ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 16, April 2017

Performance Analysis of Floating Point Arithmetic

P.Pushpalatha [1], Pinapothu Srilakshmi [2]

Assistant professor, Department of Electronics and Communication Engineering, UCEK, JNTUK Kakinada, Andhra Pradesh, India[1]

P.G. Student, Department of Electronics and Communication Engineering, UCEK, JNTUK Kakinada, Andhra Pradesh, India[2]

pushpalatha86@gmail.com [1], Srilakshmi.hyma@gmail.com [2]

ABSTRACT: This paper mainly presents performance analysis of floating point arithmetic. Floating point arithmetic performs addition, subtraction, multiplication and division. This paper briefly explains about the floating addition, subtraction and multiplication operations. These operations are used in numerical calculations, FFT and butterfly operations. Floating point numbers are one possible way of representing real numbers in binary format. The arithmetic units in modern microprocessors execute advanced applications such as 3D graphics, multimedia, signal processing, and a variety of scientific computations that require complex mathematic computations. The fixed-point number system is not sufficient to handle such complex computations. In contrast, the floating-point notation, which is specified in the IEEE-754 Standard for floating-point arithmetic, represents a wide range of numbers from tiny fractions to extremely large numbers. Floating point multiplier is simple in terms of overall structure it requires more logic area and delay compared to the floating-point adder and floating point subtraction.

KEYWORDS: Floating point, Floating point multiplication, Floating point arithmetic.

I. INTRODUCTION

Floating point numbers are one possible way of representing real numbers in binary format. The IEE 754 standard presents two different floating point formats, binary interchange format and decimal interchange format. Multiplying floating point numbers is a critical requirement for DSP applications involving large dynamic range. This paper focuses on the floating point arithmetic in single precision. Fig. 1 shows the IEEE 754 single precision binary format representation. It consists of a one bit sign (S), an eight bit exponent (E), and a twenty three bit fraction (M or Mantissa). An extra bit is added to the fraction to form what is called the significand1. If the exponent is greater than 0 and smaller than 255, and there is 1 in the MSB of the significand then the number is said to be a normalized number. The arithmetic units in modern microprocessors execute advanced applications such

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 16, April 2017

as 3D graphics, multimedia, signal processing, and a variety of scientific computations that require complex mathematic computations. The fixed-point number system is not sufficient to handle such complex computations. In contrast, the floating-point notation, which is specified in the IEEE-754 Standard for floating-point arithmetic, represents a wide range of numbers from tiny fractions to extremely large numbers. Floating point representation equation

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

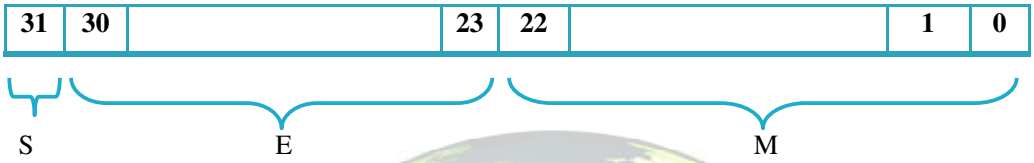


Fig 1: IEEE 754 single precision format.

II. FLOATING POINT ARITHMETIC

Arithmetic functions on floating point numbers consist of addition, subtraction, multiplication and division. Floating Point numbers are frequently used for numerical calculations in computing systems. Arithmetic units for floating-point numbers are more complex than fixed-point numbers. Floating-point arithmetic is being used for 3-D graphics and automotive control, among other applications. Modern mobile processors are also fitted with FPUs for their performance benefits. [4] proposed a system, this paper presents an effective field programmable gate array (FPGA)-based hardware implementation of a parallel key searching system for the brute-force attack on RC4 encryption. The design employs several novel key scheduling techniques to minimize the total number of cycles for each key search and uses on-chip memories of the FPGA to maximize the number of key searching units per chip. Based on the design, a total of 176 RC4 key searching units can be implemented in a single Xilinx XC2VP20-5 FPGA chip. Operating at a 47-MHz clock rate, the design can achieve a key searching speed of 1.07 x 107 keys per second. Breaking a 40-bit RC4 encryption only requires around 28.5 h.

