ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 5, May 2017

# Area and Delay Efficient Fused Four Term Dot Product

Anjana Anil[1], Sharon Mathew[2]

P.G. Student, Department of Electronics and Communication Engineering, Indira Gandhi College of Engineering and Technology for Womens, Nellikuhzy, Kothamangalam, India[1]

Assistant Professor, Department of Electronics and Communication Engineering, Indira Gandhi College of engineering and Technology for Womens, Nellikuhzy, Kothamangalam, India[2]

**ABSTRACT**: Dot product is one of the most widely used mathematical operation used in wide variety of applications. Past reseraches computes the fused floating point four term dot product unit with an increased area and delay.Some modifications can be brought to this work which can reduces the area and delay.On implementing the present method there will be an effective reduction in area and delay.

**KEYWORDS**: Booth multiplier, Brent kung adder, Exponent compare, LZA and normalization.

## I.INTRODUCTION

Floating-point arithmetic is attractive for the implementation for a variety of Digital Signal Processing (DSP) applications. Normally when more than two numbers are added,the process begins with adding first two numbers and the result is then added to the third one and so on.This will be a time and area consuming process,Thus this project focuses on fused numbers i.e join or blend the numbers to form a single entity.The design uses single precision floating point numbers to compute the dot product.The work goes through several steps such as exponent compare and significand alignment scheme,dual-reduction, early normalization, four-input Leading Zero Anticipation (LZA), and compound addition and rounding.Finally the obtained result will be much more accurate when compared to previous reasearches.At the end of the work the efficient reduction in area,delay and power for the computation of dot product of fused floating point unit can be obtained.

## II.RELATED WORK

1) This IEEE 754 standard specifies interchange and arithmetic formats and methods for binary and decimal floating-point arithmetic in computer programming environments. An implementation of a floating-point system conforming to this standard may be realized entirely in software, entirely in hardware, or in any combination of software and hardware. For operations specified in the normative part of this standard, numerical results and exceptions are uniquely determined by the values of the input data, sequence of operations, and destination formats, all under user control.This standard specifies formats and methods for floating-point arithmetic in computer systems standard and extended functions with single, double, extended, and extendable precision and recommends formats for data interchange. Exception conditions are defined and standard handling of these conditions is specified.
2) The discrete floating-point add–subtract unit produces the sum and difference simultaneously by executing two identical floating-point additions. However, much of the logic such as exponent comparison, significand swap and alignment in the two floating-point adders is nearly the same for the two operations.. The fused floating-point add–

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 5, May 2017

subtract unit produces the sum and difference results simultaneously by executing the shared logic such as the exponent comparison, significand swap and alignment.

3) The floating-point fused two-term dot product unit takes four normalized operands and computes the sum or difference of the two products as,P = AB ± CD.Floating-point fused dot product unit using the new alignment, earlynormalization, fast rounding and the four-input LZA is introduced.In traditional design the 4 input floating point numbers are unpacked as sign,exponent and significant.Two multiplier trees are used to provide sum and carry.The largest exponent and the difference is computed.The operation is also selected according to sign bit.The sum and carry pairs are aligned and inverted depending on the sign.Then two pairs are given to reduction tree which is a carry save adder.Then it undergoes addition and the result is normalised by using the shift amount produced by LZA.Normalized significands are rounded and exponents are adjusted according to carry and normalization shift.

### III.A Fused Floating-Point Four-Term Dot Product Unit

This design fig (1) present a novel design to compute fused four term dot product unit.It computes the dot product of four terms i.e 8 inputs. These four terms are unpacked as sign, exponent and signific and. The sign bit is given to sign logic to determine the sign of the result. The exponent bits are given to exponent compare where it finds the largest exponent and the difference to be shifted.The significand is given to dadda multiplier tree which produces sum and carry output. It then undergoes significand alignment followed by inverter.The corresponding outputs are given to the reduction trees.It consist of 4:2 carry save adder and generate pair of outputs.These outputs are added using kongg stone adder. The Kogge–Stone adder is a parallel prefix form carry look-ahead adder. The Kogge–Stone adder takes more area to implement.So that researches goes on to obtain a reduced area.Then it is normalized and send for rounding and compound addition.Finally sign bit from sign logic ,exponent from exponenet adjust and significant is obtained after rounding and compound addition.Finally four term dot product is obtained.

