International Journal of Scientific & Engineering Research, Volume 6, Issue 4, April-2015 914
ISSN 2229-5518

Design And Implementation Of An Efficient Single Precision Floating Multiplier Using Vedic Multiplication

Bhavesh Sharma , Amit Bakshi

Abstract— This paper contains design of a single precision floating point multiplier by modifying the proposed architecture[6] and then comparing the different floating point multiplier architecture for the various performance parameters. The designs are modeled in Verilog HDL and synthesized based on the TSMC 180nm standard cell library. Comparisons are based on the synthesis result obtained by synthesizing all the multiplier using Cadence Encounter RTL Compiler .

Index Terms— Floating Point , Floating Point Multiplier, Vedic Multiplication .

---

1 Introduction

Multipliers have an important part of the modern electronic years. Floating point multiplier's can be found in electronic systems that run complex calculations especially in DSP processor. In parallel multipliers the number partial product addition required determines the performance for the multipliers.

A study in [1] showed that array multiplier has the largest area and delay when compared to other multiplier design , while the Wallace tree multiplier has less area and delay compared to the array multiplier . In [2] the study stated that depending on the application either booth multiplier and wallace multiplier can be used since Booth Multiplier uses less area compared to wallace tree multiplier but wallace tree multipliers dissipates less power . In study [3] stated that in Vedic Multiplier when ripple carry adder is used the area required is less compared to the use of carry save adder .

2 Floating Point Multiplier

The floating point multiplier represented in IEEE 754 format is divided in four unit.

i. Multiplier Unit
ii. Exponent Calculation Unit
iii. Sign Calculation Unit
iv. Control Unit

The Mantissa of the resultant is calculated using a 24X24 bit multiplier. The control unit is used to raise overflow and under flow conditions. The exponent calculation is done by four 8 bit ripple carry adder in the architecture[6].

In the exponent bits are biased to 127 that

$$EA=Ea+127 \quad (1)$$

$$EB=Eb+127 \quad (2)$$

That final resultant is obtained by performing by

$$Er=Ea+Eb \quad (3)$$

Now Er is again biased to 127 .

$$ER=Er+1$$

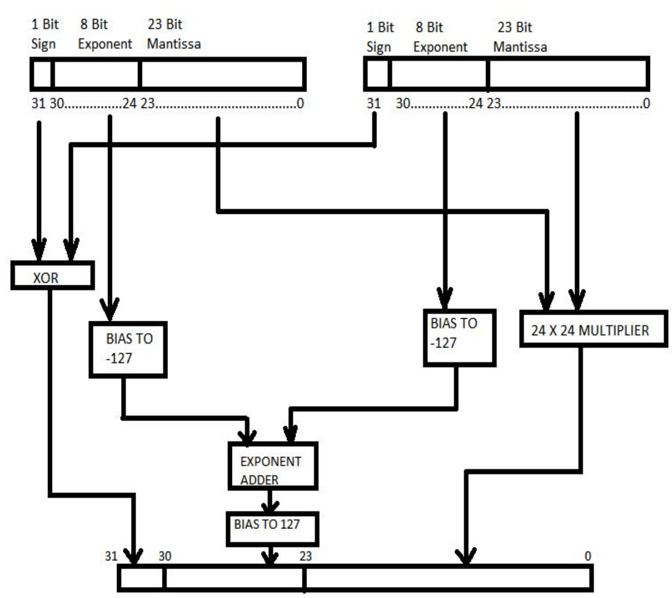

Fig. 1. Architecture For Floating Point Multiplier[6].

• *Bhavesh Sharma is currently pursuing dual degree program in electronics and telecommunication engineering in KIIT University, India PH-+919778201645. E-mail: bhavesh13121990@gmail.com*
• *Amit Bakshi is the guide and is currently an Assistant Professor KIIT University, India , E-mail: abakshi.ece@gmail.com*

ISSN 2229-5518

3 MODIFIED FLOATING POINT MULTIPLIER

The basic working of the modified architecture this same as the proposed architecture[6] with the only difference in the exponent unit. The Mantissa of the resultant is calculated using a 24X24 bit multiplier . The control unit is used to raise overflow and underflow conditions. The Exponent calculation is done by using single 8 bit carry save adder . The two 127 biased exponents are added along with the 2's complement of 127 which gives the resultant exponent. The 2's complement of 127 is added because we want the resultant output to be biased to 127. That is if ER is the final result then

$$ER = EA + EB - 127 \quad (5)$$

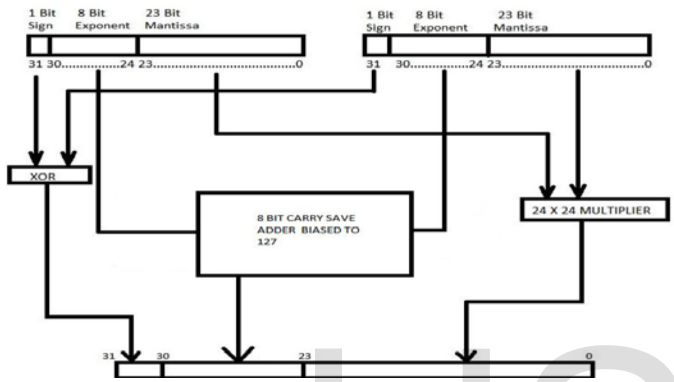

Fig. 2. Modified Architecture For Floating Point Multiplier.

