International Journal of Advance Research, Ideas and Innovations in Technology

INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 9, Issue 2 - V9I2-1183)
Available online at: https://www.ijariit.com

Design of low power single precision floating point multiplier

A. V. Mutyalamma
mutyalu.asadi@gmail.com
Bapatla Women's Engineering College, Bapatla, Andhra Pradesh

Dr. G. Srinivasa Rao
gsr.gsrao@gmail.com
Bapatla Women's Engineering College, Bapatla, Andhra Pradesh

K. Krishnaveni
krishnaveni2002@gmail.com
Bapatla Women's Engineering College, Bapatla, Andhra Pradesh

S. Sivanandini
nandini85220@gmail.com
Bapatla Women's Engineering College, Bapatla, Andhra Pradesh

A. Dharani
dharaniannavarapu25@gmail.com
Bapatla Women's Engineering College, Bapatla, Andhra Pradesh

N. Swathi Vijaya Lakshmi
17105ec039@gmail.com
Bapatla Women's Engineering College, Bapatla, Andhra Pradesh

ABSTRACT

The floating-point multiplier is designed to provide high precision and low power for scientific simulations, engineering computations, and financial modeling applications. This paper explains a single precision floating point multiplier architecture using a Vedic multiplier. It takes two single precision floating point numbers as input and produces a single precision hovering point number as output. The proposed architecture uses a pipelined approach to increase the speed of the multiplier and to reduce the power and delay. The pipeline architecture allows multiple operations to be performed simultaneously, resulting in a faster multiplication operation. The proposed architecture generates product mantissa by use of a Vedic multiplier with a carry-save adder using a Multiplexer to reduce the power of the multiplier. The proposed architecture is compared with array multiplier-based single precision floating point multiplier and performance measures such as power and delay. The architecture will be performed in Xilinx Vivado 2016.4 software by selecting the ZED board.

Keywords: *Single Precision Floating Point, Pipeline, Vedic Multiplier, Carry Save Adder, Array Multiplier*

1. INTRODUCTION

A floating-point multiplier is a critical component in many high-performance computing applications, such as scientific simulations, data analytics, and artificial intelligence. It is responsible for performing complex arithmetic operations involving real numbers with varying precision. The IEEE 754 standard defines the representation of floating-point numbers, including single precision and double-precision format, which is widely used in many applications that require high precision.

The single precision floating point format uses 32 bits to represent a number, with 1 bit for the sign, 8 bits for the exponent, and 23 bits for the mantissa. The mantissa represents the significant digits of the number, while the exponent determines the scale of the number.

Figure-1:IEEE 754 standard single precision floating point format

Z = (-1s)*2(Exponent – Bias)*(1. Mantissa)
Where, Bias = 2(k-1)-1 (127)
Mantissa =23 bit
Exponent= 8 bits
Sign= 1 bit

II. ADVANCED WORK

Floating point multiplication is a common operation in advance Digital signal processing applications by Krishnanet al.[1],the multiplier generates only the needed MSB bits of the product mantissa by making use of divide and Conquer algorithm.. To reduce the accuracy degradation by the single operand approximation, a rounding scheme and an operand selection scheme are additionally introduced by Sravaniet al.[4], As compare to conventional multiplier it's having higher throughput, means it can execute more instructions or tasks per unit time. The Standard specifies interchange an arithmetic formats and methods for binary and decimal floating-point arithmetic in computer programming environments [2]. The multiplier implementation handles the overflow and underflow cases by Asharafyet al.[3],rounding is not implemented to give mode precision when using the multiplier in a multiply and accumulate(MAC) unit. Adders are designed in such a way to reduce the propagation delay which is also a cause for power consumption by Archanaet al.[7],the two new designs adopted for low power and high speed ripple carry adder featuring Gate Diffusion Input (GDI) structure and hybrid CMOS logic style. A multiplier design which is suitable to be applied in FFT processor had been proposed by Gonget al.[8].Here a radix-4 booth recorder algorithm with an improved 4:2 compressor structure and carry look ahead summation circuit has been adopted to design a floating point multiplier.

III. SINGLE PRECISION FLOATING POINT MULTIPLICATION ALGORITHM

To multiply two floating point numbers the following is done:
1. Obtaining the sign.
2. Adding the exponents.
3. Multiplying the significand.
4. Placing the decimal point in the significant result.
5. Normalizing the result.
6. Rounding the result to fit in the available bits.
7. Checking for underflow/overflow.