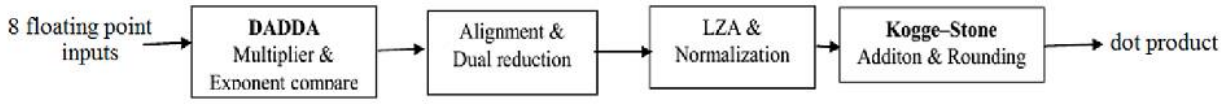


Fig 1:Block diagram of fused four term dot product unit

The main problem with this research is that it requires more area and delay.It is proved with the evidence of the summary report:

Number of Slices: : 5074 slices out of 1728
Maximum combinational path delay: 192.253ns

To overcome the above disadvantages the new design can be used.The new research reduces the area and time delay for the fused four term dot product unit.

### IV.AREA AND DELAY EFFICIENT FUSED FOUR TERM DOT PRODUCT

In previous research dadda multipliers are used as multiplier tree and kogg stone adders are used for addition.Dadda multipliers are fast multipliers,but it consumes more area.Parallel prefix adders are faster adders and these are faster adders and used for high performance arithmetic structures in industries.Kogg stone adders are parallel prefix adders which took more area for the implementation.So that the preferred design uses modified booth multiplier and brent kung adder which effectively reduces area and delay.

The traditional fused floating point unit is much better and efficient when compared to discrete floating point due to increased area and dealy.For further improvement several techniques are investigated.Fig( 2) shows the area and

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 5, May 2017

delay efficient fused floating point four term dot product unit,which consist of (1) multiplier trees,operation select and exponent compare (2) Significand alignment and dual reduction (3) LZA and normalization (4) addition,rounding and exponent adjust.