4 FLOATING POINT MULTIPLICATION

The Fig 1 shows the architecture for the floating point multiplier . Consider the multiplication of two floating point numbers A and B , where A = -131.25 and B = -7.75 The IEEE representation of the two number are

	SIGN	EXPONENT	MANTISSA
A =	1	10000110	00000110100000000000000
B =	1	10000001	11110000000000000000000

Sign bit of the resultant are obtained by XORing the sign bit of the operands. In this case the sign bit of the resultant is 0. Exponent of the resultant are obtained by first un biasing the exponents of the operand, then adding them and then finally biasing it to 127 . This operation requires four 8 bit ripple carry adders .

The Mantissa is expressed in 23 bits , so they are normalized by placing a 1 at the MSB . The two normalized mantissa are:

100000110100000000000000

111110000000000000000000

The resultant of the multiplication is a 48 bit number which is then again normalized to 23 bits by eliminating the most significant 1.

We obtain the resultant as :

0 10001000 11111100100110000000000

This resultant is converted to 1017.1875

5 VEDIC MULTIPLIER

Vedic multiplier is based on the vedic multiplication sutra. These sutras are used for the multiplication of two numbers in decimal system. The multiplier is based on Urdhava Triyakbhayam Sutra . In this concept the generation of partial product can be done and then parallel addition of these partial product is done .

For an example let us take a 3X3 multiplier which is shown in Fig. 3. Consider the numbers X and Y where

A = X1X2X3 and Y= Y1Y2Y3.

The operations are :

Su0=X3Y3 (6)

Ca1Su1 = X3Y2 + X2Y3 (7)

Ca2Su2=Ca1+X1Y3+X2Y2 + X3Y1 (8)

Ca3Su3=Ca2+X1Y2+X2Y1 (9)

Ca4Su4=Ca3+X1Y1 (10)

Now the final result of multiplication of X and Y is Ca4Su4Su3Su2Su1Su0. (11)

The 24X24 bit multiplier is design by using four 12X12 bit multiplier . The 12X12 Multiplier is again designed using four 6X6 multiplier . The 6X6 Multiplier is again designed using four 3X3 multiplier. So the multiplier uses hierarchical structure to reduce the number of partial product generation .

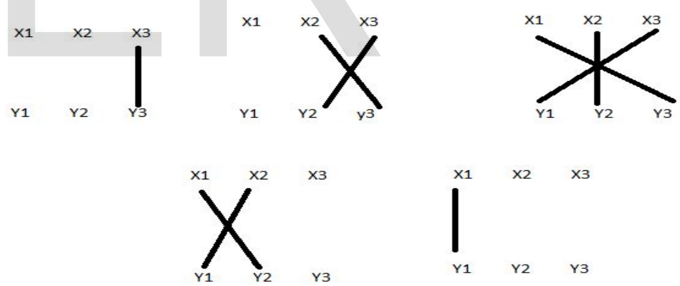

Fig. 3. Vedic Multiplication Technique

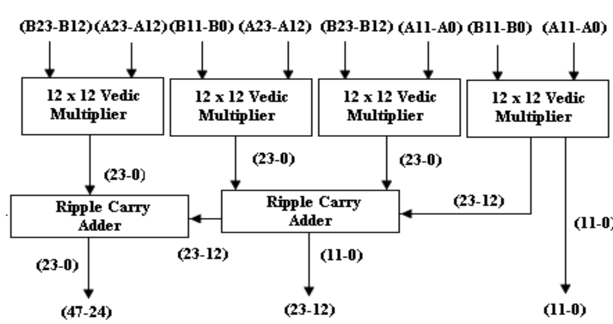

Fig. 4. 24X24 Bit Vedic Multiplier.

ISSN 2229-5518

6 Result And Discussion

In [7] the proposed architecture was used design a floating point multiplier using different multiplier algorithms for the 24X24 multiplier. Upon comparison it was concluded that floating point multiplier with vedic multiplier has the least area and power delay product . The same vedic multiplier is therefore used for comparison in the modified architecture.

