ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Design and Implementation of Fast Floating Point Multiplier Unit

Ch.Sreedharani
PG Scholar, VLSI & ES,
Dept of ECE,
Vidya Bharathi Institute of Technology, Janagaon, Warangal, Telangana.

MD Moin Pasha
Associate Professor,
Dept of ECE,
Vidya Bharathi Institute of Technology, Janagaon, Warangal, Telangana.

Dr.M.Ramachandra Reddy,
M.SC, Ph.D, Professor,
Dept of H&S,
Vidya Bharathi Institute of Technology, Janagaon, Warangal, Telangana.

Abstract:
Floating point numbers are the quantities that cannot be represented by integers, either because they contain fractional values or because they lie outside the range re presentable within the system's bit width. Multiplication of two floating point numbers is very important for processors. Architecture for a fast floating point multiplier yielding with the single precision IEEE 754- 2008 standard has been used in this project. The floating point representation can preserve the resolution and accuracy compared to fixed point. Pipeline is a technique where multiple instructions are overlapped in execution. Multiple operations performed at the same time by pipeline will increase the instruction throughput. In several high performance computing systems such as digital signal processors, FIR filters, microprocessors, etc multipliers are key components. The most important aim of the design is to make the multiplier quicker by decreasing delay. Decrease of delay can be caused by propagation of carry in the adders having smallest amount power delay constant.

Keywords: Floating point number, radix 4 Booth ,Encoder, Kogge Stone adder, Wallace tree structure, ,pipeline, Synopsys Design Compiler, FPGA.

1. INTRODUCTION:

The change in the level of integration brought about by up to date VLSI trends has rendered possible the mixing of many complex components in a single device. This single device has ready the systems work faster and useful for applications such as portable device like mobile phone, multimedia and methodical computation.

As the technology is advancing need for high speed is on a rise. Multiplier unit uses maximum time and power compared to other arithmetic unit. To reduce the computation time efficient multiplier units are used. Thus, the speed, power and size of a multiplier has been a key issue and therefore the many focus on their research projects. Many researchers working on innovative algorithms and circuit analysis to decrease delay, power, etc. However, with ever-increasing need of portable device, the study try to look into design of adder that work quicker. In arithmetic computing, representation of real numbers in floating point will use extensive range of values. Floating point unit is widely used in various applications. This makes the developer to work on faster floating point multiplier units.

Floating-point representation can keep its resolution and accuracy when evaluate to fixed-point representations. The floating point operators require excessive area (or time) for normal implementations. The system designers are considering area and throughput constraints for the floating point representations. Floating-point numbers have been used in arithmetic unit for several years however, for the last decade the commonly used representation is IEEE Standard for Floating-Point Arithmetic [IEEE 754- 2008].[5]

II. IEEE 754 FLOATING POINT STANDARD

The representation of using floating point numbers has been bring out by IEEE is known as IEEE 754[5] and used in all CPU implementation. For representing floating-point numbers which has negative numbers and de-normal numbers together with a set of floating-

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

point operations that operate on IEEE 754 standard. It has 4 modes of rounding which are round to nearest, round to 00, round to O, round to even and five exceptions counting when the exceptions occur. The usability of a processor will be limited when dealing with fixed point arithmetic [3]. If operations on numbers with fractions, very tiny numbers (e.g. 0.000004), or very huge numbers (e.g. 62.445x 1 05) are required, then a different form of representation is used in the floating-point arithmetic. In order to get some notation away of the way, let us talk about a few floating-point -6.24x 1 03• The negative symbol represents the sign part of the number, while the '624' shows the significant digits part of the number, and to conclude the three shows the scale factor part of the number.

The string of major digits is officially termed the mantissa part of the given number, while the factor is fittingly called the exponent part of the number .The broad representation of floating point is (-1) S* M * 2E (1) Where S stands for sign bit, M stands for - mantissa bit and E stands for - exponent bit. Single Precision Floating Point Numbers The single-precision number representations have 32 bits. There are three main fields in single precision number which are S,M and E. The 7 –digit decimal number represented in 24-bit mantissa, in that while an 8-bit represent to a base 2 which present a scaling factor with an fitting range.

