ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Indian Journal of Science and Technology, Vol 8(25), DOI: 10.17485/ijst/2015/v8i25/80236, October 2015

An Efficient Multi-Precision Floating Point Adder and Multiplier

Budda Nagendra Reddy* and P. Augusta Sophy Beulet

Department of School of Electronics Engineering, VIT University, Chennai - 632014, Tamil Nadu, India;
budda.nagendra2013@vit.ac.in, augustasophyt.p@vit.ac.in

Abstract

Background: Floating Point (FP) computation is an indispensible task in various applications. The floating point additions and multiplications are core operations in complex multiplication, in which inputs should be given in IEEE 754 standard formats. **Methods:** The proposed design performs one double precision, or two single precision addition and multiplication operations in parallel, have been designed efficiently using resource sharing for both precision operands with minimal multiplexing circuitry. The proposed floating point multiplier makes use of Vedic multiplication algorithm, because in array multiplication sharing of multiplication is not possible. **Findings**: The proposed architecture has been synthesized using 0.18µm standard cell library. Compared to previous architectures the proposed DPdSP architecture has 20% reduced power and 5% area reduction. **Conclusion**: The DPdSP adder and multiplier consume less power than the conventional adder. Using Vedic multiplication technique shows a minimum power compared with all other type of architecture.

Keywords: Double Precision, Floating Point Arithmetic, Single Precision, Urdhva Tiryakbhyam, Vedic Mathematics

1. Introduction

Floating Point operations are of key importance in several modern applications like 3D graphics accelerators, Digital Signal Processors (DSPs), High Performance Computing etc. These applications typically involve floating point calculations in single and/or double precision format. For this reason, most of the Floating Point Units (FPUs) provide support for executing both single and double precision operations. As FP arithmetic requires larger area per unit computation, a unified multi precision architecture is required. In literature some authors have focused on multi precision adder architectures that will work for normalized numbers only[1–3]. A. Akkas[4] has shown multi-precision architectures for FP addition, which can be additionally extended with single path and double path design[5]. However, each of them is designed to support solely normalized numbers. The computation associated with sub-normal numbers and exceptional case handlings were left for software package process.

The types of architecture for multiplier are mainly array, Vedic multipliers and booth multiplier. Number of fractional bits enhances the accuracy of the output floating point number. The FP multiplier unit needs unsigned multiplier for multiplication of bits. The Vedic Multiplication technique is helpful for designing this unit. The beauty of Vedic arithmetic lies within the proven fact that it reduces the otherwise cumbersome-looking calculations in conventional arithmetic to a really easy one. The Vedic multiplication technique follows a method similar to way the human mind operates. Vedic arithmetic is supported by sixteen Sutras. Out of those sixteen Vedic Sutras the Urdhva-triyakbhyam is employed for multiplication purpose. Urdhva Tiryakbhyam is a general multiplication formula applicable to any or all cases of multiplication.

It virtually means that “Vertically and crosswise”. It is supported by a completely unique concept, through that the generation of all partial products through with the synchronous addition of those partial Products[6]. The correspondence in generation of partial products and their summation is obtained in Urdhva Triyakbhyam.

*Author for correspondence

The planned design fully supports normal and sub-normal computations, with round-to-nearest rounding methodology. Different rounding strategies can be simply enclosed. The results are compared with the best optimized implementations accessible within the literature. The main contributions of this work are summarized as follows:

- Proposed architecture for DPdSP multiplier, that can perform on-the-fly either a Double Precision (DP) or dual (two parallel) Single Precision.
- Compared to previous works, the proposed work provides more computational support, and has smaller area over-head over only DP design with similar or smaller delay overhead.

2. DPdSP Adder Architecture

The present design of DPdSP Adder architecture has been designed for the support of the dual mode operation. The projected design of double precision with two Single precision support (DPdSP) floating point adder is shown in Figure 2. Two 64-bit input operands, might contain either 1-set of double precision or 2-sets of single precision operands. First quantity contains either first input of DP or first input of each SP's, and second quantity contains second input of either DP or each SP's supported by the signal dpsp, it can be dynamically switched to either double precision or two Single precision mode (dpsp :1 → DP Mode, dpsp :0 → Dual SP Mode). All the process steps in twin mode is discussed below. DPdSP adder architecture contains 53-bit adder/subtractor architecture which will work as a 53-bit adder/subtractor when the DPdSP signal is 1 otherwise it will work as a two 24-bit adder/subtractor. It has a Leading One Detector, which detects first '1' bit position of the mantissa bits. It also contains normalization circuit along with Left and Right shifters for updating exponential bits in DPdSP adder architecture which depends on DPdSP signal. It also contains mantissa normalization circuit which normalizes the mantissa bits using the leading one detector.

