ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 4, Issue 6, June 2015

Single Precision Floating point Numbers Multiplication using standard IEEE 754

Tejaswini.H.N.
Asst Professor, Dept of ECE
Sambhram Institute of Technology
Bengaluru, India

Dr. Ravishankar. C. V
Professor & HOD, Dept of ECE
Sambhram Institute of Technology
Bengaluru, India

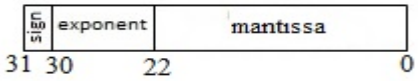
Abstract- Floating-point arithmetic involves manipulating exponents and shifting fractions, the bulk of the time in floating point operations is spent operating on fractions using integer algorithms. This paper proposes the multiplication of floating point numbers. IEEE Standard 754 floating point is the most common representation today for real numbers on computers, including Intel-based PC's, Macintoshes, and most Unix platforms. In this paper, IEEE 754 standard which is the standardized computer representation for binary floating-point numbers is used. It is the most commonly used representation in modern computing machines. This standard defines several different precisions. The most popular formats are single precision and double precision.

I. INTRODUCTION

Several different representations of real numbers have been proposed. The most widely used is the floating point representation. This section describes the survey that was done on the single precision floating point numbers multiplication. The Multiplication of the floating point numbers is done using IEEE 754 standard which is the standardized computer representation for binary floating-point numbers

In the past, each machine had its own implementation of floating point arithmetic hardware and software. It was impossible to write portable programs that would produce the same results on different systems. It wasn't until 1985 that the IEEE 754 standard was adopted. Having a standard at least ensures that all compliant machines will produce the same outputs for the same program. The IEEE standard gives an algorithm for addition, subtraction, multiplication, division and square root. Thus, when a program is moved from one machine to another, the results of the basic operations will be the same in every bit if both machines support the IEEE standard [7]. The IEEE has standardized the computer representation for binary floating-point numbers in IEEE 754 standard. The IEEE 754 standard defines several different precisions. Most popular are Single precision numbers and Double precision numbers. Double precision numbers have an one sign bit, 11-bit exponent field and a 52-bit mantissa, for a total of 64 bits.

We have done floating point multiplication for Single precision numbers which include an one sign bit, a 8-bit exponent field and a 23-bit mantissa, for a total of 32 bits. The standard mandates binary floating point data be encoded on three fields: a one bit sign field, followed by exponent bits encoding the exponent offset by a numeric bias specific to each format, and bits encoding the significand



The scientific notation of IEEE 754 standard numbers is shown below and are stored in signed magnitude format.

$$\pm \text{mantissa} * 2^{\text{exponent}}$$

The sign bit is 0 for positive numbers and 1 for negative numbers.

II. ARCHITECTURE

The Single Precision Unit Architecture comprised of two Operands A and B. The RM MODE is the round unit mode. For Normalization we are using two Pre Normalize block, one for Addition and Subtraction, other for Multiplication.

.

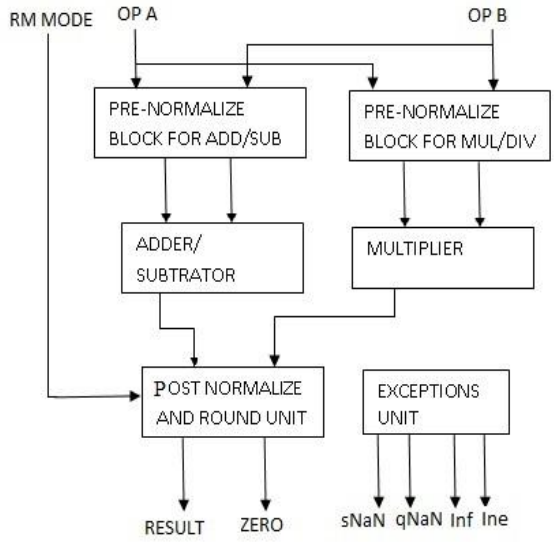


Fig 1 Architecture of Single Precision Unit

1. Pre Normalize Block for Adder and Subtractor : Calculate the difference between the smaller and larger exponent. Adjust the smaller fraction by right shifting it, determine if the operation is an add or subtract after resolving the sign bits. Check for NaNs on inputs.

2. Pre Normalize Block for Multiplication: Computes the sum/difference of exponents, checks for exponent overflow, underflow condition and INF value on an input.
   i. Add and Sub - 24 bit integer adder and subtractor.
   ii. Multiply - 2 cycle 24-bit boolean integer multipler

3. Post Normalize and Round Unit - Normalize fraction and exponent. Also do all the roundings in parallel and then pick the output corresponding to the chosen rounding mode.