III. FLOATING POINT ADDITION

Addition is one of the simplest and commonly used operations and is in most cases a speed determining factor for arithmetic operations. The addition of two binary numbers is the fundamental arithmetic operation in microprocessors, digital signal processors, and data-processing application-specific integrated circuits. Addition of two numbers in floating point format is done by

1. Compare the exponents of the two numbers and shift smaller number to right until its exponent would match the larger exponent.
2. Add the significands.
3. Normalization the sum, either shifting right and incrementing exponent or shifting left and decrementing the exponent.
4. Check the result overflow or underflow.
5. Round the significand to the appropriate number of bits and finally obtain floating point addition result.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 16, April 2017

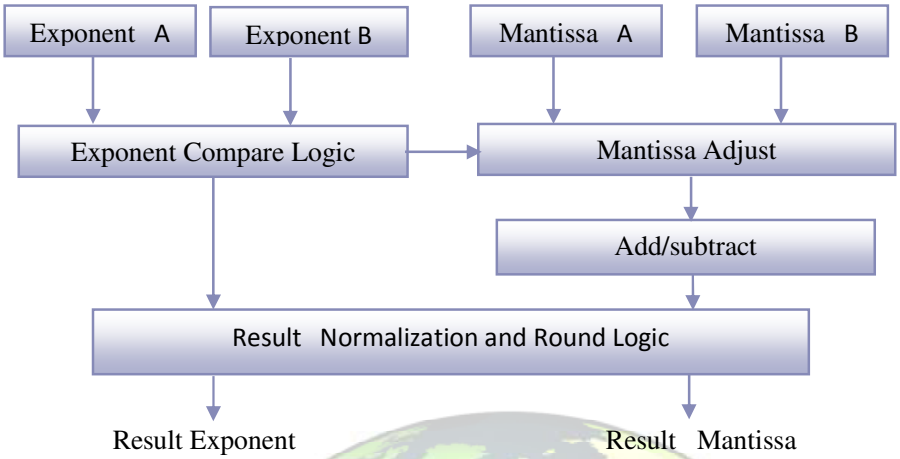


Fig 2: Addition/Subtraction of floating point numbers.

IV. FLOATING POINT SUBTRACTION

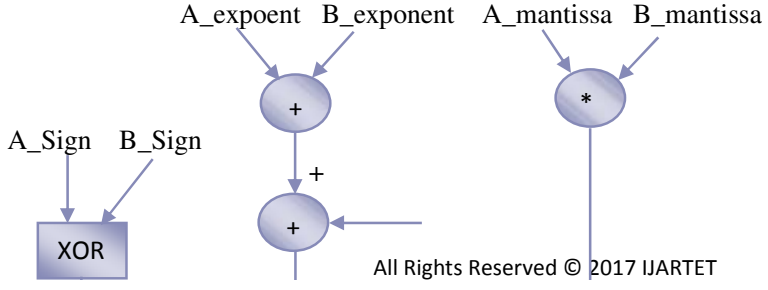
A simple method to subtract floating point numbers is to first represent them with the same exponent. Subtraction of two numbers in floating point format is done by

1. Compare the exponents of the two numbers and shift smaller number to right until its exponent would match the larger exponent.
2. Subtraction the significands.
3. Normalization the sum, either shifting right and incrementing exponent or shifting left and decrementing the exponent.
4. Check the result overflow or underflow.
5. Round the significand to the appropriate number of bits and finally obtain floating point subtraction result.

V. FLOATING POINT MULTIPLICATION

The floating-point multiplier takes two input operands and produces a rounded product result. Although the floating-point multiplier is simple in terms of overall structure, it requires more logic area and power consumption compared to the floating-point adder.

Multiplying two numbers in floating point format is done by adding the exponent of the two numbers then subtracting the bias from their result, multiplying the significand of the two numbers and calculating the sign by xoring the sign of the two numbers. In order to represent the multiplication result as a normalized number there should be 1 in the MSB of the result.



ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 16, April 2017

- Bias

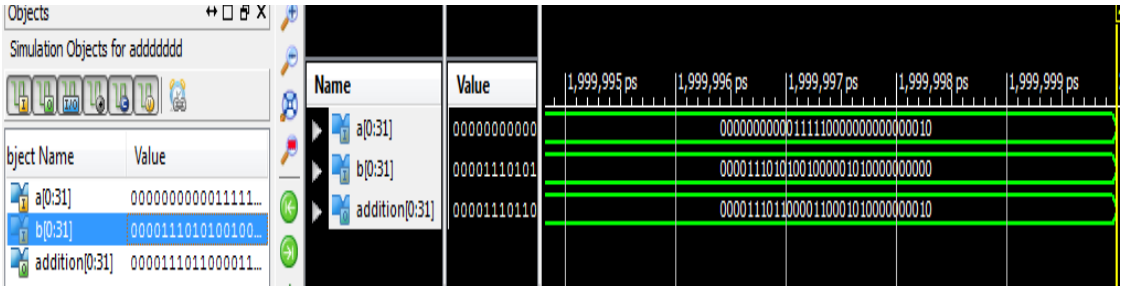
Fig 3: floating point multiplication

To multiply two floating point numbers the following is done:
1. Multiplying the significand (1.M1*1.M2).
2. Placing the decimal point in the result.
3. Adding the exponents (E1 + E2 – *Bias*).
4. Obtaining the sign s1 xor s2.
5. Normalizing the result obtaining 1 at the MSB of the results significand.
6. Rounding the result and finally obtain floating point multiplier result.

VI. RESULT COMPARSION TABLE

S.NO	TYPE OF OPERATION	AREA	DELAY
1.	ADDITION	489 slices	23.690 ns
2.	SUBTRACTION	509 slices	22.996 ns
3.	MULTIPLING	694 slices	31.07 ns

VII. SIMULATION RESULT



ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 16, April 2017

Fig 4: floating point addition

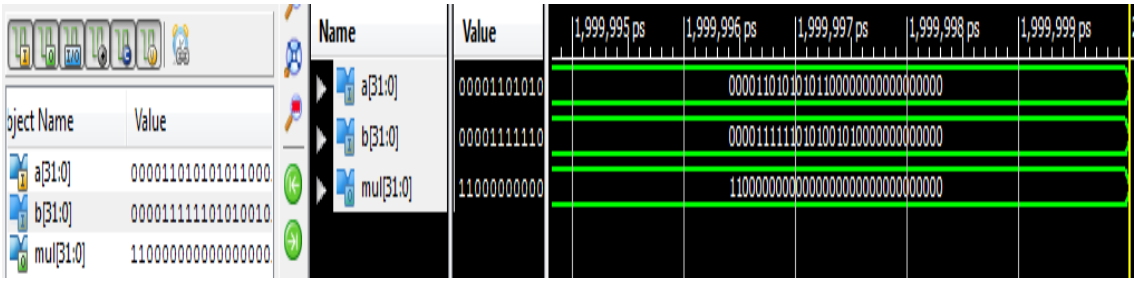


Fig 5: floating point multiplication

VI. CONCULSION

Floating arithmetic operations are most commonly used mathematical operations in numerical calculations. In this mainly discuss about the performance analysis of the floating point addition, floating point subtraction and floating point multiplication. Floating point multiplication requires more area and delay when compare to addition and subtraction.

REFERENCES

[1]. IEEE Standard for floating point arithmetic, IEEE Standard 754-2008, New York, Inc., Aug.29, 2008.
[2].IEEE Standard 754 Floating Point Numbers, by Steve Hollasch.
[3]. Floating point arithmetic unit using verilog, by lalitha gongavar and rajan chaudhary.
[4]. Christo Ananth, Muthamil Jothi.M, M.Priya, V.Manjula, "Parallel RC4 Key Searching System Based on FPGA", International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE), Volume 2, Special Issue 13, March 2016, pp: 5-12
[5]. 32 bit Single Precision floating point Multiplier, Ms.Radhika Jumde, AVBIT, Pawnar, Wardha.
[6]. Normalization on floating point Multiplication using Verilog HDL, V.Narasimha, V. Swathi.
[7]. Floating Point Adder and Multiplier, Eduardo Sanchez EPFL- HEIG- VD. Aug-2013.