Three Stage Pipelined Floating Point Multiplication

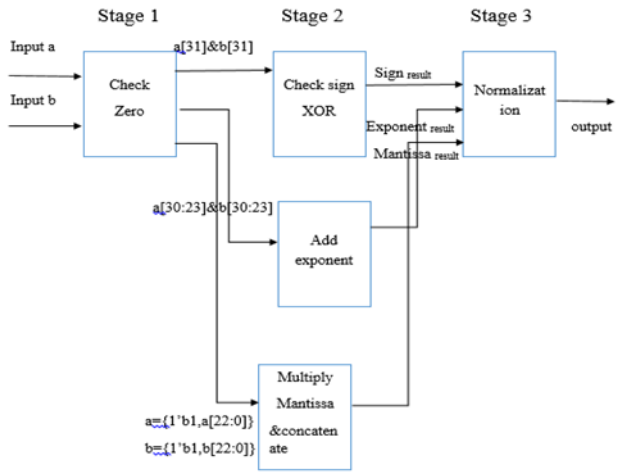

Figure 2: Three stage pipelined architecture

Sign Bit Calculation

As per the format of single precision floating point multiplier for sign bit calculation the XOR gate is used by taking the MSB bits of a[31]&b[31].The sign result will be positive if both numbers are same sign and will be negative if both numbers are of opposite sign.

Exponent addition

Exponent addition is a crucial step-in floating-point arithmetic to ensure accurate and efficient computation. The 8-bit exponent addition will perform by using Ripple carry adder. The obtained result should be subtracted by using subtractor (Ripple borrow subtractor) to the Bias.

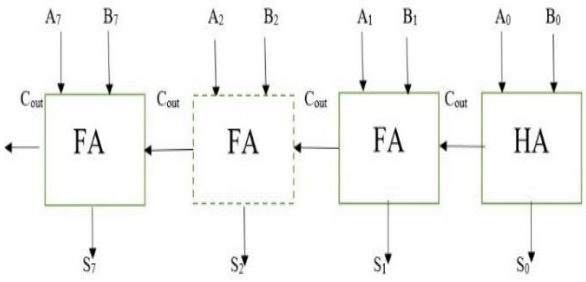

Figure-3: Ripple carry adder

Mantissa multiplication

The 23-bit mantissa are multiplied and normalized by using 24*24 vedic multiplier. If the obtained result will be occurred either underflow or overflow conditions then the shifting operation will be performed to overcome that. The obtained product should be round off to fit in to defined bits.

Vedic Multiplier Algorithm

The Vedic multiplier algorithm using the Urdhva Triyagbhyam sutra which is well-known ancient Vedic method for performing multiplication. For 24*24 bit multiplication involves breaking down input numbers into sub-parts like 3*3,6*6,12*12 performing parallel multiplication, and combining the partial products to obtain the final result. These are performed in 3 stages

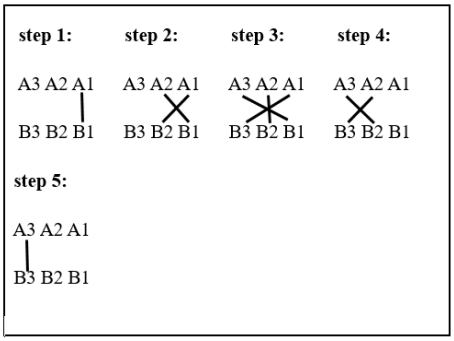

Figure-4: 3*3 multiplication using Urdhva Triyagbhyam sutra

Based on this 3*3 multiplication we design the 6*6,12*12 & 24*24 multiplication to get the mantissa result.

Pre-Processing Stage

This involves breaking down the 24 bits into smaller sub parts like 3*3 ,6*6 & 12*12 using the sutra and arranging them in a specific format. The format is such that the multiplication process becomes simpler and faster.

Multiplication stage

In this stage, the actual multiplication is performed using the prepared input 24 bits. This is done using the Urdhva Triyagbhyam sutra, which involves multiplying the subparts of the input bits those are 3*3,6*6 & 12*12 and adding them together in a specific way. The multiplication is typically performed in parallel, which means that multiple sub-parts of the numbers are multiplied simultaneously.

Post-processing stage

In this stage, the result of the multiplication is obtained by combining the partial products generated in the previous stage. This is done using a series of addition operation by using carry save adder, which involves adding together the partial products in a specific way. The final result is then obtained in the desired format.

