Design of High Speed Area Efficient IEEE754 Floating Point Multiplier

Mownika V.
Department of Electronics and Communication Engineering
Student, Narayana Engineering College, Nellore, Andhra Pradesh, India*

Girish Gandhi S.
Asst. professor, Department of Electronics and Communication Engineering
Narayana Engineering College, Nellore, Andhra Pradesh, India

Leela Mohan C.
Assoc. Prof., Department of Electronics and Communication Engineering
Narayana Engineering College, Nellore, Andhra Pradesh, India

Abstract- Basically multipliers are the functional blocks and core of the all digital signal processors (DSPs). The speed of multipliers is used to determine the speed of the DSPs. This paper deals with design of High speed Area efficient IEEE 754 double precision Floating Point Multiplier using Vedic mathematics. The mantissa is designed using the Urdhva-Tiryakbhyam sutra of Vedic Mathematics replacing the carry save multiplier. The exponent is determined using ripple carry adder. Design implementation targets signed multiplication. The proposed architecture is designed, simulated and synthesized using Xilinx tools.

Keywords – Urdhva - Tiryakbhyam Sutra, Vedic mathematics, Xilimx.

I. INTRODUCTION

The key components in many high performance systems like microprocessors, DSP processors, various FIR filters, etc are its multipliers. The overall performance of a system is determined by using the overall performance of the multiplier because the slowest element in the system is its multiplier. So the speed of the multipliers should be increased along with optimizing area is a major issue in multiplier design. The multipliers with different area-speed has been designed with fully parallel Multipliers at one quit of the spectrum and completely serial multipliers at the alternative end. The multiplication is one of the basic function in DSP applications binary floating point numbers. The Double precision floating point format consists of 64 bits.

The binary formats of a real numbers are known as floating point numbers. Floating point multiplication is much more different to that of normal integer multiplication. In DSP applications, the floating point formats are very important. This paper describes the double precision normalized binary interchange format. The double precision number format representation is shown in Figure 1.

The most significant bit 63 is the sign bit *(S),* The following 11 bits that is 52 to62 represent exponent *(E),* The remaining 52 bits represents fraction/ mantissa *(F).*

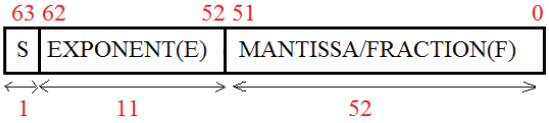


Figure 1. Double Precision Floating Point Format.

A number that consist of 'one' in the MSB of the significant and exponent is greater than zero and smaller than 1023 is said to be a normalized number. The following equations (I) & (2) represents real number.

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

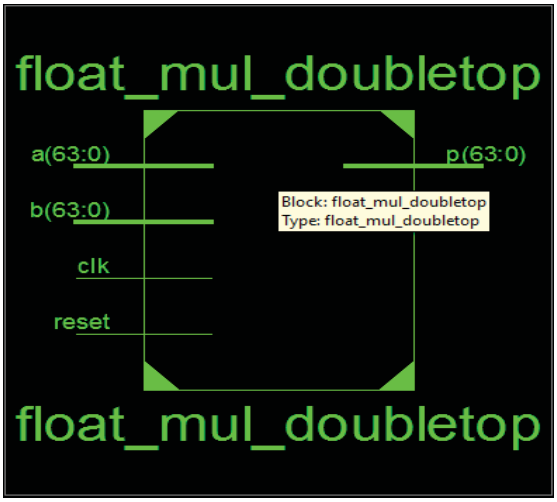
$$\text{Value} = (-1^{\text{Sign bit}})*2^{(\text{Exponent} -1023)}* (1.\text{Mantissa}) \quad (2)$$

This paper explains the "Design of high speed area efficient IEEE 754 floating multiplier" using Urdhva-Tiryakbhyam sutra of Vedic mathematics[5]. The IEEE 754 Floating point multiplier is implemented with the use of Vedic multiplication technique[1]. The mantissa multiplication is done by using the Vedic sutra[2].

II. PROPOSED ALGORITHM

A. Floating point multiplication algorithm

The double precision format facilitates to overcome the issues of single precision floating point. The implementation of Double Precision Floating Point multiplier[3] consist of four parts sign calculator, exponent calculator, mantissa calculator, and normalizer unit.



