International Journal of Engineering Research and General Science Volume 5, Issue 2, March-April, 2017
ISSN 2091-2730

A REVIEW ON A HIGH SPEED BINARY FLOATING POINT MULTIPLIER USING DADDA ALGRITHM IN FPGA

Miss Komal N.Batra [1]

[1]PG Student of HVPM'S College of Engineering and Technology Amravati (India)

Email: komal25690@rediffmail.com

Contact No- +918087744275

Prof. Ashish B.Kharate [2]

[2]Asst. Professor in Dept. of HVPM'S College of Engineering and Technology Amravati (India)

Abstract— Floating Point (FP) Multiplication is widely used in large set of scientific and signal processing computation. Multiplication is one of the common arithmetic operations in these computations. Most of the DSP applications need floating point numbers multiplication. The possible ways to represent real numbers in binary format floating point numbers are; the IEEE 754 standard represents two floating point formats, Binary interchange format and Decimal interchange format. To improve speed multiplication of mantissa is done using specific multiplier replacing Carry Save Multiplier. To give more precision, rounding is not implemented for mantissa multiplication. The binary floating point multiplier is to be implementing using VHDL and it is simulated and synthesized by using Modelism and Xilinx ISE software respectively.

Keywords—Single Precision and double precision, Dadda Multiplier, Floating point, VHDL, FPGA, Digital signal processing, IEEE Standard 754.

INTRODUCTION

Floating point numbers are one possible way of representing real numbers in binary format; the IEEE 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. Floating-point implementation on FPGAs has been the interest of many researchers. FPGAs are increasingly being used in the high performance and scientific computing community to implement floating-point based hardware accelerators. FPGAs are generally slower than their application specific integrated circuit (ASIC) counterparts, as they can't handle as complex a design, and draw more power. However, they have several advantages such as a shorter time to market, ability to re-program in the field to fix bugs, and lower nonrecurring engineering cost costs. Vendors can sell cheaper, less flexible versions of their FPGAs which cannot be modified after the design is committed. The Development of these designs is made on regular FPGAs and then migrated into a fixed version that more resembles an ASIC.

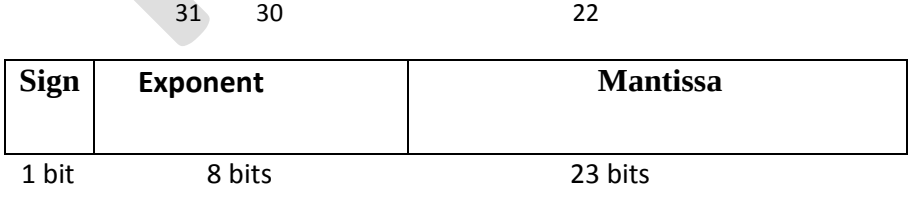


Fig1.Single Precision Floating point format

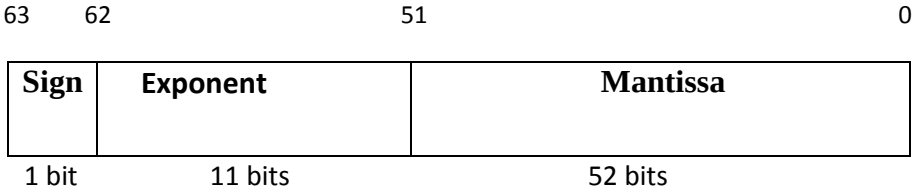


Fig2.Double Precision Floating point format

This paper presents a high speed binary floating point multiplier based on Dadda Algorithm. The coding is done for 32-bit single precision floating point multiplication using VHDL. Our method using Dadda method can have a great impaction improving the speed and reduce the area and power consumed by the Digital Signal Processors. The design achieves high speed with maximum frequency of 526 MHz compared to existing floating point multipliers. The floating point multiplier is developed to handle the underflow and overflow cases. To give more precision, rounding is not implemented for mantissa multiplication. The multiplier is implemented using VHDL. The multiplier is compared with Xilinx floating point multiplier core.

Double precision floating point numbers are 64-bit binary numbers. The 64-bits are divided into 3 parts- sign, exponent and mantissa. The 52 least significant bits (LSBs) are used to represent the mantissa of the number. The next 11-bits are used to represent the exponent of the number. The most significant bit (MSB) of the number is used as a sign bit to represent the sign of the number.

- Sign bit „0 "indicates positive number".
- Sign bit „1 "indicates negative number".

Most of the DSP applications need floating point numbers multiplication. The possible ways to represent real numbers in binary format floating point numbers are; the IEEE 754 standard represents two floating point formats, Binary interchange format and Decimal interchange format. Single precision normalized binary interchange format is implemented in this design. Representation of single precision binary format is shown in Figure 1; starting from MSB it has a one bit sign (S), an eight bit exponent (E), and a twenty three bit fraction (M or Mantissa). Adding an extra bit to the fraction to form and is defined as significand1. If the exponent is greater than 0 and smaller than 255, and there is 1 in the MSB of the significant then the number is said to be a normalized number.

