IMPLEMENTATION OF HIGH PERFORMANCE 64-BIT MAC UNIT FOR DSP PROCESSOR

[1]K.Praveen Kumar Reddy, [2]S. Aruna Mastani
[1]Digital System & Computer Electronics. Student, JNTUACEA
[2]Assistant Professor, Dept. of ECE, JNTUCEA, Ananthapuramu
Email: [1]praveenkumar7479@gmail.com

Abstract: A design of high performance 64 bit multiplier-and-accumulator is designed in this paper. Multiplier-and-accumulator unit performs important operation in many of digital signal processing applications. The multiplier is designed using Vedic multiplication algorithm which uses Urdhava tiryagbhyam sutra and the adder is done with carry save adder and performance parameters of Vedic multiplier based MAC(multiplier and accumulator) unit is compared with existing modified Wallace multiplier based MAC unit The total design is coded with Verilog-HDL and simulation and Synthesis is done using Xilinx tool 14.3.

Keywords: Vedic multiplication, Wallace multiplication, Multiplier and accumulator (MAC), Carry save adder.

1. INTRODUCTION:

MAC unit is an unpreventable component in many digital signal processing (DSP) applications involving multiplications and /or accumulations. MAC unit is used for high performance digital signal processing systems. Multiply-accumulate (MAC) operations, which are extremely used in all kinds of matrix operations, such as convolution for filtering, dot operations and even polynomial evaluations.

$$Y(n) = \sum X1(k, n)\, X2(k, n)$$

There are so many methods are there for optimizing multiplier unit and adder unit. Some of multipliers are used in optimizing multiplier unit. Array multiplier are used in olden days for optimization it gives more power consumption but delay for this multiplier is larger. It also requires larger number of gates because of which area is also increased [2]. To overcome these disadvantages booth multipliers are also used but these are mainly used for signed numbers [3]. Later Wallace introduced Wallace multiplier in 1964 [4] which are more efficient than previous multiplier and dada multiplier refined the Wallace multiplier. Now a day's Wallace multiplier are further modified as modified Wallace multiplier is more efficient than existed multiplier. But it is not suitable for higher multipliers which require more layout process.

Basic Multiply and accumulator (MAC) unit consists of multiplier and a accumulate adder. Multiplier is used to multiply the inputs which are obtained from the memory location and given to the accumulator part which contains the sum of the previous successive products. Our design consists of 64 bit Vedic multiplier, 128 bit carry save adder and a register. MAC unit consists of three main components they are multiplier unit, adder unit and accumulation unit.

Multiplier unit multiplies the input numbers and output is given to adder unit here in addition unit addition of present output and previous output will be done and carried accumulator were storing of output in register. In our design 64, bit Vedic multiplier which accepts 64 bits input and hence the output will be 128 bits the multiplier output is given as the input to the carry save adder ,which performs addition. The function of the MAC unit is given by the following equation

$$F = \sum P_I Q_I$$

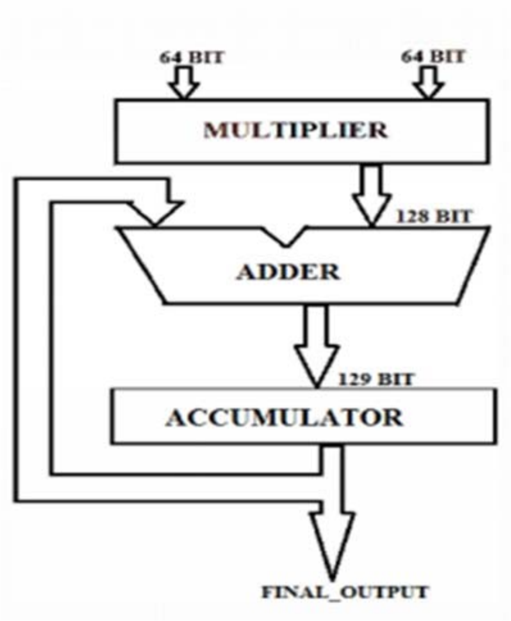


Fig 1.1: Basic architecture of MAC unit.

2. RELATED WORK:

Multipliers play a significant role in today’s digital signal processing and various other applications. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of following high speed, low power consumption, regularity of layout and hence less area or even combination of them in one multiplier, thus making them suitable for various high speed, low power, and compact VLSI implementation.

