recently, the increasing size of FPGA devices allowed researchers to efficiently implement operators in the 32-bit single precision format. Double precision and quad precision described more bit operation so at same time we perform 32 and 64 bit of operation of arithmetic unit.

II. FLOATING POINT FORMAT

IEEE Standard for Binary Floating Point Arithmetic [4] (ANSI/IEEE Std 754-1985) will be used throughout our work. The single precision format is shown in Figure 1. Numbers in this format are composed of the following three fields:

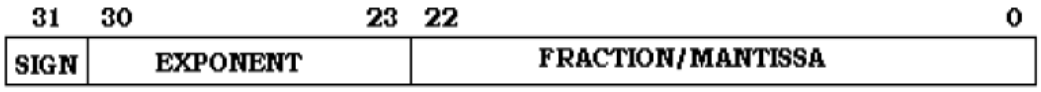


FIG 1: FLOATING POINT FORMAT USED

1-bit sign, S: A value of '1' indicates that the number is negative, and a '0' indicates a positive number.

Bias-127 exponent, e = E+bias: This gives us an exponent range from Emin = -126 to Emax = 127 **Fraction, f/mantissa:** The fractional part must not be confused with the significand, which is 1 plus the fractional part. The leading 1 in the significand is implicit. When performing arithmetic with this format, the implicit bit is usually made explicit. To determine the value of a floating point number in this format we use the following formula:

Value = ((-1) ^sign) X 2^ (exponent-127) X 1.f22 f21 f20 f19……f2 f1 f0

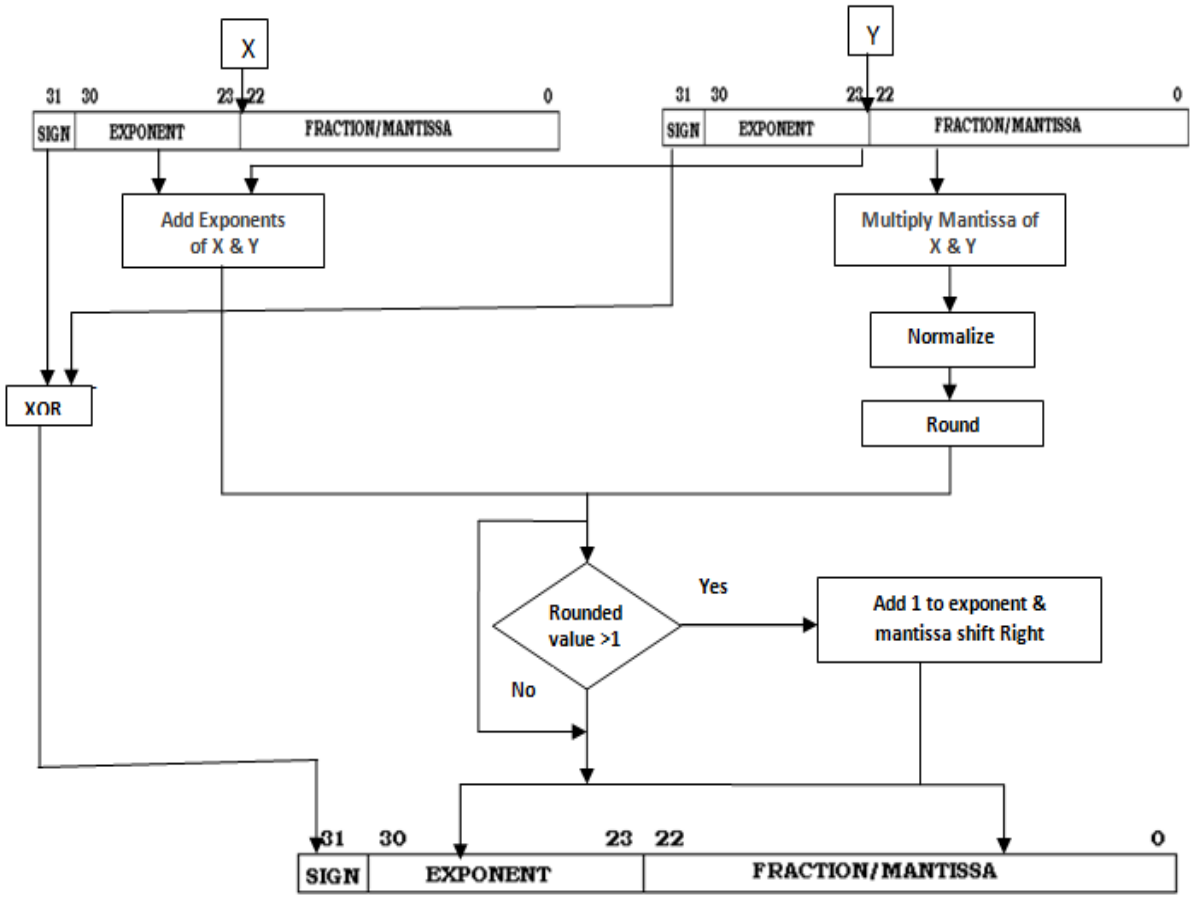
Positive floating-point numbers in this format have an approximate range of 10^{-308} to 10^{308}, because the range of the exponent is [−1022, 1023] and 308 are approximately $\log_{10}(2^{1023})$. The complete range of the format is from about -10^{308} through $+10^{308}$