3. Proposed DPdSP Multiplier Architecture

IEEE-754 standards give the floating point number representation as

$$V = (-1)\text{ sign} \times 2^{(\text{exponent-bias})} \times 1.\text{mantissa bits} \qquad (1)$$

Implicit bit is used before fraction or mantissa, whose value is '1'. Exponent bias is 2^{E-1}-1, which comes out to be 127 for single precision and 1023 for double precision exponent. Floating point multiplication is not as simple as integer multiplication. Designing of a floating point multiplier of floating point numbers represented in IEEE 754 format can be divided in different units:

- Mantissa Calculation Unit.
- Exponent Calculation Unit.
- Sign Calculation Unit.

Initially the inputs with IEEE754 format is unpacked and assigned to the check sign, add exponent and multiply mantissa. The product is positive when the two operands have the same sign; otherwise it is negative. Sign of the result is calculated by XORing sign bits of both the operands A and B. Exponents of two multiplying numbers will be added to get the resultant exponent. Addition of exponents is done using 16 bit adder.

Proposed DPdSP multiplier architecture will do 54 bit multiplication or two 27 bit multiplications and will give result accordingly. It is not possible in normal multiplication while in Vedic multiplication instead of doing direct multiplication it will do multiplication by ANDing and adding the bits accordingly. The addition depends on Vedic sutras. Figure 1 and Figure 3 shows the representation of single precision floating point number and double precision multiplication unit. Proposed 53-bit multiplier contains carry save adder which stores the carry bits and it will give exact results without doing multiplication by adding partial products of operands.

4. Vedic Multiplication

Vedic formula based architectures for floating point multiplication are power economical, suitable for high speed applications, with reduced routing complexity[8–11]. It encourages parallel computation and will

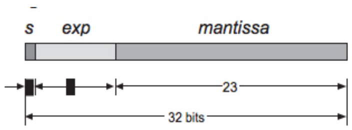

Figure 1. IEEE 754 standard single precision floating point representation.

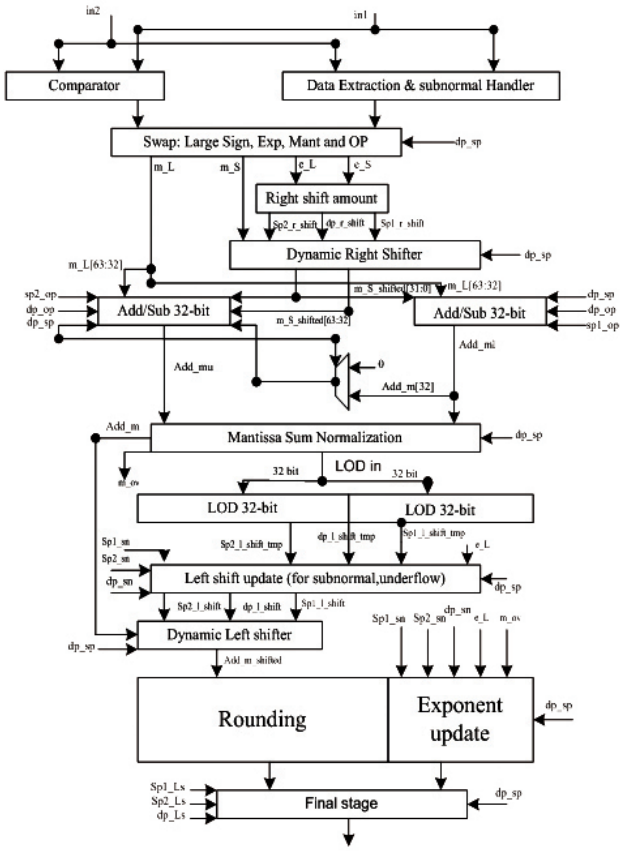

Figure 2. DPdSP adder architecture (with a 4-stage pipeline)7.

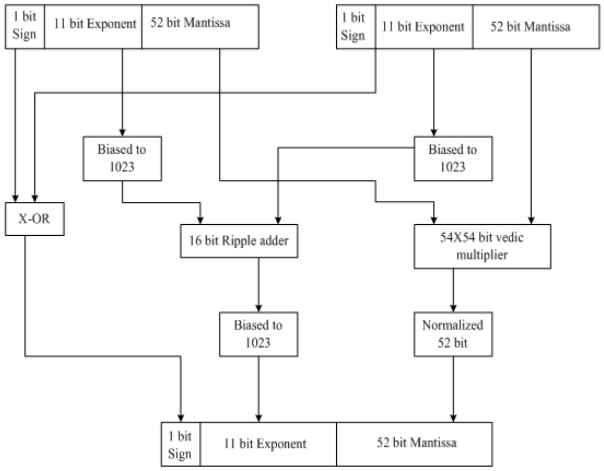

Figure 3. Proposed double precision floating point multiplier unit.

increase speed of multiplication by introducing inherent pipe-lined stages without any inflated over head. This is often achieved by exploitation of adder tree structures for addition of partial products. This formula essentially deals with fixed-point multiplication and has 3 stages for multiplication.