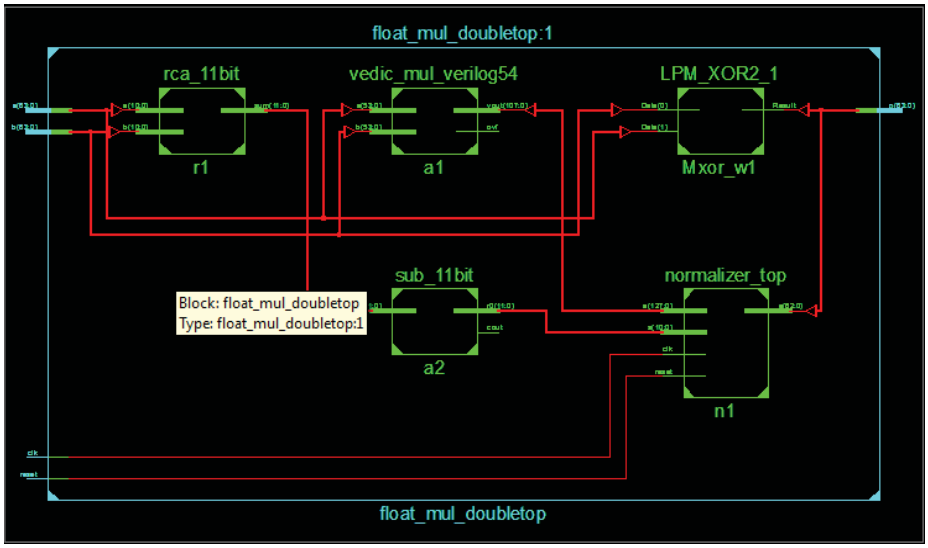


Figure 2. Architecture of double precision Floating point multiplier

Multiplication of two floating point numbers format is

1. Calculation of sign bit by XORing the sign of the two numbers(S1 xor S2).

2. Adding the exponent of two numbers and then subtracting the bias from the result (E1+E2-Bias).

3. Multiplication of significant of two numbers (1.M1*1.M2).

4. Normalizing the result (1 at MSB of result significant).

5. Rounding the result.

B. Multiplier design

The multiplier unit is based on an Urdhva Tiryakbhyam (vertical and crosswise) algorithm of ancient Indian Vedic mathematics[4]. Vedic mathematics is the ancient Indian system of mathematics. It was rediscovered in early twentieth century. Vedic mathematics is mainly based on sixteen principles.

Urdhva - Tiryakbhyam algorithm is applicable to all types of multiplications[6]. The key advantage is that it decreases the need of microprocessors to operate with high frequencies.

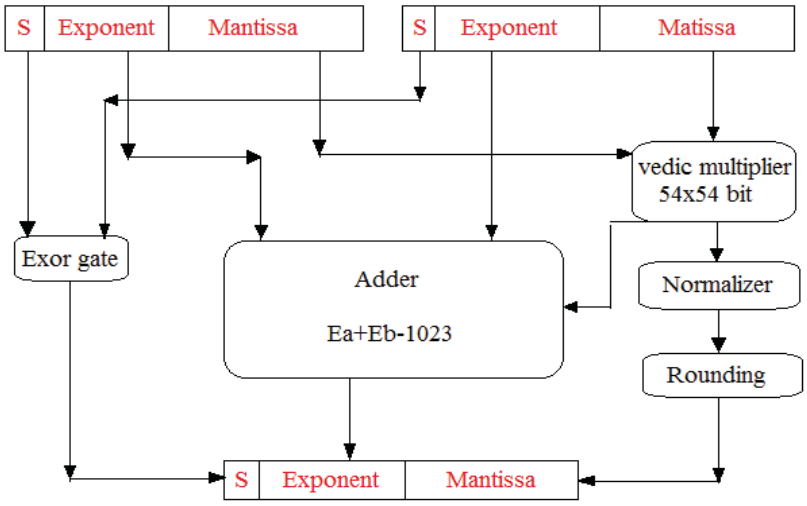


Figure 3. Block diagram of floating point multiplier

In this approach the partial products are generated simultaneously hence it decreases delay and makes this method fast.

