Implementation of High Speed Single Precision Floating Point Multiplier

Reshmashree.M.S[1] Meenakshi L Rathod[2]
[1]M. Tech Scholar [2]Assistant Professor
[1,2]DR.AIT, Bengaluru

Abstract— This paper presents design of a high speed multiplier for IEEE754 single precision floating point numbers. In all microprocessors, industrial area and applications involving arithmetic operation, it is required that the operations are carried out at a faster rate. Thus, to increase the speed of the multiplication operation, Dadda algorithm is used. The basic modules are designed using Verilog language and implemented using Xilinx 12.4 ISE software and the same is simulated using isim simulator. The Dadda multiplier is designed using Carry lookahead adder and the results are verified to achieve higher speed criteria.

Key words: IEEE754, Xilinx 12.4, Algorithm

I. INTRODUCTION

Multiplication is the arithmetic operation which is used widely in signal processing operations. The speed of multiplication can be increased by reducing the number of steps in partial product reduction stage . This is achieved by using Dadda algorithm which reduces the number of stages of reduction and thus increasing the speed. Also, the speed of multiplier can be increased by using faster adders. Thus, in the final addition stage, ripple carry adder is replaced by carry lookahead adder. Floating point multiplication involves multiplying two numbers that are in IEEE 754 format. There are two types of floating point numbers namely, single precision (32 bits) and double precision (64 bits) floating point numbers. In this paper, we have considered single precision floating point numbers. Single precision number has 32 bits comprising of three parts namely, sign, exponent and mantissa. Sign bit is 1 bit and is '0' for non-negative numbers and '1' for negative numbers. Exponent is 8 bit wide and mantissa is 23 bit wide. This can be represented as shown.

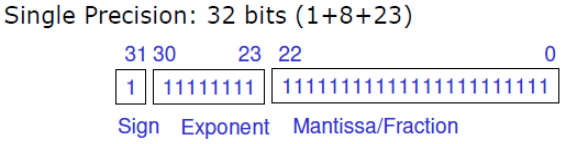

Fig. 1: Representation of Single Precision Floating Point Number

II. FLOATING POINT MULTIPLIER OPERATION

There are many algorithm and techniques that are used to design floating point multiplier. The main aim is to obtain high speed, less delay, minimize area, minimize power and obtain better performance. The important steps involved in multiplying two floating point numbers are as follow:

(1) Multiplying the significand of two numbers i.e., mantissa of two numbers.
(2) Placing the decimal point after multiplication in the result.
(3) Adding the exponents of two numbers and
   – Adding the bias if exponents are in binary format (E1+E2+bias).
   – Subtract the bias if exponent are in floating point format (E1+E2-bias).
(4) Obtaining the sign of final result i.e., performing exor operation on both sign bits.
(5) Normalizing the result i.e., obtaining 1 at the MSB of significand multiplication's result.

III. MAIN BLOCKS OF FLOATING POINT MULTIPLIER

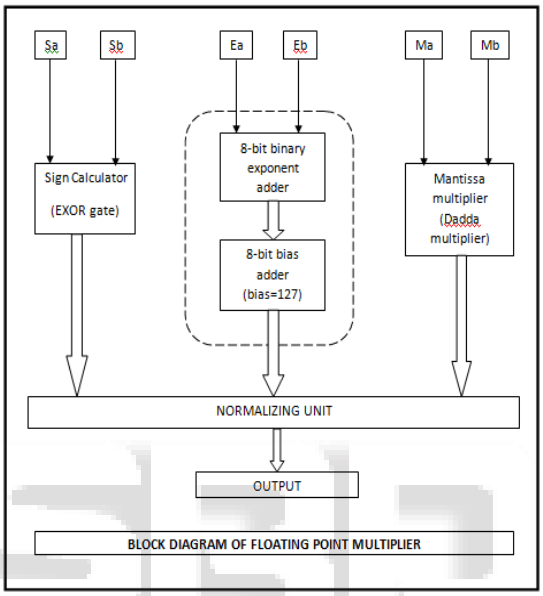

Fig. 2: Block Diagram of Floating Point Multiplier

The block diagram of floating point multiplier is as shown above. The main sub-blocks includes sign calculator, 8 bit binary exponent adder, 8 bit bias adder, mantissa multiplier and normalizing unit.

(1) Sign calculator: This block is used to find out the sign of result. The sign of two numbers can be calculated by EXORing the sign bits of two numbers such that the result is positive if both the sign bits are positive or negative and is negative if either of the sign bits are negative.
(2) Exponent calculator: This block is used to find the exponent of the result. The exponents of two numbers in binary format are added together and bias is added to the result to convert it to floating point number format. The bias for single precision floating point number is 127. For example, if the exponents are 3 and 4 i.e., 00000011 and 00000100 respectively, then the exponent is obtained by adding (00000011+00000100+01111111=10000110).
(3) Mantissa multiplier: This block is used to find the multiplication of two 23 bit mantissa. The multiplication is done using Dadda algorithm to obtain the final product. Dadda algorithm is the fastest way of multiplying two numbers with reduced number of stages for partial product

reduction. The dot matrix representation for 8x8 Dadda algorithm is as shown below.