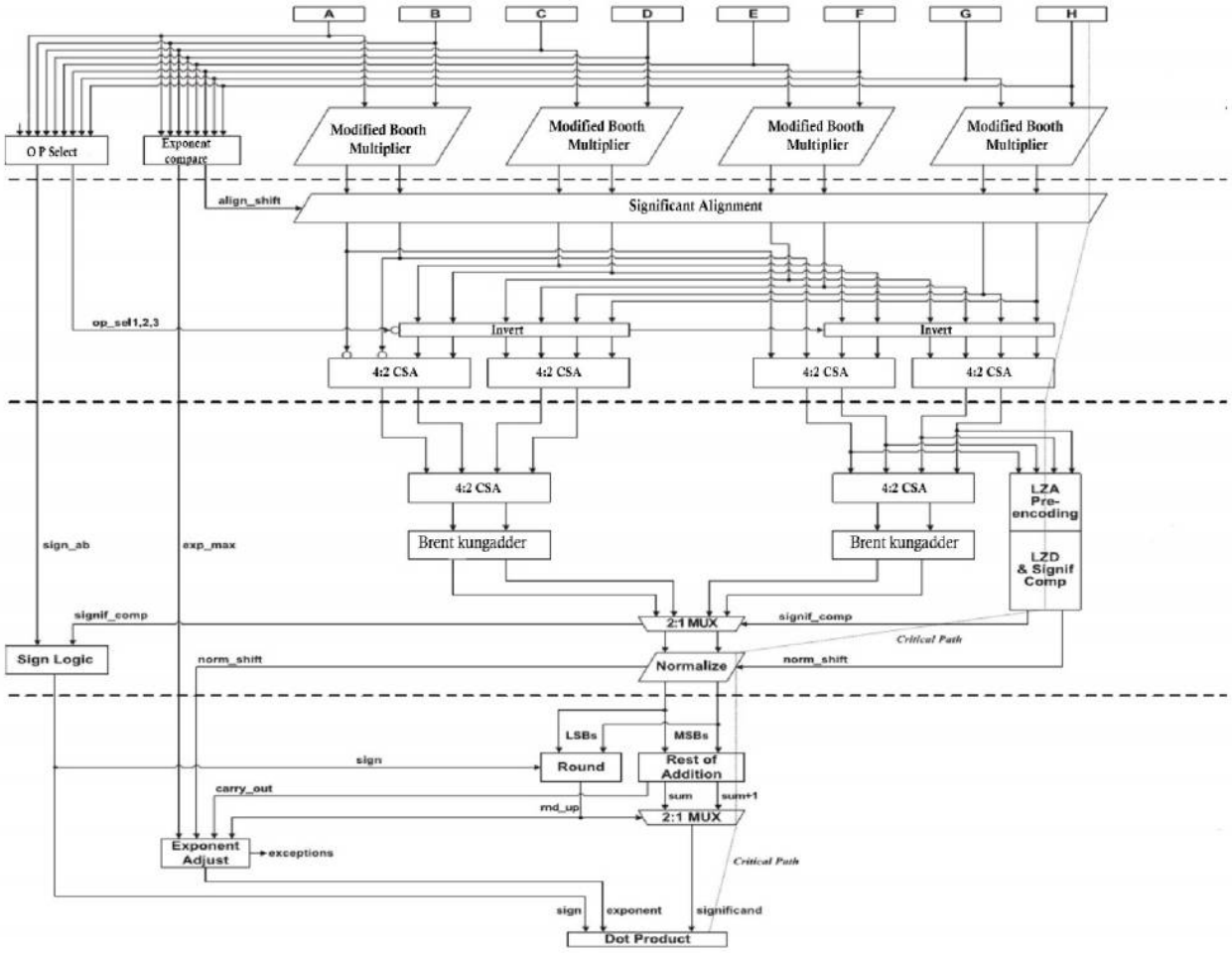


Fig 2:Area and delay efficient fused four term dot product

## V. MULTIPLIERS AND EXPONENT COMPARE

Design consist of four multipliers, operation select, exponent compare and signific and alignment logic.Any type of multiplier can be used ,modified booth multiplier is used here. Booth's multiplication algorithm is a multiplication algorithm that multiplies two signed binary numbers in two's complement notation.It is possible to reduce the number of partial products by half, by using the technique of radix -4 Booth recoding(table 1). In this method, the multiplier is appended by „0" in the LSB and then it is grouped in overlapping groups of three.The operation select logic is used to determine the various operations to be performed using the 8 sign bits of input.

$$\text{Op\_sel 1} = (sign_a \oplus sign_b) \oplus (sign_c \oplus sign_d) \oplus \text{op1} \quad (1)$$

$$\text{Op\_sel 2} = (sign_a \oplus sign_b) \oplus (sign_e \oplus sign_f) \oplus \text{op2} \quad (2)$$

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 5, May 2017

$$\text{Op\_sel 3} = (sign_a \oplus sign_b) \oplus (sign_g \oplus sign_h) \oplus \text{op3} \quad (3)$$

Table 1:Radix four multiplication

| M(i+1) | M(i) | M(i-1) | Operation |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1*Multiplicand |
| 0 | 1 | 0 | 1*Multiplicand |
| 0 | 1 | 1 | 2*Multiplicand |
| 1 | 0 | 0 | -2*Multiplicand |
| 1 | 0 | 1 | -1*Multiplicand |
| 1 | 1 | 0 | -1*Multiplicand |
| 1 | 1 | 1 | 0 |

The exponent compare logic is used to find the largest exponent and the shift amount to be applied for the significand alignment as shown in fig (3).The control logic determines the largest exponent and the shift amounts of corresponding significand pairs based on the comparison results as shown in Table ( 2).