Features of single precision floating number are represented in Table 3.1

Feature	Single Precision Number
Word length	32
Significant bits	23+1(hidden)
Significant Range	[1,2-2-23]
Exponent Bits	8
Exponent Bias	127
Zero	E + bias = 0, f = 0
Not-a-Number (NAN)	E + bias = 255, f != 0
Denormal	E + bias = 0, f != 0
Infinity	E + bias = 255, f = 0
Minimum	2^-126 to 1.2 *10^ -38
Maximum	~2^128 to 3.4 * 10^ 38

TABLE 3.1: FEATURES OF THE ANSI/IEEE STANDARD FLOATING POINT REPRESENTATION

III.FLOATING POINT MULTIPLIER



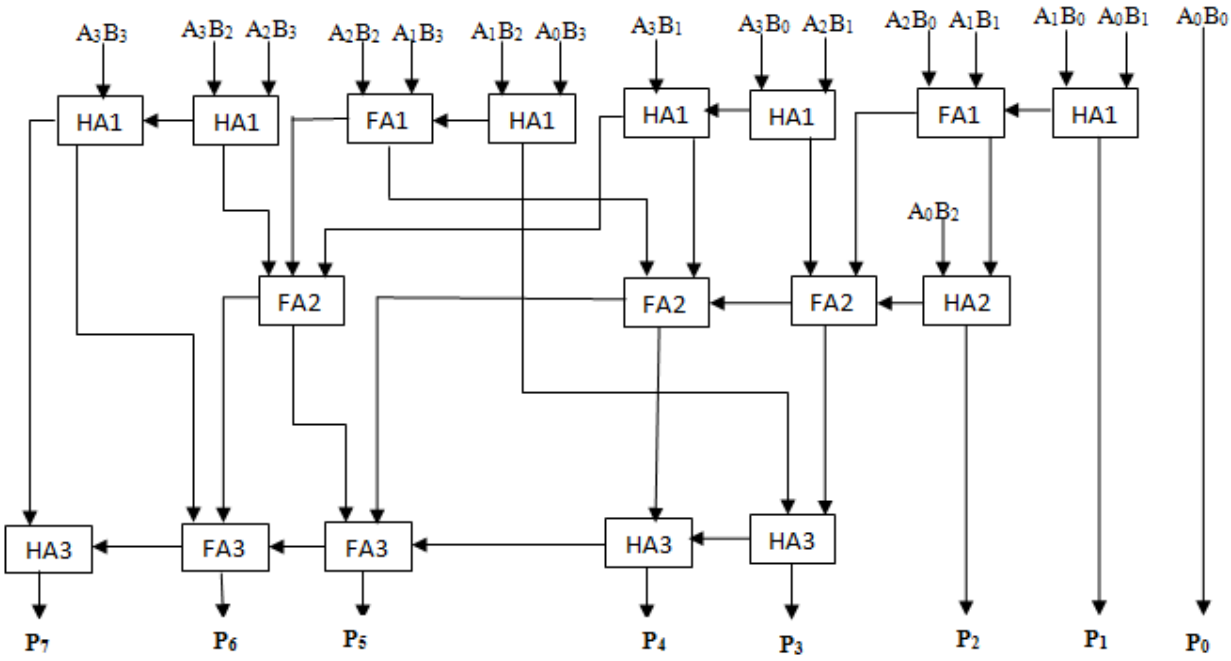
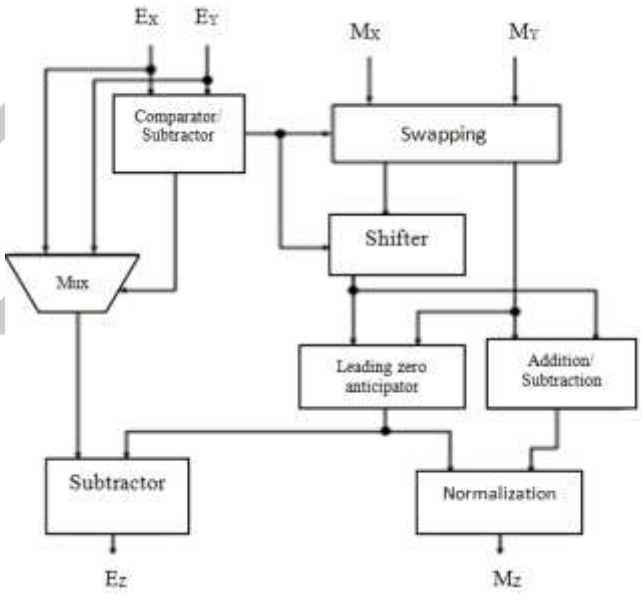