- Partial Product Generation.
- Accumulation.
- Final Addition.

There are number of techniques that can be used to design mantissa calculation Unit. Mantissa Calculation Unit dominates overall performance of the Floating Point Multiplier. This unit requires unsigned multiplier for multiplication of bits. This unit requires unsigned multiplier for multiplication of 54×54 bits by using this technique we design the 3×3 bit multiplier from that 9×9 multiplier and by using this 9×9 we design the 27×27 at last by the use of this 27×27 we design the 54×54 mantissa multiplier. Figure 4 shows the block diagram of 54X54 bit mantissa multiplier and Figure 5 shows the line diagram

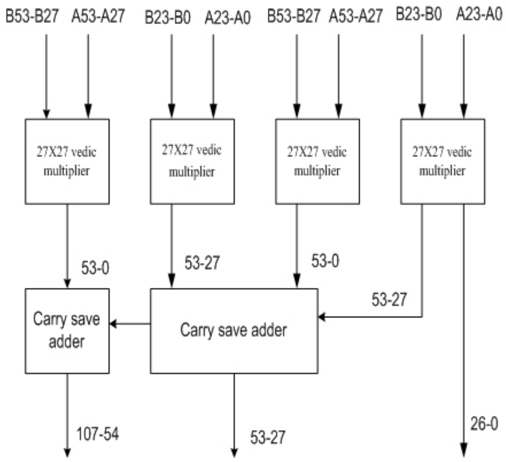

Figure 4. Proposed 53-bit mantissa multiplier unit by using vedic multiplication.

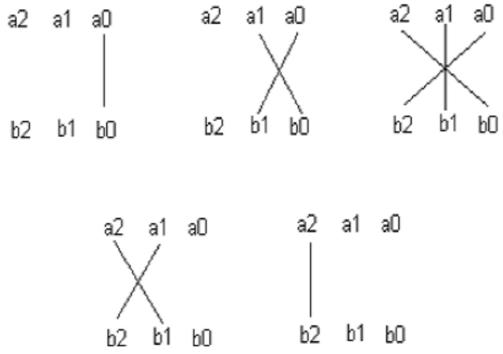

Figure 5. Line diagram for multiplication of two 3 bit numbers.

for 3 bit numbers multiplication taking place by using the Urdhva-triyakbhyam sutra with detailed execution process of sutra.

Consider the numbers A and B where $A = a_2a_1a_0$ and $B = b_2b_1b_0$.

The LSB of A is multiplied with the LSB of B:

$$s_0 = a_0b_0 \tag{2}$$

Then a_0 is multiplied with b_1, and b_0 is multiplied with a_1 and the result is added together as:

$$c_1 s_1 = a_1 b_0 + a_0 b_1 \tag{3}$$

Here c1 is carry and s1 is sum. Next step is to add c1 with the multiplication results of a0 with b2, a1 with b1 and a2 with b0.

$$c_2 s_2 = c_1 a_2 b_0 + a_1 b_1 + a_0 b_2; \tag{4}$$

Next step is to add b_0 with the multiplication results of a_1 with b_2 and a_2 with b_1.

$$c_3 s_3 = c_2 + a_1 b_2 + a_2 b_1; \tag{5}$$

Similarly the last step

$$c_4 s_4 = c_3 + a_2 + b_2; \tag{6}$$

Now the final result of multiplication of A and B is $c_4 s_4 s_3 s_2 s_0$.

Partial products are generated by using Vedic multiplication as shown in Figure 5. The final result is generated by adding partial products in proper way according to Vedic sutras[6,8,9,12].

5. Results and Discussion

As discussed in earlier sections that DPdSP adder and multiplier implementations basically involve single or double precision Floating point add/subtract, Floating point multiplier unit. Coding of each of these adder and multiplier are carried out using Verilog code and simulation is done in Modelsim/Xilinx ISE and the synthesis is carried out in Cadence encounter. Codes are simulated for functional verification and the comparison results of different adders and multipliers are shown in Table 3.The layout view of DPdSP multiplier architecture is shown in Figure 6. The comparison of proposed and existence work is shown in Table 1 and 2.

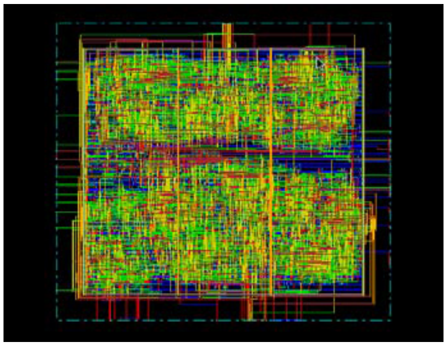

Figure 6. Layout view of DPdSP multiplier architecture.