Table 2 Control logic

| Comp_ ab_cd | Comp_ cd_ef | Comp_ ef_gh | Comp_ gh_ab | Comp_ ab_ef | Comp_ Cd_gh | Exp_ max | Shift_ ab | Shift_ cd | Shift_ ef | Shift_ gh |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | X | X | N/A | N/A | N/A | N/A | N/A |
| 0 | 0 | 0 | 1 | X | X | exp_gh | d_gh,ab | d_cd,gh | d_ef,gh | 0 |
| 0 | 0 | 1 | 0 | X | X | exp_ef | d_ab,ef | d_ab,cd | 0 | d_ef,gh |
| 0 | 0 | 1 | 1 | X | X | exp_ef | d_ab,ef | d_cd,ef | 0 | d_ef,gh |
| 0 | 1 | 0 | 0 | X | X | exp_cd | d_ab,cd | 0 | d_cd,ef | d_cd,gh |
| 0 | 1 | 0 | 1 | X | 0 | exp_gh | d_gh,ab | d_cd,gh | d_ef,gh | 0 |
| 0 | 1 | 0 | 1 | X | 1 | exp_cd | d_ab,cd | 0 | d_cd,ef | d_cd,gh |
| 0 | 1 | 1 | 0 | X | X | exp_cd | d_ab,cd | 0 | d_cd,ef | d_cd,gh |
| 0 | 1 | 1 | 1 | X | X | exp_cd | d_ab,cd | 0 | d_cd,ef | d_cd,gh |
| 1 | 0 | 0 | 0 | X | X | exp_ab | 0 | d_ab,cd | d_ab,ef | d_gh,ab |
| 1 | 0 | 0 | 1 | X | X | exp_gh | d_gh,ab | d_cd,gh | d_ef,gh | 0 |
| 1 | 0 | 1 | 0 | 0 | X | exp_ef | d_ab,ef | d_cd,ef | 0 | d_ef,gh |
| 1 | 0 | 1 | 0 | 1 | X | exp_ab | 0 | d_ab,cd | d_ab,ef | d_gh,ab |
| 1 | 0 | 1 | 1 | X | X | exp_ef | d_ab,ef | d_cd,ef | 0 | D_ef,gh |
| 1 | 1 | 0 | 0 | X | X | exp_ab | 0 | d_ab,cd | d_ab,ef | d_gh,ab |
| 1 | 1 | 0 | 1 | X | X | exp_gh | d_gh,ab | d_cd,gh | d_ef,gh | 0 |
| 1 | 1 | 1 | 0 | X | X | exp_ab | 0 | d_ab,cd | d_ab,ef | d_gh,ab |
| 1 | 1 | 1 | 1 | X | X | any | 0 | 0 | 0 | 0 |

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 5, May 2017

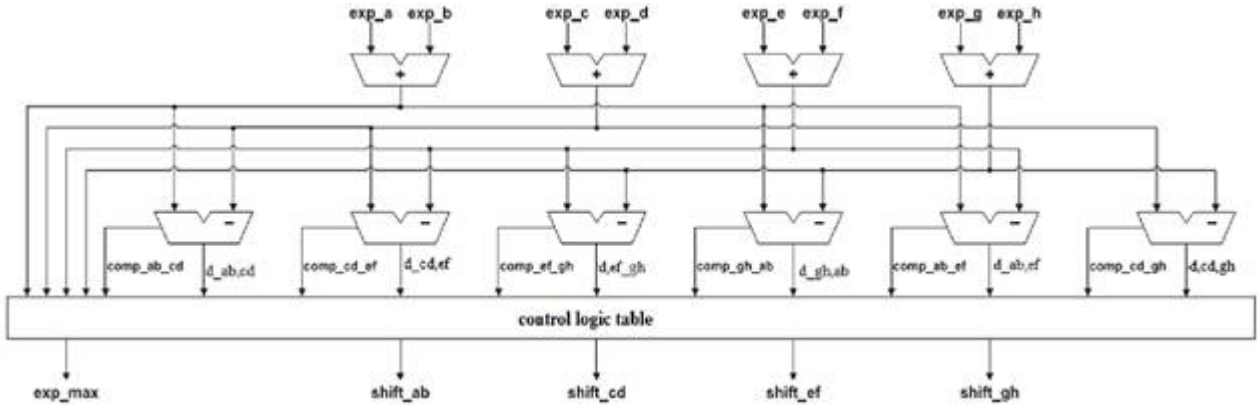


Fig 3 Exponent compare

## VI. ALIGNMENT AND DUAL-REDUCTION

The next step is to convert the sum and carry produced by the booth multiplier in to standerd form i.e it must be aligned and compressed.Since the exponent compare determines the shift amount of significand pairs,the sorting is unnecessary.It reduces the area and power consumption .Fig (4) shows the significand alignment.