Thus, a sum of 32 bit is desired for single-precision number representation. In arrange to get the stored exponent a bias of 2n-1 ,-I is added to real exponent. This bias 127 for an 8-bit exponent of the single-precision format. The totaling of bias allows using an exponent in the range from -127 to + 128, comparing to a range of 0-255 for single precision number. A range of values from 2-127 to 2+ 127, which is equal to 10-38 to 10+38 has been offered by the single precision format. If Sign bit is 0 then positive number otherwise negative number. Exponent: 8-bit and signed exponent in excess-l27 representation.Mantissa:23-bit and fractional component.

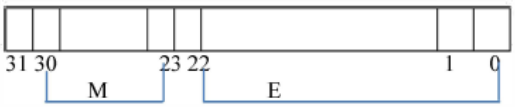


Fig l. Single Precision Floating Point

The relative size of two floating point numbers can be compared by excess-l27 representation. We will store unsigned integer representation (E' = E +127) as a replacement for storing the exponent (E) as a signed number. This exponent varies in the range E' of 0 < = E' <= 255. While the 0 and 255 range values are used to represent particular numbers (exact 0, infinity and de normal numbers), the operating range of EO becomes 1 <= E'<=" 254, thus limits the range of E to -126<= E <= 127.Table I shows the variety of values in floating point numbers.

Table l.Ranges of Floating-Point Numbers

	Binary	Decimal
Single	$\pm (2\text{-}2^{-23}) \times 2^{127}$	$\sim \pm 10^{38}$
Double	$\pm (2\text{-}2^{-32}) \times 2^{1023}$	$\sim \pm 10^{308}$

III.FLOATING POINT MULTIPLIER UNIT

1) Algorithm Figure 2 shows a algorithm flow chart for multiplier.

The mantissa of two numbers are multiplied, and the exponent are added. For floating-point multiplication, a easy algorithm is proposed

1. Add the exponents portion and subtract bias portion.

2. Multiply the mantissas portion and calculate the sign bit.

3. The output will be normalized to the prefer number of bits.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

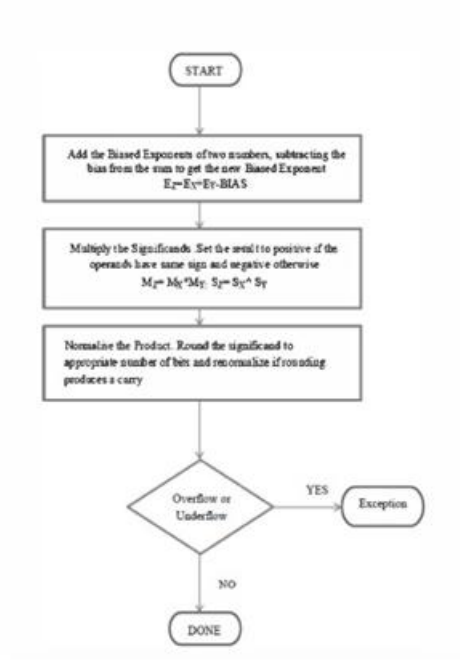


Fig 2.Alogrithm for Multiplication

2) Multiplier Flow

Multiplication of floating point number can be carried out in 3 parts [5] In the I st part, the sign product will be perform a XOR operation. In the 2nd part, the exponent bits operands are passed to an adder stage and a bias 127 is subtracted from the output. 8-bit kogge-stone adder is used for implement the addition and 2s complement addition for subtraction operations. Kogge Stone adder Kogge stone adder [7] is a related prefix form of carry look ahead adder. It create the carry in a logarithmic order. since logarithmic order it fastest adder when compared to other and also taking extra area but has lesser fan out at each stage which make better performance of adder. Order of kogge stone adder is 0 (logn) [8].Figure 4 shows the structure of Kogge Stone adder.

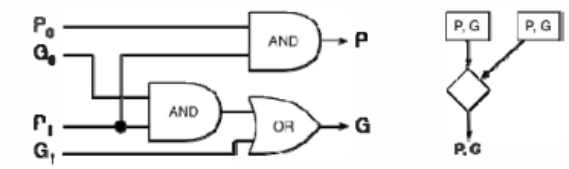


Fig 3. Diflerent Blocks of Parallel Prefix Adders

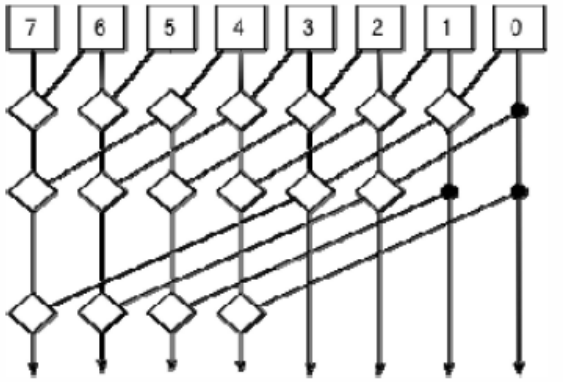


Fig 4.koggestone Adder

In the 3rd stage, find the product of the mantissa portion and the multiplication of mantissa portion is carry out in the following steps.