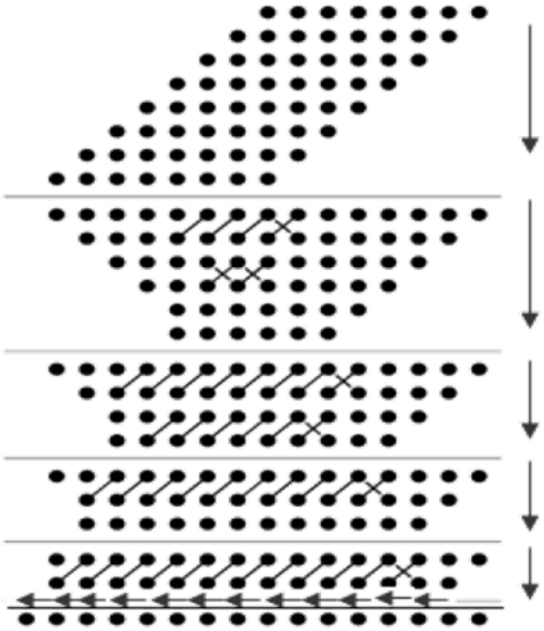

Fig. 3: Dot Diagram for 8x8 Dadda Multiplier

(4) Normalizing unit: Normalized result means they have a leading '1'to the left hand side of mantissa multiplication result. Depending upon the position of the decimal point the exponent is adjusted. Leading '1' is neglected and the remaining bits are reduced to 23 bits which forms the mantissa of final result.

IV. SCHEMATIC OF FLOATING POINT MULTIPLIER

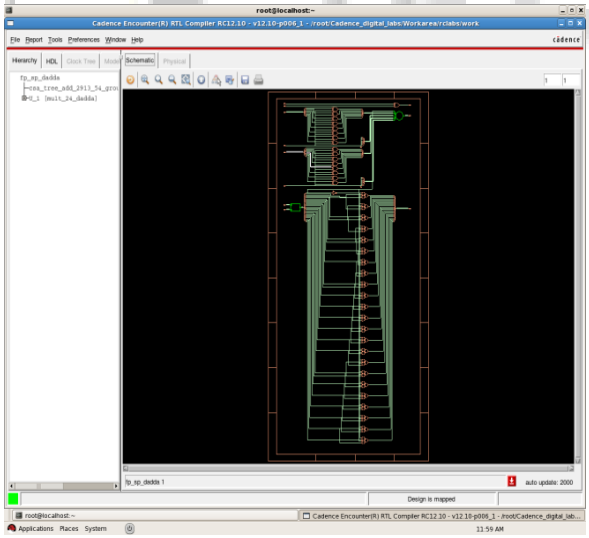

Fig. 4: Schematic of Floating Point Multiplier

V. SIMULATION RESULTS OBTAINED IN XILINX 12.4 ISE SOFTWARE

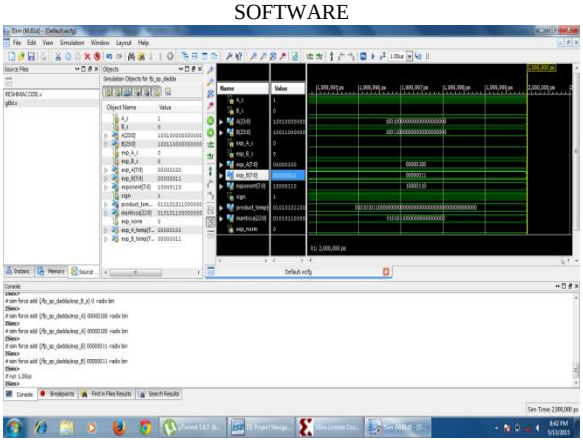

Fig. 5: Simultion Results Obtained in Xilinx 12.4 ISE Software

VI. CONCLUSION AND FUTURE WORK

This paper presents a high speed floating point multiplier that is compatible with IEEE 754 single precision binary floating point number format. For reducing delay, Dadda algorithm is used and also faster adder i.e., carry lookahead adder is used. The speed of multiplier increases but the area is also slightly increased. So the future plan will include optimizing the circuit such that the area criteria is reduced.

REFERENCE

[1] Supriya Sarkar, Sanghita Deb, G. Dilip, "A New Architecture for Signed and Unsigned Multiplier by Using Radix-4 Process", International Journal of Emerging Science and Engineering ISSN:2319-6378, Volume-2, Issue-1,November 2013.

[2] Brian Hickmann, Andrew Krioukov and Michael Schulte, "A Parallel IEEE P754 Decimal Floating-Point Multiplier", IEEE-2007.

[3] Mohamed Al-Ashrafy, Ashraf Salem, Wagdy Anis, :An Efficient Implementation of Floating Point Multiplier", IEEE-2011.

[4] Vinod Buddhe, Prasanna Palsodkar, Prachi Palsodakar, "Design and Verification of Dadda Algorithm based Binary Floating point Multiplier", International Conference on Communication and Signal Processing, April 3-5, 2014,India.

[5] T.Arunachalam and S.Kirubaveni, "Analysis of High Speed Multipliers", International Conference on Communication and Signal Processing, April 3-5, 2013, India.

[6] V.M.Dhivya, A.Sridevi,A.Ahilan, "A High Speed Area Efficient FIR Filter Using Floating Point Dadda Algorithm", International Conference on Communication and Signal Processing, April 3-5,2014, India.