TABLE 1
Performance Comparison Of The Two Architecture

Parameter	Floating point multiplier [7]	Modified Floating point Multiplier
Delay(ns)	8.0	9.2
Area	71946	65109
Power Dissipation (mw)	20.305	15.024
A.D(10^{-3})	0.575	0.599
P.D(10^{-12})	162.44	138.22
Power density(10^{-7})	2.822	2.308

6.1 Delay Comparison

Floating point multiplier with modified architecture has a larger delay compared to the architecture used in[7]. The increase in delay is 15%.

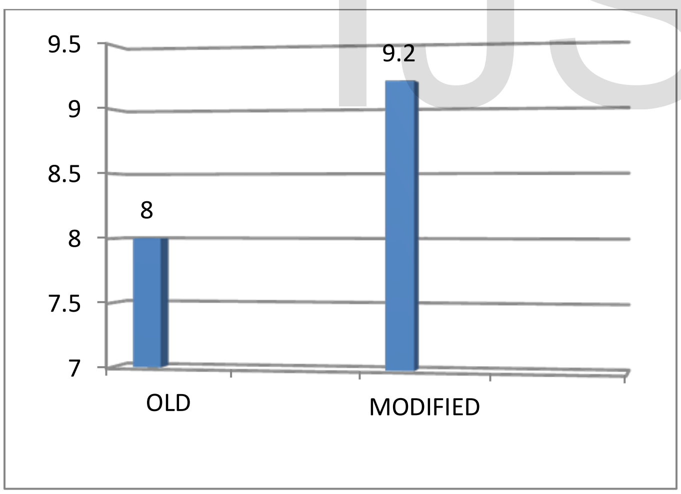

Fig. 5. Graph For Delay Comparison.

6.2 Area Comparison

Floating point multiplier with old architecture[6] has a larger area compared to the modified architecture proposed because of the additional adders used. Since in the modified architecture a single 8 Bit carry save adder is used due to which area is reduced by 9.5%.

6.3 Power Comparison

Due to the reduction in area the power of the modified architecture is reduced by 26%.

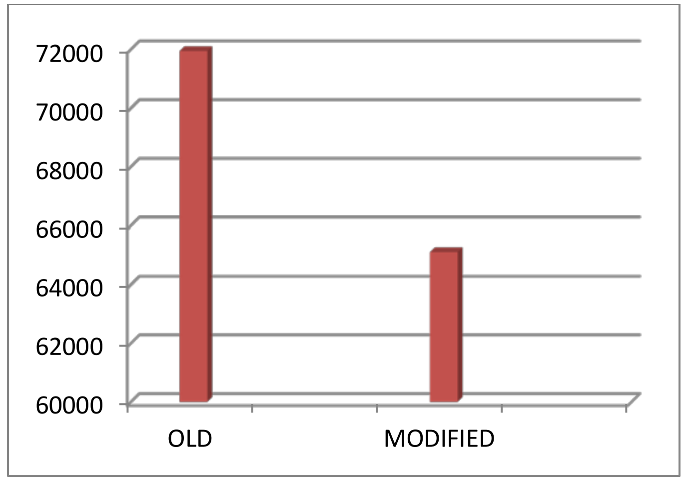

Fig. 6. Graph For Area Comparison.

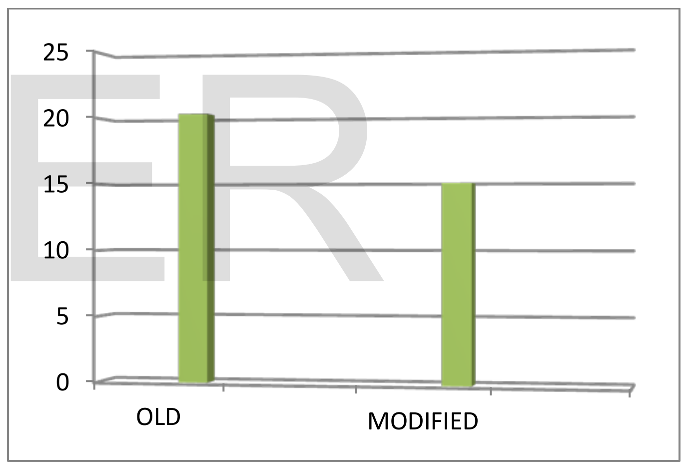

Fig. 7. Graph For Power Comparison.

7 Conclusion

Based on the experimental data gathered as seen in table 1, the modified architecture is better in aspect of area , power , power delay product and power density . Although with a slight increase in delay we achieve a low power amd low area architecture. This is a known trade off between powe and speed or delay which states that as the speed increases delay decreases.

In the modified architecture the power delay product is reduced by 14.91% and the power density is reduced by 18.21% . With so many performance parameters showing experimentally that the new architecture is better compared to the architecture in [6].