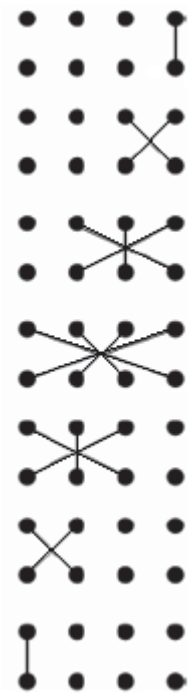


Figure 4. Line diagram of the multiplication

C. Mantissa calculation unit

The Mantissa calculation unit performance dominates the overall performance of the Floating point multiplier. In this unit it performs the multiplication of 52*52 Bits. In order to perform this multiplication the Vedic Multiplication technique[4] is choose. This technique results are in the form of speed and power. For the design of 54 bit multiplier it requires four Vedic 27 bit multiplier units.

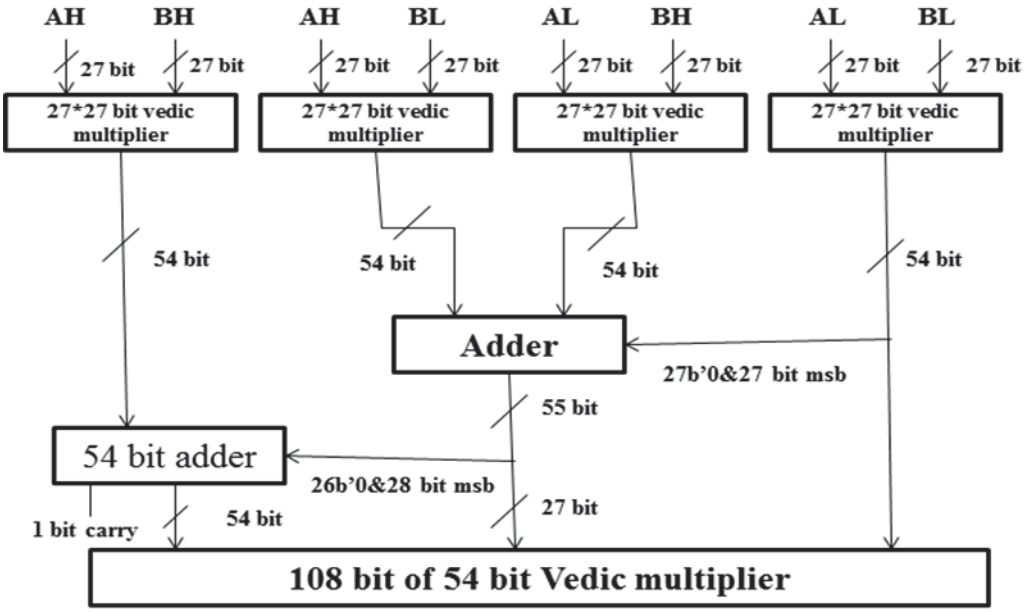


Figure 5. 54*54 bit multiplier block

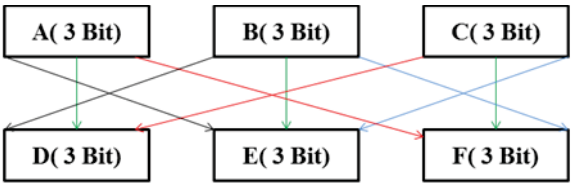


Figure 6. 9*9 bit Vedic multiplier

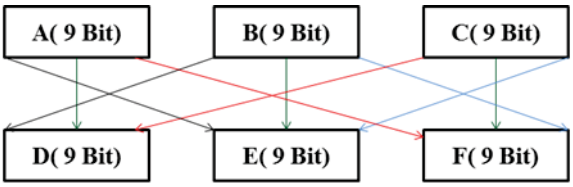


Figure 7. 27*27 bit Vedic multiplier

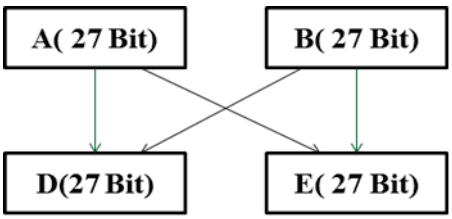


Figure 8. 54*54 bit Vedic multiplier

D. Exponent unit

In the Exponent Unit the exponent bits 53 to 63 (11 bits) are added. Then the resultant ER is subtracted with 2's complement of bias value1023 for 54 bit multiplier. The exponent of 11bit is 2^11(1023). The two numbers E1 and E2 are shifted by the bias value and it is not a real exponent value i.e.

$$EA = EA - true + bias \text{ and } EB = EB - true + bias.$$

Hence

$$ER = E1 + E2 - 1023.$$

E. Normaliser unit

A number is said to be a normalized number then it is having a leading '1' just immediate to the left of the decimal point(i.e. 106 bit of resultant significant multiplication). If the one is at bit 106 (i.e. to the left of a decimal point) the there is no need of the intermediate product and is known to be a normalized number. The product is shifted to the right and exponent is incremented by 1 if the leading one is at bit 107. Rounding of a number is also performed by the Normalizer, because floating point arithmetic operations compute result that cannot be represented in a specified amount of precision.