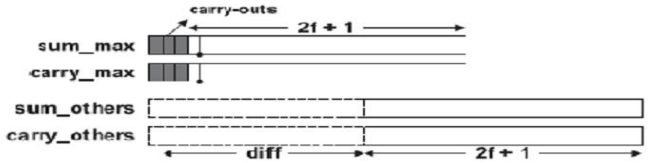


Fig 4 significand alignment.

The significand pair of the largest exponent is not shifted,and the others are shifted according to the exponent difference based on the exponent compare control logic. After the alignment ,the significand pairs are passed to the reduction trees.Here uses a dual reduction tree to eliminate the complimentation after the addition.Two levels of 4:2 CSA trees are used.The ouput from csa goes to LZA pre encoding and LZD.

## VII.LZA AND NORMALIZATION

The third stage constitues the second level 4:2 csa,LZA,normalization and sign logic.The 4:2csa generates two significand sum and carry pairs.From the first level of reduction tree the two significand pair sum and carry is given to LZA pre encoding. Leading zero anticipators predict the location of the most significant bit location of the result of a floating point addition directly from the inputs to the adder.An LZD is then employed to encode the result.The first part of the significand addition performs a PG generator as:

$$p_i = a_i \oplus b_i \quad (4)$$

$$g_i = a_i b_i \quad (5)$$

$$p_i^j = p_i p_{i-1} \dots\dots\dots p_{i-(2^j-1)} \quad (6)$$

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 5, May 2017

$$g_i^j = g_i + g_{i-1}p_i + \ldots\ldots + g_{i-(2^j-1)}p_i p_{i-1}\ldots\ldots p_{i-(2^j-2)} \quad (7)$$

Where $a_i$ *and* $b_i$ are the $i^{th}$ bits of the two significands an j is the level of the prefix tree adder.Any type of adder can be used ,but Brent Kung adder is used here. The type of structure of any adder greatly affects the speed of the circuit.The level 0 (for pi and gi)and level 1 are implemented in this project.The four input LZA can be implemented by extending the traditional two-input LZA.The four inputs are encoded by using a W vector.W=A+B+C+D.Again this W vector can be represented by using five other elements, $p_i^0, p_i^1, p_i^2, p_i^3$ and $p_i^4$ indicating $w_i$ equals to 0,1,2,3,4 respectively. Using these five elements the W vectror can be pre encoded in the form of three symbols such as $g_i$,$t_i$ and $z_i$ as shown below:

$$g_i = p_i^0 p_{i-1}^4 + p_i^1 (p_{i-1}^2 + p_{i-1}^3) + p_i^2 (p_{i-1}^0 + p_{i-1}^1) + p_i^4 p_{i-1}^4 \quad (8)$$

$$t_i = p_i^0 (p_{i-1}^2 + p_{i-1}^3) + p_i^1 (p_{i-1}^0 + p_{i-1}^1 + p_{i-1}^4) + p_i^2 (p_{i-1}^2 + p_{i-1}^3) + p_i^3 (p_{i-1}^0 + p_{i-1}^1 + p_{i-1}^4) + p_i^4 (p_{i-1}^2 + p_{i-1}^3) \quad (9)$$

$$z_i = p_i^0 (p_{i-1}^0 + p_{i-1}^1) + p_i^2 p_{i-1}^4 + p_i^3 (p_{i-1}^2 + p_{i-1}^3) + p_i^4 (p_{i-1}^0 + p_{i-1}^1) \quad (10)$$

Finally a single vector is generated that is f vector with these three symbols.

$$f_i = t_{i+1} (g_i \sim z_{i-1} + z_i \sim g_{i-1}) + \sim t_{i+1} (z_i \sim z_{i-1} + g_i \sim g_{i-1}) \quad (11)$$

The pre encoded symbols $g_i$,$t_i$ and $z_i$ can be used for the significand comparison tree which determines result of dual reduction.Depending on the signif_comp the 2:1 mux produces the significand output for normalization.