4. Exceptions Unit – This unit
   Generates the exception signals like sNAN, qNAN, Inf and Ine
   The IEEE standard defines two classes of NaNs(non numbers):
   i. quiet NaNs (qNaNs) : A qNaN is a NaN with the most significant fraction bit set.
   ii. signaling NaNs (sNaNs): A sNaN is a NaN with the most significant fraction bit clear.

A. Arithmetic and Logic Unit for Floating Point Numbers (32-bit)

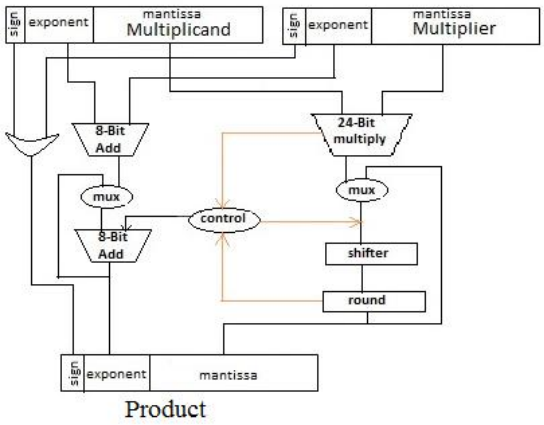


Fig 2.. Block Diagram of ALU supporting floating point Addition, Subtraction and Multipliction

The single precision floating point format is divided into three main parts corresponding to the sign , exponent and mantissa. Mutiplication of the two operands is done in three parts and thereby obtaining the Product. The first part of the product which is the sign is determined by an exclusive OR function of the two input signs. The exponent of the product which is the second part is calculated by adding the two input exponents. The third part which is the significand of the product is determined by multiplying the two input significands each with a '1'concatenated to it.

A multiplication of two floating-point numbers is done in four steps:

• non-signed multiplication of mantissas: it must take account of the integer part, implicit in normalization. The number of bits of the result is twice the size of the operands (48 bits).
• normalization of the result, the exponent can be modified accordingly .
• addition of the exponents, taking into account the bias.
• calculation of the sign.

III. FLOWCHARTS

In this section, flowcharts of the Adder, Subtractor and Multiplier is shown in Fig 3, 4 and 5.