2.1 : Wallace tree Multiplier

To make the conventional Wallace multiplier more efficient we use modified Wallace multiplier. Here in this modified Wallace multiplier our main aim is to reduce the number of half adder by replacing them with full adders. Generally in conventional Wallace multipliers many full adders and half adders are used in their reduction phase. Half adders will not reduce number partial products bits. Therefore minimizing the number of half adders with a very slight increase in the number of full adders will somewhat reduces the delay. Modified Wallace multiplier consists of three stages. First stage the N×N product matrix is formed and before passing on to the second phase the product matrix is rearranged to take the shape of inverted pyramid. Now in second phase the inverted pyramid is grouped into non-overlapping group based on the below formula

$$r_{i+1} = 2[r_i/3] + r_i \text{mod} 3$$

$$\text{if } r_i \text{mod}3=0, \text{ then } r_{i+1}=2r_i/3$$

If the valve calculated from the above equation for number of rows in each stage in the second phase and the number of rows that are formed in each stage of the second phase does not match, only then the half adders will be used. The final product of this phase will be in a height of two bits and passed to the third phase .During this stage carry save adder is used for better performance rather than carry select adder and a ripple carry adder. The 64 bit modified Wallace multiplier is difficult to represent, so for understanding purpose a typical 8-bit by 8-bit reduction is shown in the below.

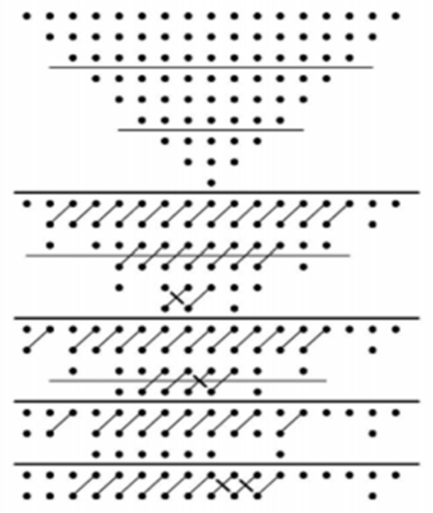


Figure 2: Modified Wallace 9bit by 9 bit reduction.

As shown above the two dots joined by a diagonal line indicates that these two are the output from the full adder. Similarly two dots joined by a crossed diagonal line indicate that these two dots are the output from half adder. Even though the method is efficient in area and speed, the circuit layout is not easy and also the circuit is quite irregular. So reduce this disadvantages which occur in this method we are going for Vedic algorithm based multiplier.

The high performance MAC unit is obtained by reducing the area and delay of the multiplier block for that purpose there are many algorithm among them some are array multiplier, booth multiplier, and conventional Wallace multiplier. Array Multiplier gives more power consumption as well as optimum number of components required, but delay for this multiplier is larger. It also requires larger number of gates because of which area is also increased; due to this array multiplier is less economical .so, to overcome the disadvantages of array multiplier an Australian computer scientist Chris Wallace developed an efficient methodology for multiplying of two integers in 1964. And named it as conventional Wallace multiplier which reduces the delay in order of O (log n) [4]. This multiplier is less regular, thus making it more difficult to layout in VLSI design.

3. IMPLEMENTING METHOD:

In MAC unit multiplier block efficiency is increased using Vedic method when compared to existing methods. Vedic mathematics for computation of algorithms of the coprocessor will reduce the computational time, complexity, power, area, etc. Vedic mathematics is based on the 16 sutras of Vedas. This system is simpler and faster than the modern mathematics. Jagadguru Swami Sri Bharati Krishna Tirthaji Maharaj who introduce Vedic mathematics and acknowledge the work of various people on Vedic mathematics. Later Anvesh Kumar, Ashish Raman, they gave the idea that Vedic sutras should be used to design the ALU [5]. They suggest two sutras for the designing of ALU. They are Nikhilam Sutra and Urdhva triyakbyham Sutra. Multiply Accumulate block is extensible used here. Multiplication algorithm is implemented using Verilog HDL.

a) Urdhva Triyakbyham Sutra:

The sutras in Vedic mathematics help to do almost all types of numeric calculations in easy and fast manner [6]. Among above sutras the Urdhva Triyakbyham is typically used for the multiplication purpose, applicable to all types of multiplication. Any bit binary number can be multiplied quickly by using this sutra. The meaning of this sutra is vertically and crosswise.

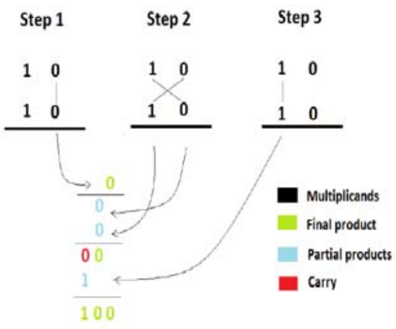


Fig3.1: 2×2 multiplier.