$$signif_{comp} = z_n + t_n z_{n-1} + t_n t_{n-1} z_{n-2} \ldots + t_n z_{n-1} t_{n-2} \ldots\ldots\ldots t_1 z_0 + t_n t_{n-1} t_{n-2} \ldots t_0 \quad (12)$$

Thus final sign decision can be made by using first two signs and significand comparison.

$$\text{Sign} = (sign_a \oplus sign_b) \oplus signif_{comp} \quad (13)$$

## VIII.ADDITION AND ROUNDING

Final stage of the area and delay efficient fused floating point unit contains the significand addition,rounding,exponent adjust logic.The normalized significand bits that are higher than the second LSB are passed to the main addition and the rest of the lower bits are used for the rounding as in fig(5).The compound adder takes upper f+1 bits and computes the sum and sum+1simultaneously. The significand sums can be right shifted by 1 or 2 depending on the overflows. Then, one of them is selected by the round decision fig(6).

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 5, May 2017

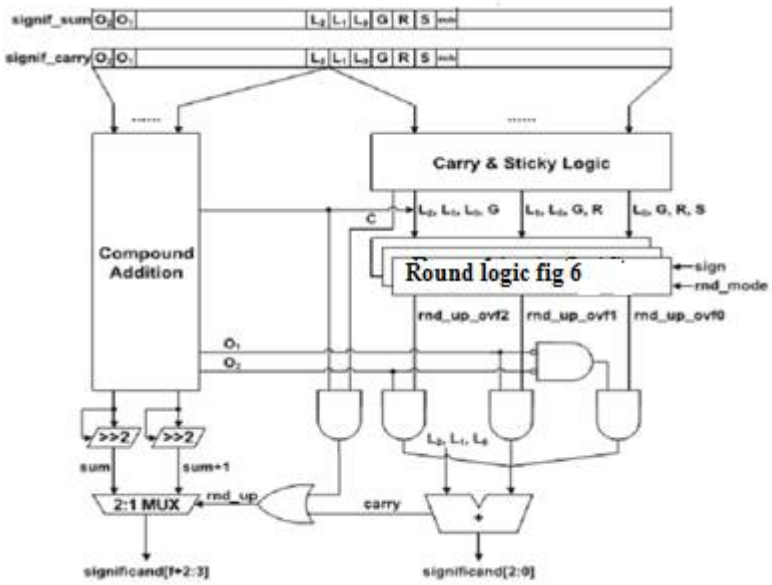


Fig 5 Addition and rounding

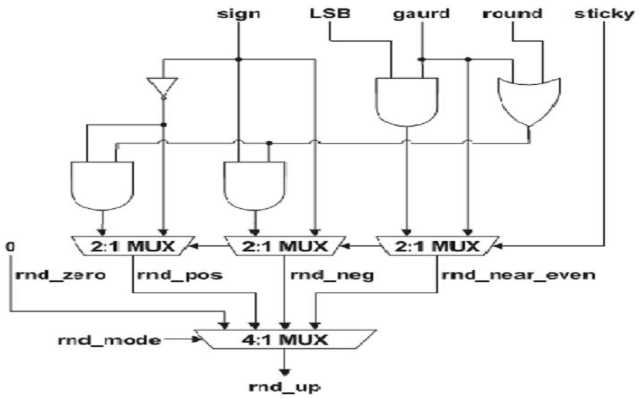


Fig 6 Round logic

## IX.EXPERIMENTAL RESULTS

In order to simulate the area and delay efficient fused four term dot product unit uses the Verilog programming environment. The Model sim softwares could run the Verilog program for the new design and the simulated ouput is shown in fig 7.The.comparison table is also shown in table (3)
Below shows the simulation result for the following input:
A=1.5=00111111110000000000000000000000 B=2.1=01000000000001100110011001100110

