Performance Comparison of Fast Multipliers Implemented on Variable Precision Floating Point Multiplication Algorithm

Neelima Koppala[1], Rohit Sreerama[2], Paidi Satish[2]

[1,2]*Sree Vidyanikethan Engineering College, Tirupathi*

koppalaneelima@gmail.com

Abstract:- The multiplication is the basic arithmetic operation in any typical processor. The multiplication process requires more hardware resources and processing time when compared with addition and subtraction. The accuracy of a multiplication mostly relies on the precision of the multiplication; a variable precision multiplier will have more accuracy than single or double precision multipliers. In this paper, a variable precision floating point multiplier is considered and total architecture for the variable precision multiplier is proposed also four different multipliers are implemented using the variable precision algorithm. The comparative study on performance analysis like delay characteristics and area is done for the considered multipliers. The best multiplier to be used for the variable precision algorithm is proposed.

Keywords:- **Array Multiplier, Carry Save Multiplier, Modified Booth Multiplier, Vedic Multiplier, Variable Precision, Floating Point Multiplication, Speed, Accuracy.**

I. INTRODUCTION

The computation speed of the computers has increased dramatically during the last decade. This increase in the speed is due to the development of VLSI technology which enabled the integration of millions of transistors on single chip [1]. Even the computational speed has increased the accuracy of the systems is not increased to that extent. Without accuracy, errors can easily occur in any system. The accuracy of a multiplication mostly relies on the precision of the multiplication; a variable precision floating point multiplier will have more accuracy than single or double precision multipliers [1].

The multiplication is the most fundamental operation in any arithmetic logic unit. Also the multipliers will take much more time for execution, so the need for speed multiplier with accuracy is desired. Many fast multipliers like array multiplier, booth multiplier etc., are proposed to increase the speed of the multiplication operation. The fast multipliers plays key role in VLSI high speed processor [2]. To design a best processor we need to consider both the accuracy and speed of operation. So a variable precision floating point multiplier when implemented with fast multipliers will have the accuracy and speed which is desired in any processors.

This paper is organised as follows section 2 recalls the variable precision floating point number representation format and the existing variable precision algorithm, section 3 describes the functionality of various multipliers used in the paper, section 4 describes the proposed architecture for the variable precision floating point multiplier, section 5 gives the results of comparative study of the various multipliers implemented on variable precision floating point multiplier, section 6 concluded the paper followed by references.

II. EXISTING VARIABLE PRECISION FLOATING POINT MULTIPLIER ALGORITHM

In this section existing variable precision floating point multiplier is described. The variable precision floating point multiplier is based on the variable precision floating point number representation format [1]. The format for variable precision floating point is shown in the figure 1. The variable precision representation is different when compared with the single or double precision that is proposed by IEEE 754 format. The variable precision floating point representation will have a sign bit (S), a type field (T), a length field (L), 16 bit exponent and significant word which varies from F(0) to F(L) [1].

The sign bit is either positive or negative depending on the value. If the value of sign bit is 1 then the number is negative, if the sign bit is 0 then the number is positive. The type field consists of two bits, it represents the type of number. Depending on the value of type field the number is considered as normalized, infinite, zero or NaN. The length field is of five bit length, it shows the number of m bit words present in the significant. The words in the significant are stored in the format of most significant F (0) to least significant F (L) [1]. The existing variable precision floating multiplier is based on the algorithm which can be implemented easily on any hardware [1]. In this algorithm only mantissas are considered. The algorithm reduces the

memory that is used to store the partial products that are generated during computation in classic multiplication method by adding the partial products as soon as they are computed.

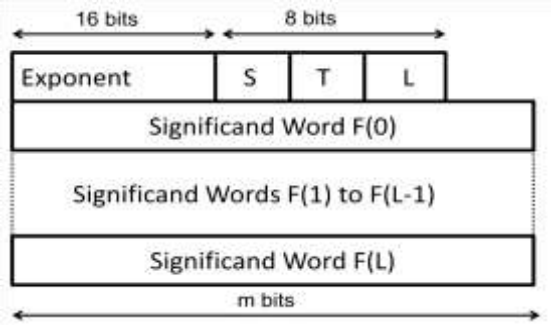

Fig 1:- Variable Precision Format.

This algorithm only uses the memory of (n x 2m) bits instead of (n^2 x 2m) bits that are used in the classic multiplication. This algorithm splits the operands A and B and the result into m bits. Depending on the value of the m the size of the multiplier and the memory are considered [1].

III. FUNCTIONALITY OF EXISTING FAST MULTIPLIERS

In this section four fast multiplier are considered and their functionality is explained. The fast multipliers play a key Role in VLSI high speed processors. The four different fast multipliers that are considered in this paper are Array Multiplier, Carry Save Multiplier, Vedic Multiplier and Modified Booth Multiplier.

A. Array Multiplier

Array multiplier is an efficient layout of a combinational multiplier. By employing array of full adders and half adders the multiplication of two binary numbers is carried out in the array multiplier. For the simultaneous addition of all the product terms the array is used in the multiplier [3]-[4]. To generate the product terms an array of AND gates are used before the adder array. The figure 2 shows the array multiplier.