A. Partial product generator:- For a given multiplier [6] there are many ways to generate partial products. The radix-4 booth programming was found to be quicker in which we had found out, so it will be put into operation in the final multiplier architecture. Twelve partial products are the output of this stage. Radix 4 booth encoder To recode the terms, divide it into block of three and in that every one block overlaps the prior block by one bit. The bits are grouped from the LSB, and I st block only takes 2 bits of the multiplier for grouping (no prior block to overlap): Two bits the multiplier have been in use by the least significant block, and consider a 0 for the third bit.

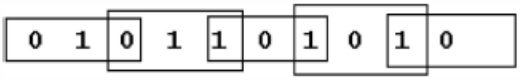


Table 2 Radix4 Booth Recoding Strategic Block Values

Block	Partial product
000	0
001	1 * Multiplicand
010	1 * Multiplicand
011	2 * Multiplicand
100	-2 * Multiplicand
101	-1 * Multiplicand
110	-1 * Multiplicand
111	0

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

B. Partial result accumulator: - The partial result obtained from the proposed multiplier will be used in the Wallace tree structure which is 4:2compressors. The propagation carry time is less in 4:2 compressor technique, used in carry save adders. Wallace tree structure Partial results is added in Wallace tree which is a new technique to propagate the carry that is ;sed in the carry save adders [4]. The 4:2 compressor structures compresses five partial products bits into three. Figure 5 shows the block diagram for the data distribution among a
tree architecture that utilizes 4:2 compressors.

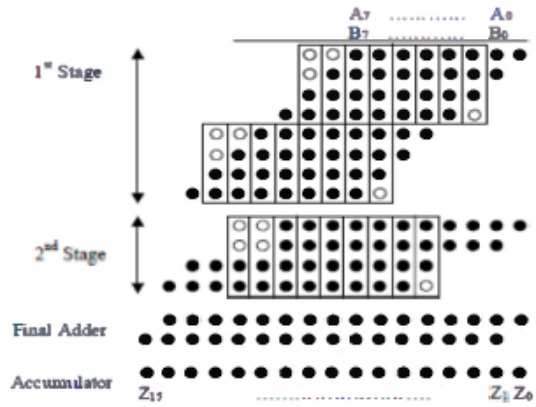


Figure 5.Data distribution among tree architecture Each packet consists of the bits that fed into a 4:2 compressor group. The partial products result can be reduce by ratio of 2: 1 for two stages of 4:2 compressors. Figure 5 show the cutback tree of 8 partial products to form two operands, added in to form a final product by using a speedy carry propagate adder.

C. Final stage adder: - The products of mantissas are specified by the 48-bit sum and carry outputs obtained from the partial product are added in the final stage adder. This stage adders should have a small amount delay and high speed. After research, comparing and implementing the power and delay uniqueness of various adders, we found out that the Kogge Stone adder is the fastest among of all the adder.

D. Normalization and rounding: - The product of mantissas is normalized and round off. The excess one is detected and the exponent is adjusted for normalization. We are reducing the implied bit which is foremost one [3]. The left over bits are reduced to a 26-bit value. For precision a few extra bits is added to the reduced value. The reduced value is finally rounded off using the rounding to nearest value to give the 23 bit mantissa of the product. A zero detect block is used in the multiplier architecture to avoid unnecessary calculations of zero in the input.