LITERATURE REVIEW :

Various researches have been done to increase the performance on getting best and fast multiplication result on two floating point numbers. Some of which are listed below-

Addanki Puma Ramesh, A. V. N. Tilak, A.M.Prasad [1] the double precision floating point multiplier supports the LEEE-754 binary interchange format. The design achieved the increased operating frequency. The implemented design is verified with single precision floating point multiplier and Xilinx core, it provides high speed and supports double precision, which gives more accuracy compared to single precession. This design handles the overflow, underflow, and truncation rounding mode resp.

Itagi Mahi P and S. S. Kerur [2] ALU is one of the important components within a computer processor. It performs arithmetic functions like addition, subtraction, multiplication, division etc along with logical functions. Pipelining allows execution of multiple instructions simultaneously. Pipelined ALU gives better performance which will evaluated in terms of number of clock cycles required in performing each arithmetic operation. Floating point representation is based on IEEE standard 754. In this paper a pipelined Floating point Arithmetic unit has been designed to perform five arithmetic operations, addition, subtraction, multiplication, division and square root, on floating point numbers. IEEE 754 standard based floating point representation has been used. The unit has been coded in VHDL. The same arithmetic operations have also been simulated in Xilinx IP Core Generator.

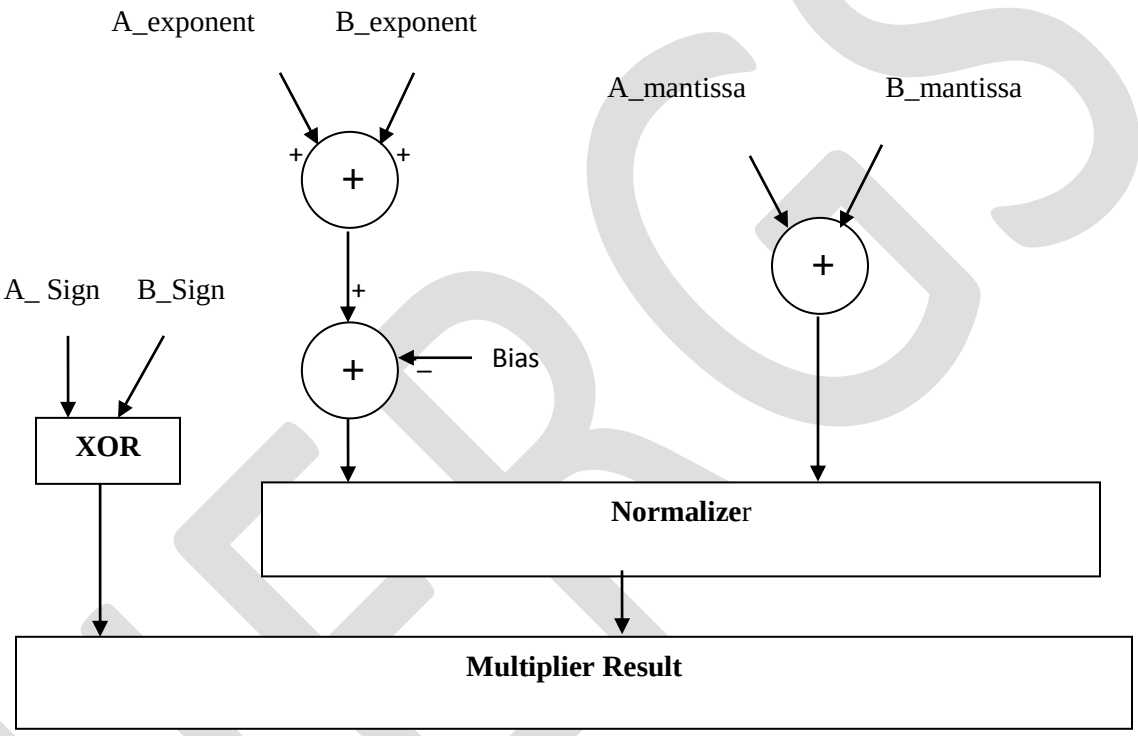
Remadevi R [3] Multiplying floating point numbers is a critical requirement for DSP applications involving large dynamic range. This paper presents design and simulation of a floating point multiplier that supports the IEEE 754-2008 binary interchange format, the proposed multiplier does not implement rounding and presents the significant multiplication result. It focuses only on single precision

normalized binary interchange format. It handles the overflow and underflow cases. Rounding is not implemented to give more precision when using the multiplier in a Multiply and Accumulate (MAC) unit.

Rakesh Babu, R. Saikiran and Sivanantham S [4] A method for fast floating point multiplication and the coding is done for 32-bit single precision floating point multiplication using Verilog and synthesized. A floating point multiplier is designed for the calculation of binary numbers represented in single precision IEEE format. In this implementation exceptions like infinity, zero, overflow are considered. In this implementation rounding methods like round to zero, round to positive infinity, round to negative infinity, round to even are considered. To analyse the working of our designed multiplier we designed a MAC unit and is tested. These results are compared with the previous work done by various authors.