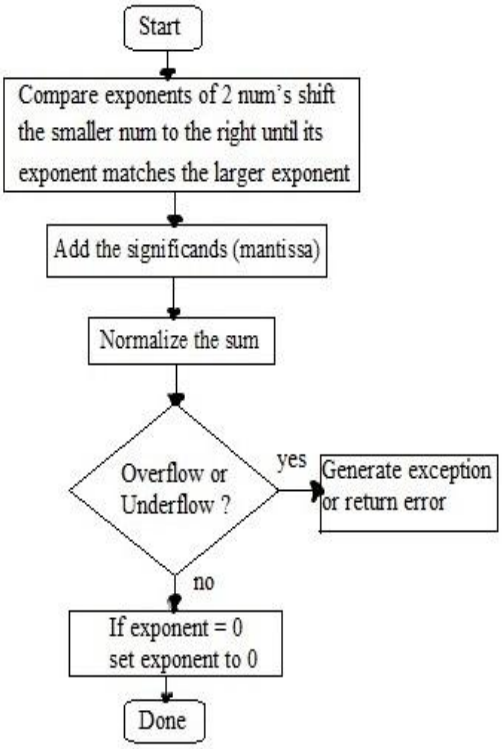


Fig 3.. Flow chart of Adder

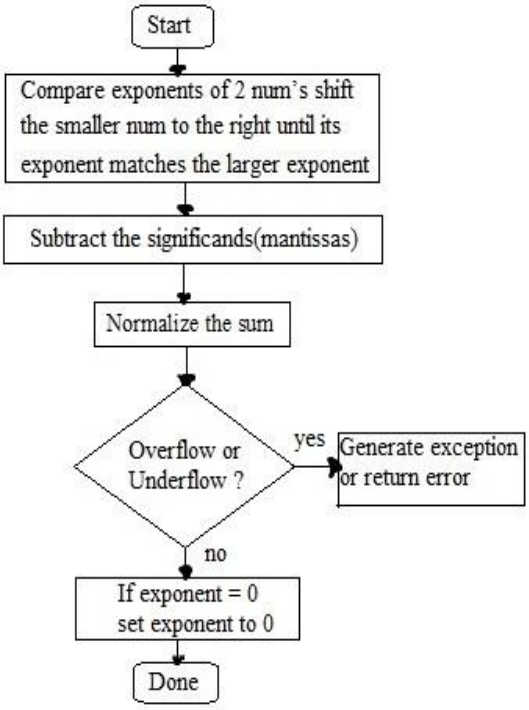


Fig 4. Flowchart of Subtractor

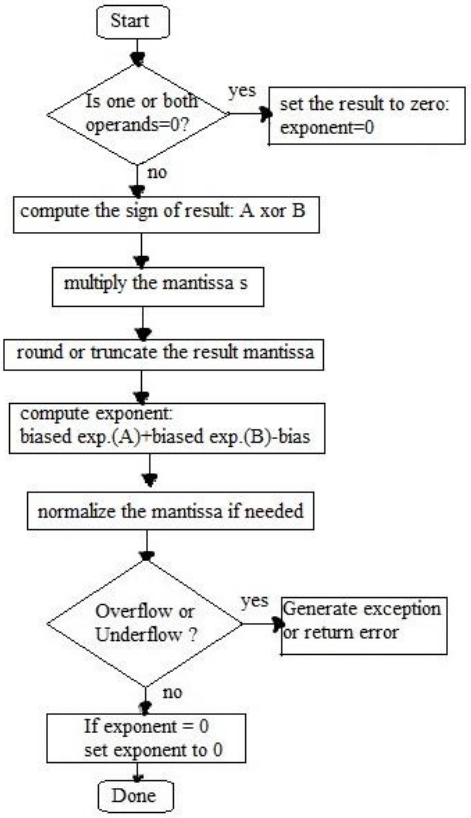


Fig 5. Flowchart of Multiplier

IV. SIMULATION RESULTS

Simulation is a process where the coded design is analyzed under a set of conditions. This is carried out by specifying the possible inputs to the design and verifying the outputs. Xilinx 12.2 tool and ModelSim simulator was used to simulate the design. We have applied the sample (floating point) inputs to each block to check the correctness of the design. The state of various entities in the process of execution of the code is also shown.

V. CONCLUSION

Floating-point representation is the most common solution. Basically represents reals in scientific notation. Scientific notation represents numbers as a base number and an exponent. Multiplication of Single Precision floating point numbers in IEEE 754 format is suitable for the 32 bit DMA Controller, 32 bit Memory and Memory Controller of the typical AMBA (Advanced Microcontroller Bus Architecture) AHB (Advanced High Performance Bus). The main blocks of the design are Floating point Complex multiplier, Floating point Complex adder, Floating point Complex

Subtractor. These three components of the design has been coded by. verilog HDL and the simulation has been done by using xilinx 12.2 tool.