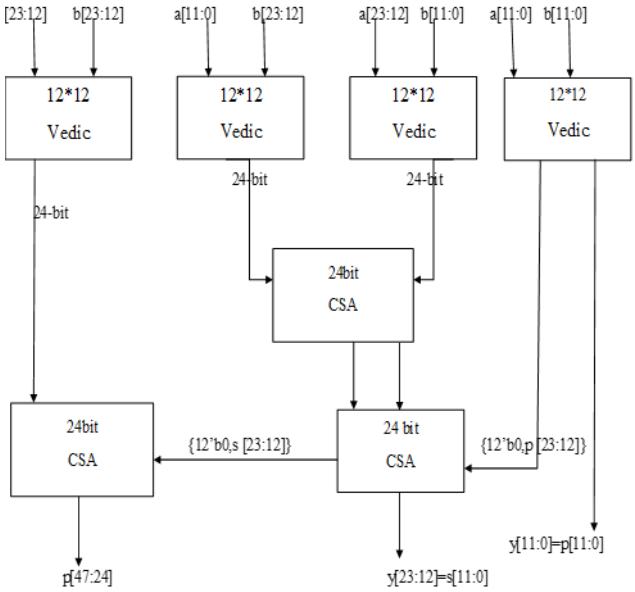

Figure-5: 24*24 Vedic multiplier

Normalization

The normalization is performed by using shifter when the obtained result goes overflow or underflow conditions. If the obtained result will be in overflow condition, then the left shift is used otherwise right shift is used.

IV. RESULT AND ANALYSIS

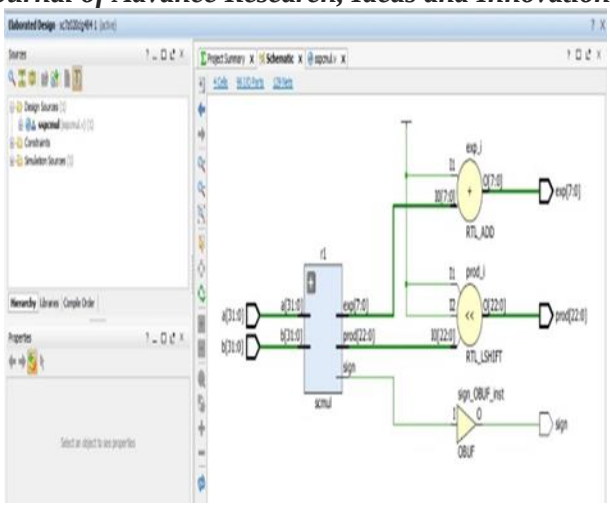

Figure-6: RTL Result

Figure-7: Simulation Result of Single Precision Floating Point Multiplier

V. CONCLUSION & FUTURE SCOPE

The proposed architecture using a Vedic multiplier hence for adding the generated partial products by using Carry save adder shows less power and delay. It supports the IEEE 754-2008 standard binary interchange format. The design of single precision floating point multiplier is implemented on Xilinx Vivado 2016.4 version software by using ZED board.

In conclusion the field of single floating-point multiplication has many future scopes for research and development. Continued efforts to optimize algorithms, explore alternative techniques, develop specialized hardware, integrate with other arithmetic operations and application-specific optimization can lead to significant improvements in performance and efficiency furthermore, reduction of power may occur by using another type of board. Delay may reduce using another type of adders & multipliers.

Table-1: Comparison Table

Parameter	Array multiplier	Vedic multiplier
LUT	418	306
Power(W)	43.222	37.971
Delay(ns)	23.632	21.594

VI. REFERENCES

[1] Thiruvenkadam Krishnan and saravanan"design of low area an high speed pipelined single precision floating point multiplier"
[2]"IEEE Standard for Floating-Point Arithmetic," IEEE Std 754-2008, pp.1–70, Aug 2008, DOI: 10.1109/IEEESTD.2008.4610935.
[3]Mohamed Al-Ashrafy,Ashraf Salem,Wagdy Anis. "An efficient implementation of floating point multiplier".DOI:10.1109/SIECPC.2011.5876905.
[4] Rapaka. Sravani, Dr.MD. Salauddin "Design of Single Precision Floating Point Multiplier Using FPGA".
[5]Benisha Bennet, S. Maflin "Modified energy efficient carrysaveadder"DOI:10.1109/ICCPCT.2015.71595 03.
[6] Taewhan Kim,W. Jao S. Tjiang. "Arithmetic using carrysaveadders".DOI: 10.1109/DAC.1998.724511
[7]S. Archana,G. Durga. "Design of low power and high speed ripple carry adder".DOI: 10.1109/ICCSP.2014.6949982.
[8] Paras Gulati, Harsh Yadav, Manoj Kumar Taleja. " Implementation of an efficient multiplier using Vedic multiplication Algorithm."DOI:10.1109/CCAA.2016.7813946.