Table 1. Comparison between existing and proposed DPdSP adder

Architecture	Power(mW)	Area(Mm^2)
Proposed DPdSP Adder	7.13	0.17
Existing DPdSP Adder	9.76	0.165

Table 2. Comparison between existing and proposed DPdSP multiplier

Architecture	Power(mW)	Area(Mm^2)
Proposed DPdSP Multiplier	51	0.266
Existing DPdSP Multiplier	61	0.28

Table 3. Comparision of different adders and multipliers

Operation	Power	Area
DPdSP Addition(proposed)	7.13 mW	174270µm^2
Single Precision Addition	2.15mW	38560µm^2
Single Precision Multiplication	6.957mW	44953µm^2
Double Precision Addition	4.49mW	118653µm^2
Double Precision Multiplication	47 mW	191045µm^2
Single Precision Multiplication Using Vedic Multiplication	5.74mW	47291µm^2
Double Precision Multiplication Using Vedic Multiplication	40.7mW	231524µm^2
Double + 2 Single Precision Multiplication	61 Mw	280951µm^2
DPdSP Multiplication Using Vedic Multiplication(proposed)	51mw	266173µm^2

6. Conclusion

This paper mainly focuses on efficient implementation of the DPdSP adder and multiplier which forms the major part in any of the floating point processor. This paper presents the performance evaluation of DPdSP architecture and also compares it with double precision multiplication using Vedic multiplication. Currently the scope of work is limited to implementing DPdSP adder and multiplier for single and double precision. Future research shall cover the efficient DPdSP adder and multiplier implementation for half, single and double precision with minimal usage of hardware based on efficient adder and multiplier.

7. References

1. Baluni, Merchant F, Nandy SK, Balakrishnan S. A fully pipelined modular multiple precision floating point multiplier with vector support. Proceedings of ISED; Kochi, Kerala. 2011. p. 45–50.
2. Manolopoulos K, Reisis D, Chouliaras V. An efficient multiple precision floating-point multiplier. Proceedings of 18th IEEE International Conference on Electronics, Circuits and Systems; Beirut. 2011. p. 153–6.
3. Isseven A, Akkas A. A dual-mode quadruple precision floating-point divider. Proceedings of 40th ACSSC; 2006. p. 1697–701.
4. Akkas. Dual-mode quadruple precision floating-point adder. Proceedings of Euromicro Symposium on Digital Systems Design; Dubrovnik. 2006. p. 211–20.
5. Akkas. Dual-mode floating-point adder architectures. Journal of Systems Architecture. 2008 Dec; 54(12):1129–42.
6. Lienhart G, Kugel A, Manner R. Using floating-point arithmetic on FPGAs to accelerate scientific body simulations. IEEE Symposium on Field-Programmable Custom Computing Machines. IEEE Computer; 2002 Apr. p. 182–91.
7. Dhanabal R, Barathi V. Implementation of floating point MAC using residue number system. 2014 International Conference on Reliability, Optimization and Information Technology (ICROIT 2014); India. 2014 Feb 6-8. p. 461–5.
8. Lee B, Burgess N. Parameterizeable floating-point operations on FPGA signals, systems and computers. Conference Record of the 36th Asilomar Conference 2002; 2002 Nov 3-6. p. 1064–8.
9. Govindu G, Zhuo L, Choi S, Prasanna V. Analysis of high-performance floating-point arithmetic on FPGAs. l8th International Parallel and Distributed Processing Symposium (IPDPS 2004); 2004 Apr 26-30. p. 2043–50. doi:10.1109/IPDPS.2004.1303135.
10. Thapliyal H. Modified montgomery modular multiplication using 4:2 compressor and CSA adder. Proceedings of the Third IEEE International Workshop on Electronic Design, Test and Applications (DELTA 06); 2006 Jan.
11. Chong K-S. A micro-power low-voltage multiplier with reduced spurious switching. IEEE Transactions on VLSI systems. 2007 Feb; 13(2):255–65.
12. Erle MA. Decimal floating-point multiplication via carry-save addition. IEEE International Conference on Application-Specific Systems, Architectures, and Processors; 2003 Jun. p. 348–58.
13. Kapre N. Optimistic parallelization of floating-point accumulation. 18th IEEE Symposium on Computer Arithmetic (ARITH'07); 2007. p. 205–16. doi:10.1109/ARITH.2007.25.
14. Seidel P-M, Even G. Delay-optimized implementation of IEEE floating-point addition. IEEE Transactions on Computers. 2004 Feb; 53(2):97–113.