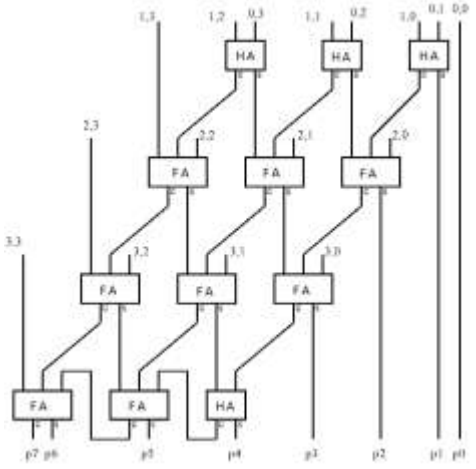

Fig 2:- Array Multiplier.

In array multiplier, consider two binary numbers A and B, of m and n bits. There are mn partial products that are produced in parallel by a set of mn AND gates. For a n x n bit multiplier requires n (n-2) full adders, n half-adders and n^2 AND gates. Also, in array multiplier worst case delay would be (2n+1) td [3]-[4]. The power consumption of the array multiplier is more and also the delay is more. Due to this the array multiplier is fast multiplier but the hardware complexity is more for the array multiplier [4].

B. Carry Save Multiplier

The carry save multipliers are much more similar to the array multipliers [5]. In the carry save multiplier the partial products are generated in parallel and the carry save adder are used to sum all the partial products which results in faster array multiplier [5].

C. Modified Booth Algorithm

A Modification of the Booth algorithm a triplet of bits is scanned instead of two bits. The booth algorithm, usually called the Modified Booth algorithm, can be generalized to any radix. In this technique the number of partial products are reduced by one half regardless of the inputs [6]. The Recoding is performed in two steps: encoding and selection. The purpose of the encoding is to scan the triplet of bits of the multiplier and define the operation to be performed on the multiplicand, as shown in the following figure 3. The modified booth algorithm is fast but the hardware complexity increases [6].

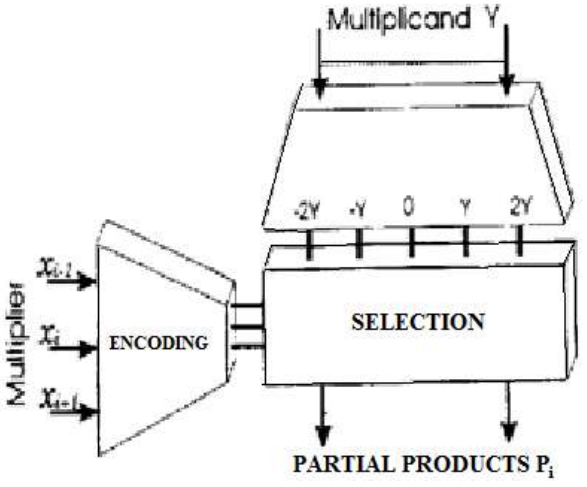

Fig 3:- Implementation of Modified Booth Algorithm.

D. Vedic Multiplication

Vedic multiplication is one of the fastest multiplication method that was followed in ancient mathematics. Nikhilam sutra is one of the Vedic methods of multiplication [7]. Nikhilam Sutra means "all from 9 and last from 10". When large numbers are

involved the nikhilam sutra is the most efficient method to consider. The compliment of the large number from its nearest base is calculated to perform the multiplication operation on it. So larger the original number, lesser the complexity of the multiplication [7]. The nikhilam sutra implementation is shown in the figure 4.

All the above four fast multipliers are considered in the paper, a comparative study is made on the performance of all the multipliers when implemented on the variable precision floating point multiplier.

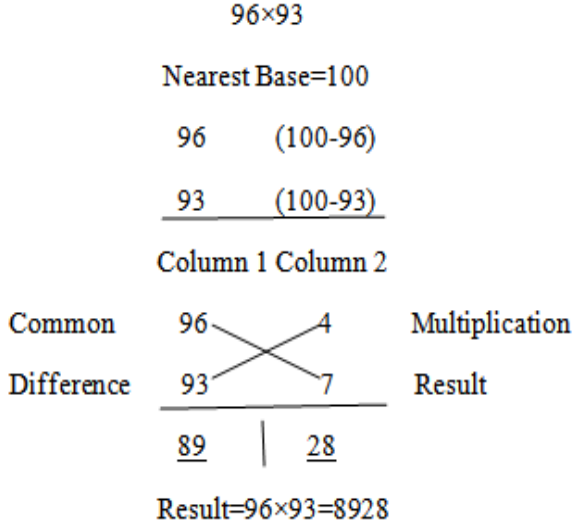

Fig 4:- Example of Nikhilam Sutra.

IV. Proposed Architecture For Variable Precision Floating Point Multiplier

In this section architecture for variable precision floating point multiplier is proposed. The figure 5 shows the architecture of the variable precision floating point multiplier.