IV. PIPELINING

Pipelining is a method where multiple instructions are overlapped. It separated into segments. Each segment will carry out its operation all together with other segments. After finishing of one step of operation outcome of that step will be passes to next step in channel, carries the next operation into next segment. The final outcome of each instruction will be produced at end. The pipeline method techniques are widely used for making better performance for digital circuits. The overall performance can be enlarged by increasing the pipe lining stages which decreases the path delay in each and every stage.

a) 4-Stage Pipelining

The multiplier structure is prearranged as a four-stage pipeline. This arrangement allows to generate one result in each clock cycle , after make the first three values, outcome have been entered into the unit. To increase the performance of multiplier, four stage pipe lining is used. In order to increase the performance of multiplier operating frequency of multiplier is increased, pipeline stages are inserted in the critical path. Pipe lining stages reduce the latency in the output by four clocks [8]. The pipelining stages are followed in following steps:

I. Pre-processing the input data.
II. Addition of partial products
III. Subtracting the Bias and Compressing the partial result in Wallace tree.
IV. Normalization and final carry is adder.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

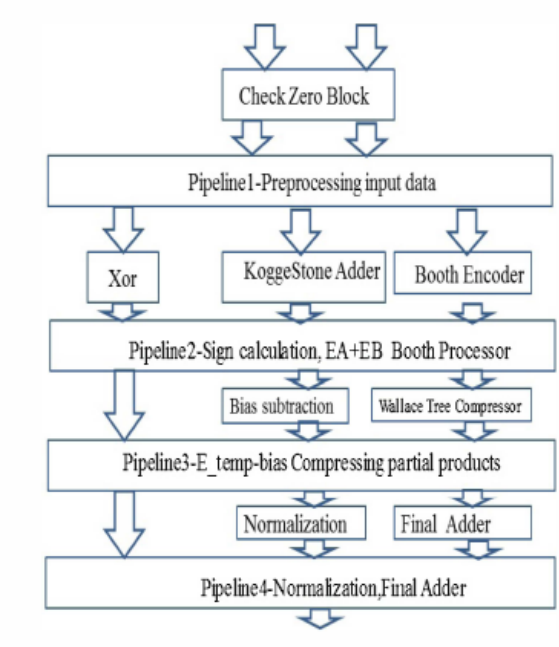


Fig 6 Proposed Multiplier unit

Pipeline registers is implement with least number of register in pipeline, positions of the pipeline register is inserted in Kogg eStone Adder in order to get the high speed. Kogge Stone adder have three pipeline stages as shown in figure 7. Implementing the pipelining register is made at three stages. First stage is where two inputs of the adder is set, next stage is after the estimate of partial product stage 1 in KoggeStone adder, and final stage is after the stage 2 of adder.

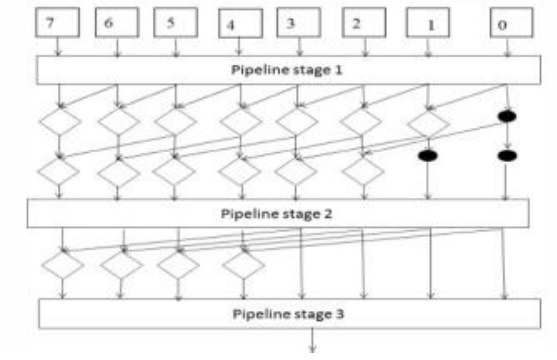
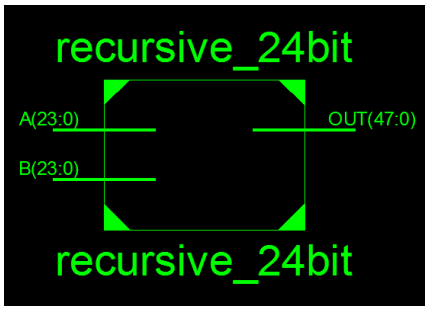


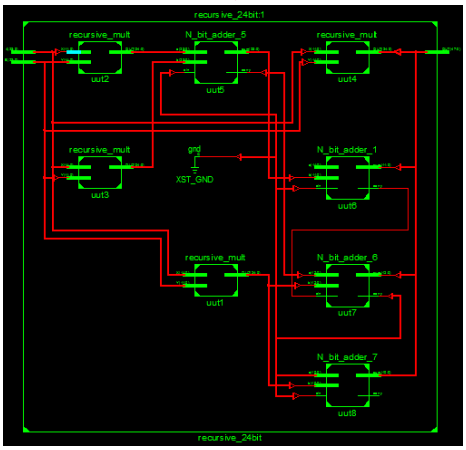
Fig 7.Pipelined KoggeStone Adder

V. EXPERIMENTAL RESULTS

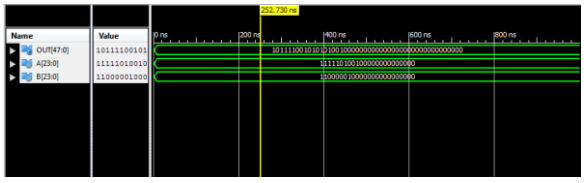
The following below figures are the existing and proposed Area, Delay & Waveform results.