III. SIMULATION RESULT

A=12.5
B=12.5
P=156.25

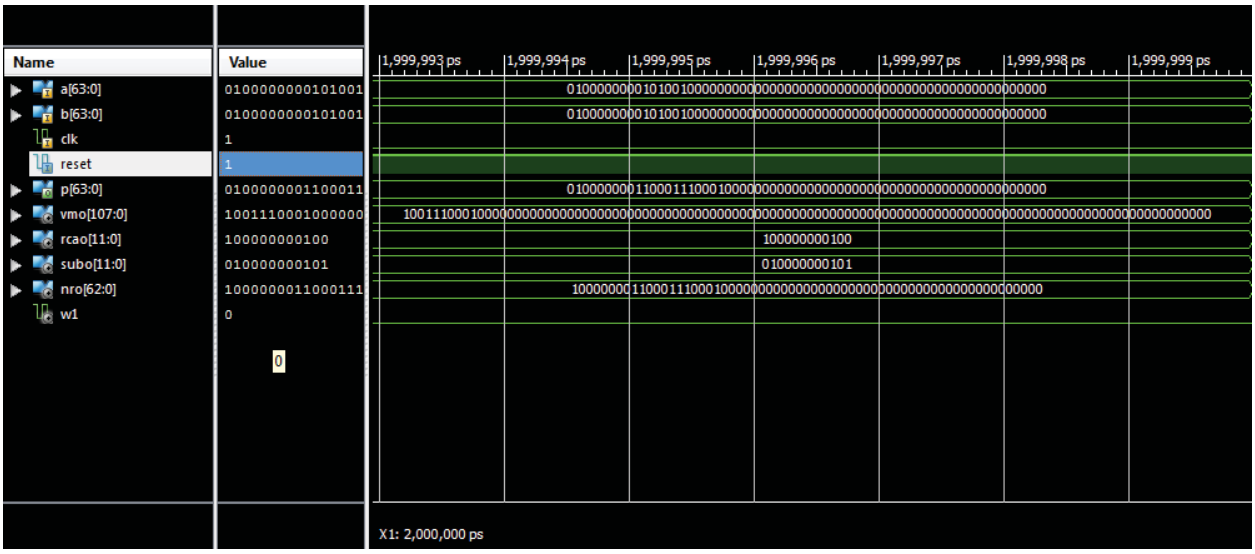


TABLE 1: DESIGN SUMMARY

PARAMETER	EXISTING	PROPOSED
Delay (ns)	152.587	132.887
No. of LUTs	7157	7097
No .of I/o s	214	194

IV. RESULT AND CONCLUSION

This paper presents an efficient double precision floating point multiplier design. High speed can be achieved by using the Vedic multiplier. The design algorithm and results show that this Vedic multiplier requires moderate area and high speed as compared to the conventional multipliers. This multiplier can be used in future in digital signal processing or VLSI signal processing applications such as FFT, IFFT, complex floating multiplier.

REFERENCES

[1] Sushma S. Mahakalkar, Sanjay L. Haridas, ”Design of High Performance IEEE754 Floating Point Multiplier Using Vedic mathematics”, 2014 Sixth International Conference on Computational Intelligence and Communication Networks.

[2] Aniruddha Kanhe, Shishir Kumar Das, Ankit Kumar Singh,” Design and Implementation of Floating Point Multiplier based on Vedic Multiplication Technique” 2012 International Conference on Communication, Information & Computing Technology (ICCICT), Oct. 19-20, Mumbai, India.

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

[4] Honey Durga Tiwari, Ganzorig Gankhuyag, Chan Mo Kim, Yong Beom Cho, "Multiplier design based on ancient Indian Vedic Mathematics”, 2008 International SoC Design Conference, PP 65-68.

[5] Jagadguru Swami Sri Bharati Krisna Tirthaji Maharaja, “Vedic Mathematics Sixteen Simple Mathematical Formulae from the Veda,”1965.

[6] Ramesh pushpangadhan, Vineeth sukumaran, Rino innocent, Dinesh sasikumar, Vaisak sundar, “High speed Vedic multiplier for digital signal processing.