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 5, May 2017

C=.87=00111111010111101011100001010010 D=1.53=00111111110000111101011100001010
E=2.8=01000000001100110011001100110011 F=1.5=00111111110000000000000000000000
G=.87=00111111010111101011100001010010 H=3=01000000000000000000000000000000
After normal multiplication of these dot product the output is as follows
1.5 * 2.1 + 0.87 * 1.53 + 2.8 * 1.5 + 0.87*3 = 11.2911 = 0_10000010_01101001010100001011000
From the simulation result it shows that,
Sign = 0 Exp_adj = 10000010 Signif = 01101001010100001011000
Final Output is 0,10000010,01101001010100001011000 = 11.2911

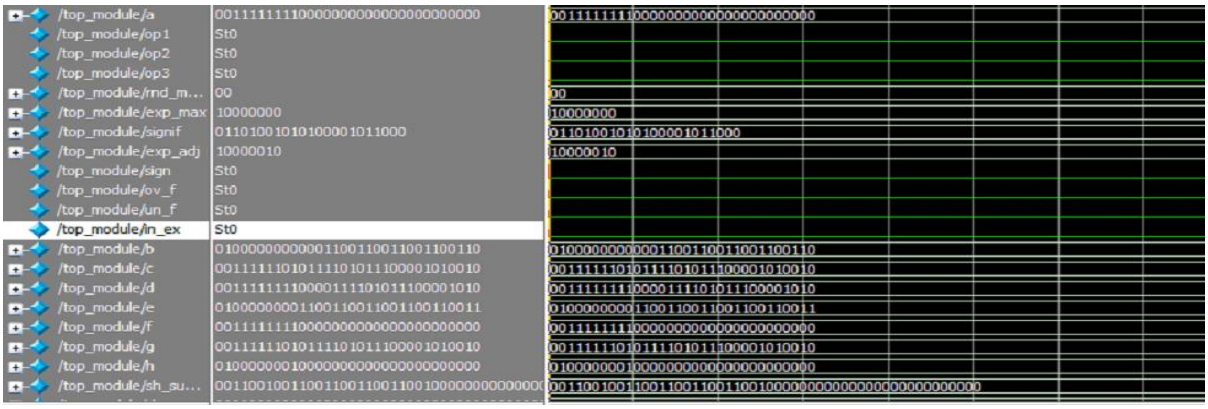


Fig 7 simulation result

Table 3 comparison table

| Design | Traditional | Proposed |
|---|---|---|
| Area | 5074 slices | 3891 slices |
| Delay | 192.253 ns | 185.697 ns |

## X.CONCLUSION

The design which is put forward uses modified booth multiplier and brent kung adder.Both these reduces the area and delay of the previous researches.Since booth multiplier only produces the half of the partial products when compared to other multipliers such as dadda and wallace.Brent-Kung adders was been proposed to solve the disadvantages of Kogge-Stone adders.When compared with past researches the newly put forwarded design shows an extreme reduction in area and delay during the execution.

## REFERENCES

[1] IEEE Standard for Floating-Point Arithmetic, ANSI/IEEE Std. 754-2008,2008
[2] H. H. Saleh and E. E. Swartzlander, Jr., "A floating-pointfused dot-product unit," inProc. IEEE Int. Conf. Comput. Design, 2008, pp. 427–431
[3] A Fused Floating-Point Four-Term Dot Product Unit Jongwook Sohn,Member, IEEE, and Earl E. Swartzlander, Jr., Life Fellow, IEEE . ieee

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 5, May 2017

transactions on circuits and systems—I
[4] VHDL Implementation of Advanced Booth Dadda Multiplier International Journal of Engineering Research and General Science Volume 3, Issue 4, July-August, 2015 ISSN 2091-2730
[5] Design of 32 bit Parallel Prefix Adders P.Chaitanya kumari,R.Nagendra IOSR Journal of Electronics and Communication Engineering (IOSR JECE)e-ISSN: 2278-2834,p- ISSN: 2278-8735. Volume 6, Issue 1 (May. - Jun. 2013), PP 01-06
[6] J. D. Bruguera and T. Lang, "Floating-point fused multiply-add: Reduced latency for floating-point addition," in Proc. 19th IEEE Symp. Comput.Arith., 2005, pp. 42–51.
[7] J. Sohn and E. E. Swartzlander, Jr., "Improved architectures for a fused floating-point addsubtract unit," IEEE Trans. Circuits Syst. I