a)



b)



c)

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	1265	2400	52%
Number of fully used LUT-FF pairs	0	1265	0%
Number of bonded IOBs	96	102	94%

d)

```
Timing Summary:
---------------
Speed Grade: -3

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 41.509ns
```

e)

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	1623	2400	67%
Number of fully used LUT-FF pairs	0	1623	0%
Number of bonded IOBs	96	102	94%

f)

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

```
Timing Summary:
---------------
Speed Grade: -3

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 51.360ns
```

g)

Fig 8: a)RTL Block for proposed method , b) Internal RTL Schematic for proposed method , c) Waveform for Proposed method, d)Area report for proposed method , e)Delay report for Proposed method , f) ,Area report for Existing method g) ,Delay report for existing method

VI. CONCLUSION

Proposed architecture is based fast floating point multiplier on the IEEE-754 standard single precision format is studied. The modules is written in Verilog HDL to optimize the performance architecture .The proposed design used in this project can be effectively interfaced with any DSP processor. The pipe lining is the most widely used technique to get better the performance of digital circuits. Proposed the high-speed Floating point multiplier is implemented with pipeline method.

VII.REFERENCES

[l].Anna Jain, Baisakhy Dash, Ajit Kumar Panda,"FPGA Design of a Fast 32-bit Floating Point Multiplier Unit", International Conference on Devices Circuits and Systems(lCDCS)-2012.

[2].BJeevan,S.Narender, Dr.C.V Krishna Reddy, K.Sivani,"A high speed binary floating point multiplier using Dadda algorithm" Automation Computing , Communication, Control and Compressed sensing - International Multi Conference -2013

[3].Mohammed AI Ashrafy, Asharf Salem, Wagdy Anis, "An efficient implementation of floating point multiplier" Electronics, Communications and Photonics Conference (SIECPC)-2012

[4].Jung-Yup Kang, Jean-Luc Gaudiot, "A Simple High Speed Multiplier" IEEE Transactions On Computers, Vol. 55, No. 10, October 2006

[5].S.Paschalakis, P.Lee, "Double Precision Floating-Point Arithmetic on FPGAs", In Proc. 2003, 2nd IEEE International Conference on Field Programmable Technology (FPT '03), Tokyo, Japan, Dec. 15-17, pp.352-358, 2003.

[6].Hamacher, Carl, Vranesic, Zvonko, Zaky, SafWat, "Computer Organization" Fifth Edition, pp. 367-390

[7].Neil H Weste,David Harris,Ayan Bamujee, "CMOS VLSI DESIGN",Pearson Education,3'd Edition,2009

[8]. Gong Renxi, Zhang Hainan, Meng Xiaobi,Gong Wenying,"Hardware Implementation of a High Speed Floating Point Multiplier Based on FPGA,"2009 4th International Conference on Computer Science & Education.

[9].Vojin G. Oklobdzija, Bart R. Zeydel, Hoang Q. Dao, Sanu Mathew,and Ram Krishnamurthy. "Comparison of High-Performance VLSI Adders in the Energy-Delay Space", IEEE Transactions On Very Large Scale Integration (VISI) Systems, Vol. 13, No. 6, June 2005,pp.754-758

[I 0] IEEE standards board, IEEE standard for floating-point arithmetic, 2008

Programmable Technology (FPT '03), Tokyo, Japan, Dec. 15-17, pp.352-358, 2003.

[6].Hamacher, Carl, Vranesic, Zvonko, Zaky, SafWat, "Computer Organization" Fifth Edition, pp. 367-390

[7].Neil H Weste, David Harris, Ayan Bamujee, "CMOS VLSI DESIGN",Pearson Education,3'd Edition,2009

[8]. Gong Renxi, Zhang Hainan, Meng Xiaobi,Gong Wenying,"Hardware Implementation of a High Speed Floating Point Multiplier Based on FPGA,"2009 4th International Conference on Computer Science & Education.

[9].Vojin G. Oklobdzija, Bart R. Zeydel, Hoang Q. Dao, Sanu Mathew,and Ram Krishnamurthy. "Comparison of High-Performance VLSI Adders in the Energy-Delay Space", IEEE Transactions On Very Large Scale Integration (VISI) Systems, Vol. 13, No. 6, June 2005,pp.754-758 [I 0] IEEE standards board, IEEE standard for floating-point arithmetic, 2008.