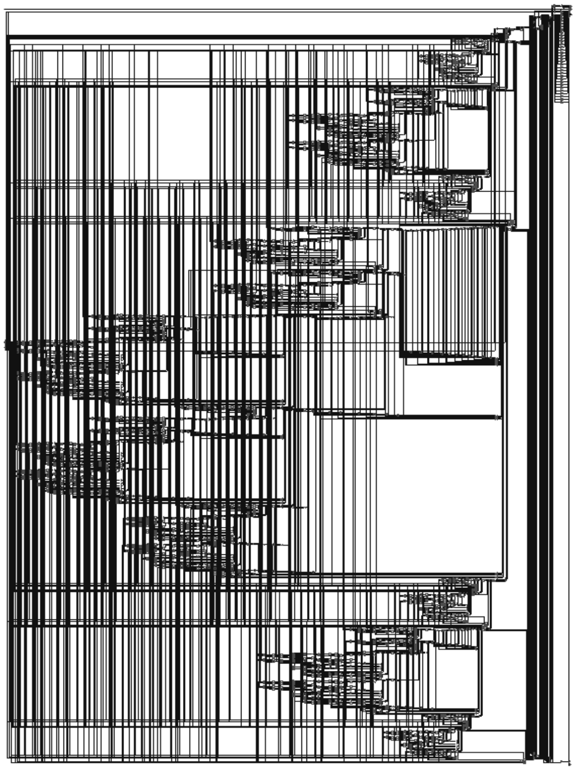

Fig. 8. RTL Schematic Of Modified Architecture.

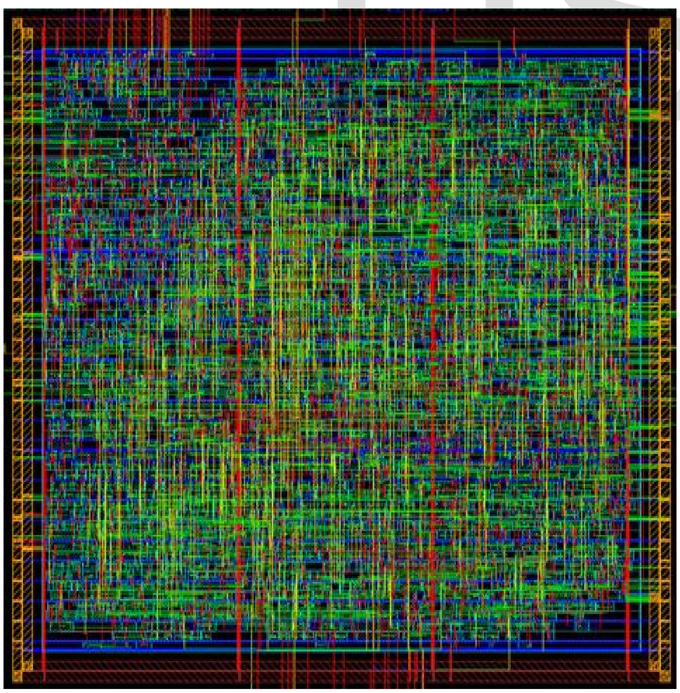

Fig. 9. Layout Of Modified Architecture.

REFERENCES

[1] Chris Y.H. Lee, Lo Hai Hiung, Sean W.F. Lee Nor Hisham Hamid, Universitiy Teknologies PETRONAS and Emerald Systems Sdn. Bhd., "A Performance Comparison Study on Multiplier Designs".

[2] S.Shah, A.J.AKhabb, D.AI Khabb "Comparison of 32-bit Multipliers for Various Performance Measures" The 12th International Conference on Microelectronics , Tehran 2000.

[3] Shikha Kaushik , Javed Ashraf "Implementation of Vedic Multiplier using Different Architecture " International Journal for Research in Science & Advanced Technologies Issue-2,Volume-2,062- 066.

[4] Vadiraj Sagar, Shripad Sagar, Sudhindracharya, Vedavyas Mathad, Subhash Kulkarni "Vhdl Implementation of Vedic Mathematical Sutras" Department of Electronics & Communication, PDA College of Engineering .

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

[6] Aniruddha Kanhe, Shishir Kumar Das, Ankit Kumar Singh"Design and Implementation of Floating Point Multiplier based on Vedic Multiplication Technique."ICCICT 2012.

[7] Bhavesh Sharma, Ruchika Mishra ,Nilesh Didwania , AritraMitra , Amit Bakshi "Comparison Of Single Precision Floating Point Multiplier Using Different Multiplier Algorithms" ICIEEE April 2015 , PUNE, INDIA.

[8] Bhavesh Sharma , Amit Bakshi "Comparison Of 24X24 Bit Multiplier For Various Performance Parameter" ICATEST 2015.

[9] IEEE 754-2008, IEEE Standard for Floating-Point Arithmetic, 2008.