PROPOSED WORK:

In this paper we implemented a double precision floating point multiplier with exceptions and rounding. Figure shows the multiplier structure that includes exponents addition, significant multiplication, and sign calculation.



The normalized floating point numbers have the form of

Z= (-1S) * 2 (E - *Bias*) * (1.M). The following algorithm is used to multiply two floating point numbers.

1. Significant multiplication; i.e. (1.M1*1.M2).
2. Placing the decimal point in the result.
3. Exponent's addition; i.e. (E1 + E2 - Bias).
4. Getting the sign; i.e. s1 xor s2.
5. Normalizing the result; i.e. obtaining 1 at the MSB of the results significant.
6. Rounding implementation.

7. Verifying for underflow/overflow occurrence.

Consider the following IEEE 754 single precision floating point numbers to perform the multiplication, but the number of mantissa bits is reduced for simplification. Here only 5 bits are considered while still considering one bit for normalized number.

Dadda proposed a sequence of matrix heights that are predetermined to give the minimum number of reduction stages. To reduce the N by N partial product matrix, dada multiplier develops a sequence of matrix heights that are found by working back from the final two-row matrix. In order to realize the minimum number of reduction stages, the height of each intermediate matrix is limited to the least integer that is no more than 1.5 times the height of its successor. The process of reduction for a dadda multiplier is developed using the following recursive algorithm.

1. Let d1=2 and dj+1 = [1.5*dj], where dj is the matrix height for the jth stage from the end. Find the smallest j such that at least one column of the original partial product matrix has more than dj bits.

2. In the jth stage from the end, employ (3, 2) and (2, 2) counter to obtain a reduced matrix with no more than dj bits in any column.

3. Let j = j-1 and repeat step 2 until a matrix with only two rows is generated.

ACKNOWLEDGMENT

We are very grateful to our College of HVPM College of Engineering and Technology to support and other faculty and associates of ENTC department who are directly & indirectly helped me for these paper.

CONCLUSION

The main focus of this paper is to introduce a method for calculating the multiplication of two floating point numbers with comparatively lesser time. It describes an implementation of a floating point multiplier that supports the IEEE 754- 2008 binary interchange format. The multiplier is more precise and it presents the significant best multiplication result. It may be used applications such as digital signal processors, general purpose processors and controllers and hardware accelerators etc. The design achieves high speed with maximum frequency of 526 MHz compared to existing floating point multipliers. The floating point multiplier is developed to handle the underflow and overflow cases. The implemented design is also efficient in terms of device utilization **.**The idea proposed here may set path for future research in this direction. Future scope of research this is to reduce area requirements and can be extended to various fields of DSP.

REFERENCES:

[1]. Addanki Puma Ramesh, A. V. N. Tilak, A.M.Prasad, "An FPGA Based High Speed IEEE-754 Double Precision Floating Point Multiplier using Verilog", 978-1-4673-5301-4/13/2013 IEEE.

[2]. Itagi Mahi P and S. S. Kerur, "Design and Simulation of Floating Point Pipelined ALU Using HDL and IP Core Generator", ISSN 2277 – 4106 ©2013 INPRESSCO.

[3]. Remadevi R, "Design and Simulation of Floating Point Multiplier Based on VHDL", Vol.3, Issue 2, March -April 2013.

[4]. A. Rakesh Babu, R. Saikiran and Sivanantham S, "Design of Floating Point Multiplier for Signal Processing Applications", ISSN 0973-4562 Volume 8, Number 6 (2013).

[5]. Gargi S. Rewatkar, "Implementation of Double Precision Floating Point Multiplier in VHDL", Volume.1, Issue 1, April 2014 (IJIREC).

[6]. P.Gayatri, P.Krishna Kumari, V.Vamsi Krishna, T.S.Trivedi, V.Nancharaiah, "Design of Floating Point Multiplier Using Vhdl", Volume 10, Issue 3 (March-2014), IEEE.

[7]. W. Kahan "IEEE Standard 754 for Binary Floating-Point Arithmetic,"1996

[8]. Michael L. Overton, "Numerical Computing with IEEE Floating Point Arithmetic," Published by Society for Industrial and Applied Mathematics,2001.

[9]. D. Narasimban, D. Fernandes, V. K. Raj , J. Dorenbosch , M. Bowden, V. S. Kapoor, "A 100 Mhz FPGA based floating point adder",Proceedings of IEEE custom integrated circuits conference,1993.

[10]. Jim Hoff; "A Full Custom High Speed Floating Point Adder" Fermi National Accelerator Lab, 1992.

[11]. Subhash Kumar Sharma,Himanshu Pandey,Shailendra Sahni ,Vishal Kumar Srivastava, "Implementation of IEEE_754 Addition and Subtraction for Floating Point Arithmetic Logic Unit", Proceedings of International Transactions in Material Sciences and Computer,pp.131-140,vol.3,No.1,2010.

[12]. Shaifali, Sakshi, " FPGA Design of Pipelined 32-bit Floating Point Multiplier", International Journal of Computational Engineering & Management, Vol. 16, 5th September 2013.