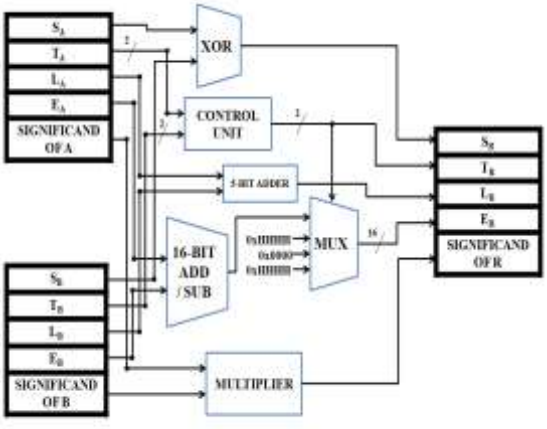

Fig 5:- Proposed Architecture for Variable Precision Floating Point Multiplier

The total architecture is based on the variable precision floating point representation. The sign bit of the result R that is S_R is obtained by the XOR operation of the sign bit of both operands A and B. The type field is obtained from the control unit. Depending on the type of the input operands the type of the result is obtained [8]. The exponent is obtained by the 16 bit adder/substractor. The significand is obtained from the multipliation of both the significands of input operands. The length field is obtained by adding the length field of both the input operands. All the additions are carried out using carry look ahead adder circuits.

V. Results

The comparative study is made on four fast multipliers implemented on variable precision floating point multiplier that are considered in the paper. The delay characteristics and the area are calculated and tabulated.

TABLE 1:- COMPARISION RESULTS OF FAST MULTIPLIERS IMPLEMENTED ON VARIABLE PRECISION FLOATING POINT MULTIPLIER.

Type of Multiplier	No. Of Slices	No. of 4 input LUTs	No. of bonded IOBs
Array Multiplier	838 out of 4656 17%	1501 out of 9312 16%	128 out of 232 55%
Carry Save Multiplier	794 out of 4656 17%	1424 out of 9312 15%	128 out of 232 55%
Vedic Multiplier	598 out of 4656 12%	1139 out of 9312 12%	128 out of 232 55%
Modified Booth Multiplier	384 out of 4656 17%	712 out of 9312 16%	128 out of 232 55%

TABLE 2:- COMPARISION RESULTS OF FAST MULTIPLIERS IMPLEMENTED ON VARIABLE PRECISION FLOATING POINT MULTIPLIER

Type of Multiplier	Max. combinational path delay	No. of MULT18X18SIOs
Array Multiplier	59.120ns	--
Carry Save Multiplier	56.854ns	--
Vedic Multiplier	54.963ns	--
Modified Booth Multiplier	55.010ns	20 out of 20 100%

The table 1 and table 2 shows the comparision results. The XILINX ISE 10.1 is used to simulate and synthesis. The FPGA family selected is Spartan 3E XC3S500E. The coading is done in VERILOG HDL.

The simulation result for the total architecture is shown in the figure 6.

Fig 6:- Simulation Result of Total Architecture of Variable Precision Floating Point Multiplier.

VI. CONCLUSION

In this paper four different fast multipliers are implemented using the variable precision floating point algorithm and design utility and path delays are compared. The comparative results concludes that the vedic multiplier will have less delay when compaed with other multipliers and modified booth algorithm will occupy less area when compared with other multipliers. So we can conclude that depending upon the requirement of the processor either vedic multiplier or modified booth multiplier can be used with the variable precision floating point algorithm. The total architecture for variable precision floating point multiplier unit which follows the variable precision format is proposed. The simulation and synthesis results are analysed using XILINX ISE.

REFERENCES

[1] Rohit Sreerama, Paidi Satish, K Neelima. "An Algorithm for variable precision based floating point multiplication", *proc* International Conference on Advances in Information Technology and Mobile Communication, AIM 2012, page no-238-242.

[2] Sumit R. Vaidya, D. R. Dandekar "Performance Comparison of Multipliers for Power-Speed Trade-off in VLSI Design" Recent Advances In Networking, Vlsi And Signal Processing.pg 263-266.

[3] Ravi, N.; Subbaiah, Y.; Prasad, T.J.; Rao, T.S. **"**A novel low power, low area array multiplier design for DSP applications" Signal Processing, Communication, Computing and Networking Technologies (ICSCCN), 2011 International Conference Page(s): 254 - 257.

[4] Gorgin, S.; Jaberipur, G.; Parhami, B. "Design and evaluation of decimal array multipliers" Signals, Systems and Computers, 2009 Conference Record of the Forty-Third Asilomar Conference Page(s): 1782 – 1786.

[5] Raghunath, R.K.J.; Farrokh, H.;Naganathan, N.; Rambaud, M.; Mondal, K.; Masci, F. Hollopeter "A compact carry-save multiplier architecture and its applications" Circuits and Systems, 1997. Proceedings of the 40th Midwest Symposium Page(s): 794 - 797 vol.2

[6] Elguibaly, F. "A fast parallel multiplier-accumulator using the modified Booth algorithm" Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions Volume: 47, Page(s): 902- 908

[7] Kumar, A.; Raman, A. "Low power ALU design by ancient mathematics" Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference Page(s): 862 – 865

[8] IEEE-754 Reference Material http://babbage.cs.qc.cuny.edu/IEEE-754.old/References.xhtml