FIG: 4 Bit Multiplier

IV. FLOATING POINT ADDITION and SUBTRACTOR



V.RESULTS

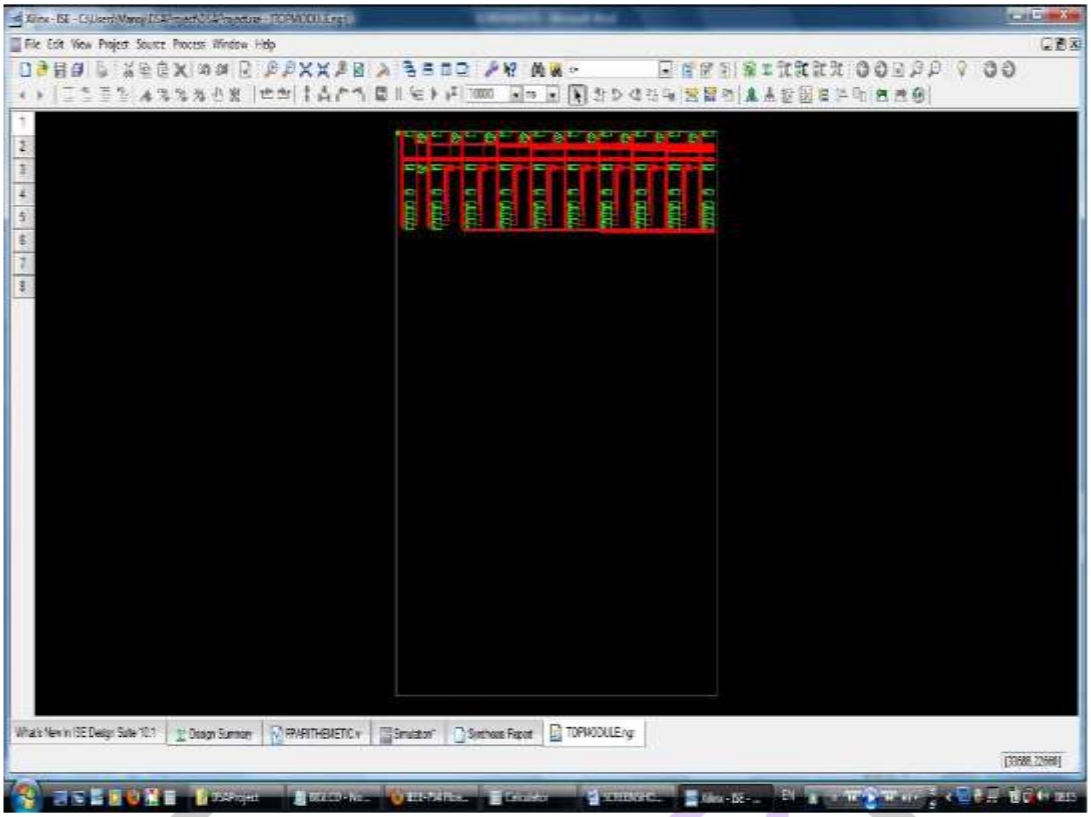


FIG: RTL SCHEMATIC

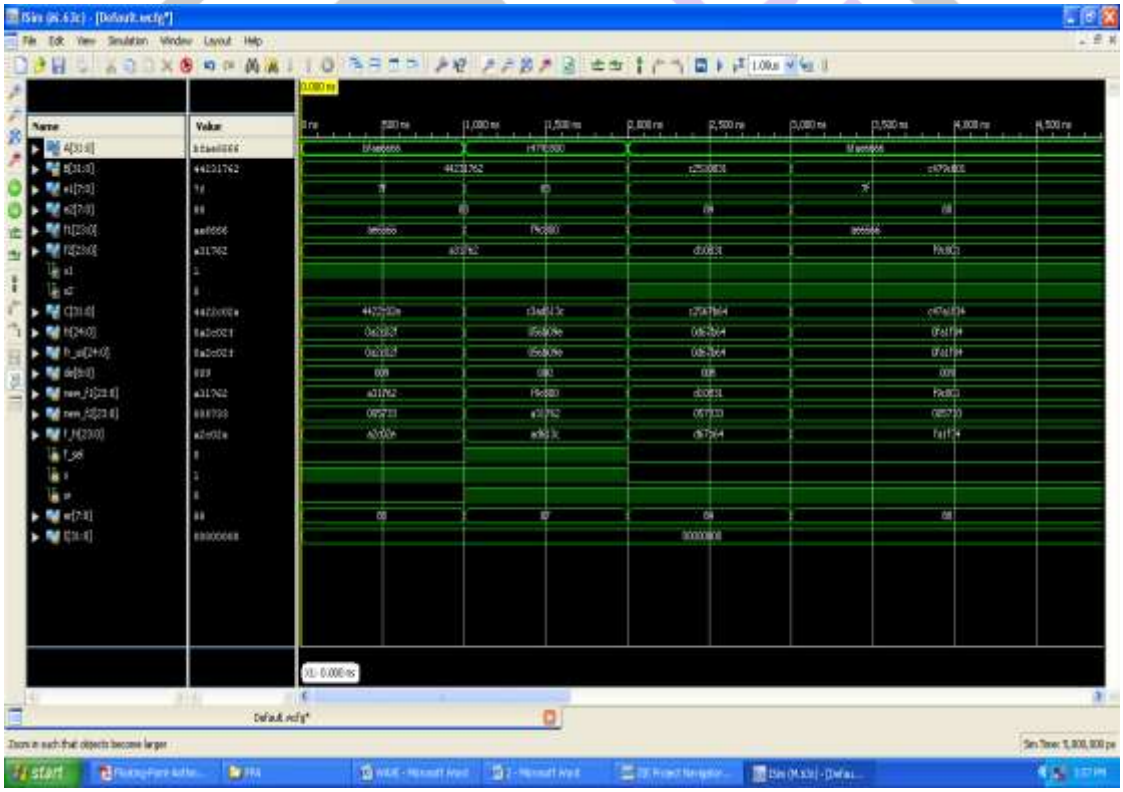


FIG: FLOATING POINT ADDER

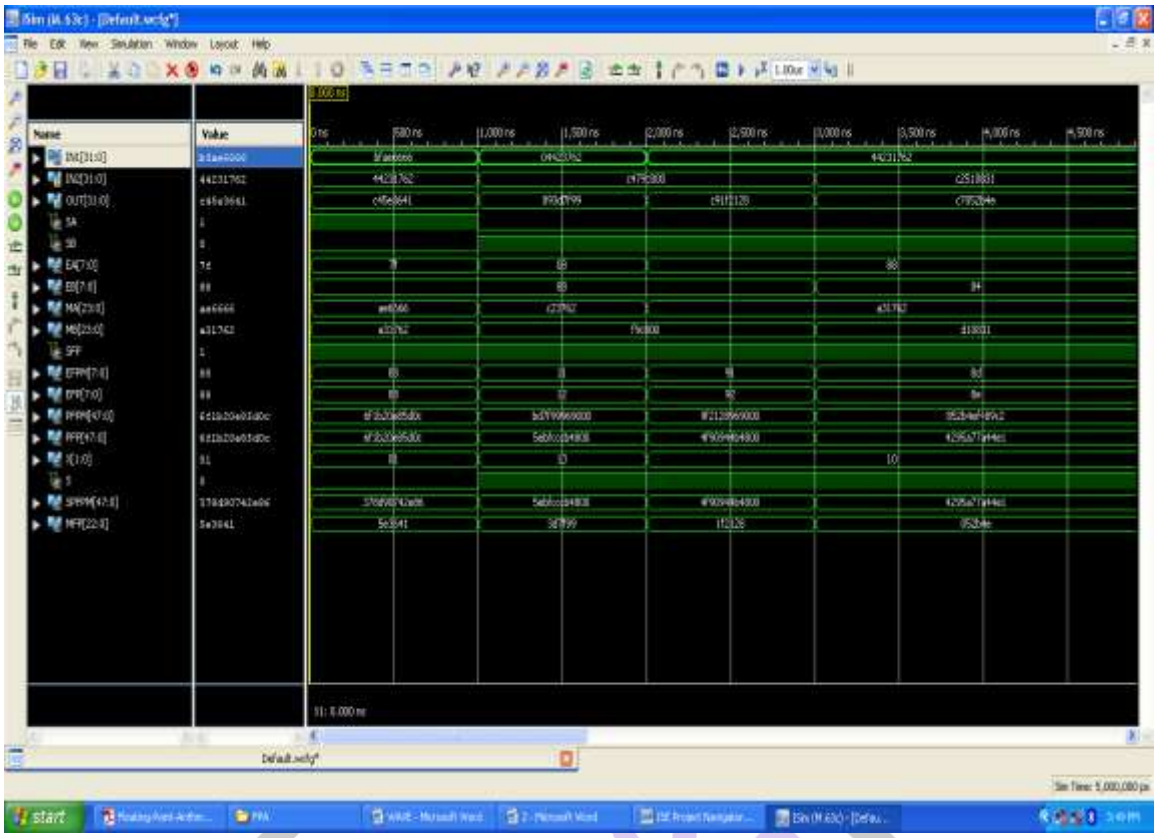


FIG: FLOATING POINT MULTIPLIER

VI. CONCLUSION

Hence Floating point arithmetic (addition, subtraction & multiplication) is successfully implemented on IEEE single precision floating point numbers on FPGA. This paper presented an efficient realization of floating Adder, subtractor and multiplier, which is required for signal processing applications. Simulation results show that the proposed design has an improvement of 13.6% in terms of delay and around 6.3% improvement in power-delay product when compared with existing architecture.

References

[1] H.H. Saleh, "Fused Floating-Point Arithmetic for DSP," PhD dissertation, Univ. of Texas, 2009

[2] B. Parhami, "Computer Arithmetic: Algorithms and Hardware Designs", 2nd edition, Oxford University Press, New York, 2010.

[3] "IEEE Standard for Floating-Point Arithmetic", in IEEE Std 754-2008 , vol., no., pp.1-70, Aug. 29 2008.

[4] R. K. Saxena, S. Neelam and A. K Wadhwani, "Design of Fast Pipelined Multiplier using Modified Redundant Adder", Int. J. Intelligent Syst. Applicat. (IJISA), vol. 4, no.4, pp. 47-53, 2012.

[5] IEEE Task P754, IEEE 754-2008, Standard for Floating-Point Arithmetic. New York, NY, USA: IEEE, Aug. 2008.