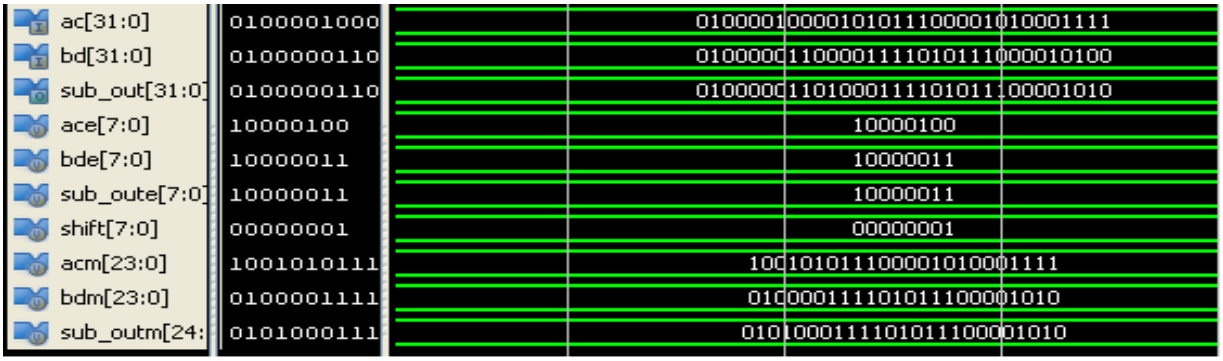


Fig .6 Timing diagram of floating point Subtractor

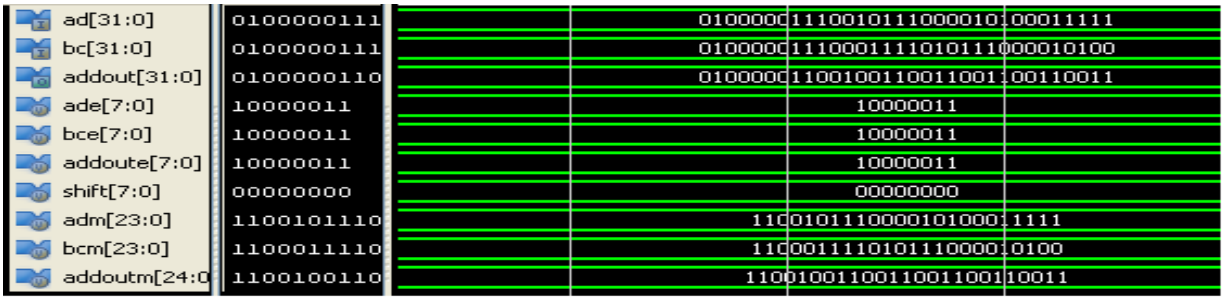


Fig.7 Timing diagram of floating point Adder

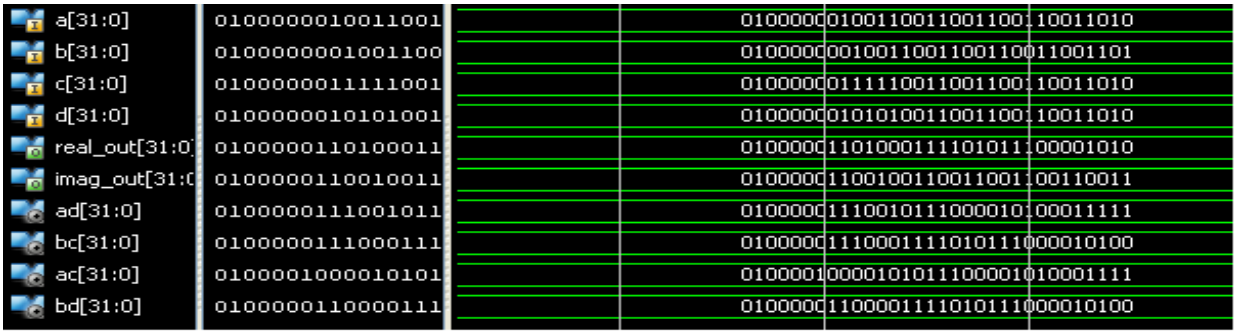


Fig.8 Timing diagram of floating point Top Module

VI. REFERENCES

[1] Louca L , Cook T A , Johnson W H , “Implementation of IEEE single precision floating point addition and multiplication on FPGAs ” 17-19 Apr 1996. Page(s) 107-116

[2] Chen D, Han L , Ko S B , “Decimal floating-point antilogarithmic converter based on selection by rounding: algorithm and architecture **“**Computers & Digital Techniques”, IET Volume: 6 , Issue: 5 Publication Year: 2012 , Page(s): 277 – 289” Devices, Circuits and Systems, 2004. Proceedings of the Fifth IEEE International Caracas Conference on 2004 , Page(s): 319 – 323

[3] Marcus G. Hinojosa P. Avila. Nolazco-FIores , “A fully synthesizable single precision,floating-point adder/ substractor and multiplier in VHDL for general and educational use

[4] Kanhe, A. ; Das, S.K. ; Singh, A.K. "Design and implementation of floating point multiplier based on Vedic Multiplication Technique" Communication,

Information & Computing Technology (ICCICT), 2012 International Conference on 2012 , Page(s): 1 – 4 .
[5] Ushasree, G. ; Dhanabal, R. ; Sahoo, S.K. "VLSI implementation of a high speed single precision floating point unit using verilog" Information & Communication Technologies (ICT), 2013 IEEE Conference on 2013 , Page(s): 803 – 808.
[6] David Goldberg, "What Every Computer Scientist Should Know About Floating-Point Arithmetic", published in the March, 1991 issue of Computing Surveys, Association for Computing Machinery, Inc.
[7] Cuyt , A., Verdonk, B., and Verschaeren, D. "A Precision- and Range-independent Tool for Testing Floating-point Arithmetic II: Conversions ACM Transactions on Mathematical Software, 27(1):119-140, May 2001.

AUTHOR'S PROFILE

Tejaswini H N
Asst. Professor
Dept. of Electronics and Communication, Sambhram Institute of Technology, Bangalore.

Dr. C V Ravishankar
Professor & HOD
Dept. of Electronics and Communication, Sambhram Institute of Technology, Bangalore.