This sutra is used for multiplying two numbers. Here we are multiplying two 64-bit binary numbers using this sutra. Basic block of our design is 2×2 multiplier block it is designed using this Urdhva Triyakbyham technique which is shown in the above figure. For converting the above multiplication into hardware structure we need four AND gates and two HALF ADDERS. Let us consider two binary numbers a and b to be multiplied. Where a0, a1 are multiplied with b0, b1 we get q0, q1, q2, q3 as output is shown below.

.

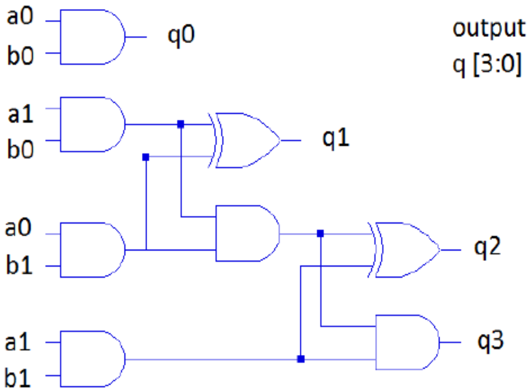


Fig 3.2 : Block diagram of 2×2 multiplier.

3.1. VEDIC MULTIPLIER:

Vedic multiplier uses hierarchical structure to reduce the number of partial product generation. The design of this multiplier starts with Multiplier design that is 2x2 bit multiplier [7]. Here, "Urdhva triyagbhyam Sutra" or "Vertically and Crosswise Algorithm" for multiplication has been effectively used to develop digital multipliers. This algorithm is quite different from the traditional method of multiplication that is to add and shift the partial products.

This Sutras will us show how to handle multiplication of a larger number (N x N, of N bits each) by breaking it into smaller numbers of size (N/2 = n, say) and these smaller numbers can again be broken into smaller numbers (n/2 each) till we reach multiplicand size of (2 x 2). For Multiplier, first the basic blocks that are the 2x2 bit multipliers can be made and then using these blocks 4x4 blocks can be implemented. Further, using 4x4 blocks, 8x8 bit block can be implemented; from 8×8 blocks, 16×16 bit block can also be implemented. This process of implementing can continue till our desired bit multiplier obtained.

The block diagram of 64-bit Vedic multiplier is as follow. Here in this diagram first 64-bit is divided into 4-block of 32-bit, and the 32-bit is divided into 4-blocks of 16-bit this will continue until basic block that is 2×2 multiplier block. But in the below diagram we shown up to four 8-bit blocks. From this 8-bits block we get 16-bits product using our Vedic algorithm which is explained below block diagram.

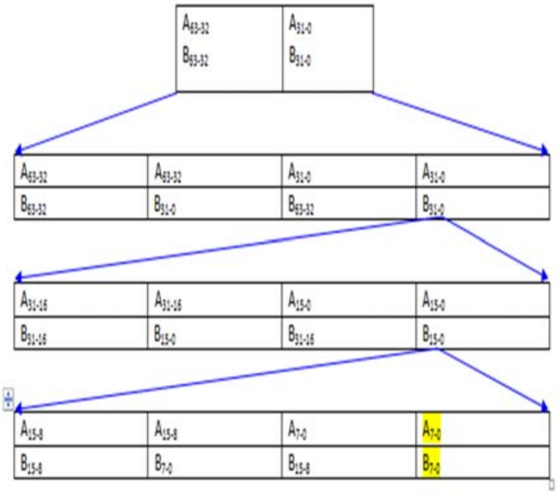


Fig 3.1: Block diagram of Vedic multiplication process.

a) 16×16 Bits Vedic multiplier:

First we have to design 8 bit block by using four 4 bit blocks which are implemented by using our basic 2 bit block. When we obtain 8*8 multiply block. Now we divided our two 16 bit numbers [a (15:0), b[(15:0)].Into 4 block of 8*8 bit multiply block. They are

First block –a [7:0], b [7:0];

Second block-a [15:8], b [7:0];

Three blocks-a [7:0], b [15:0];

Fourth block-a [15:8], b [15:8];

For every 8 bit block we get 16 bit product which are represented as q_0, q1, q2 and q_3 respectively. Now the output intermediate product is denoted as Q [31:0]. Firstly we get Q [7:0] directly from q_0 [7:0]. Where Q [7:0] = q_0 [7:0]. Next q_0 [15:8] is given to ADDER1 as one of the input and the other input is obtained from second block directly and the output obtained from ADDER1 is given to ADDER3 as one of the input and the other input is obtained from ADDER2 which gets input from third and fourth block of 8-bits multiplier. And finally we get the out from ADDER3 i.e. equal to Q[31:8].

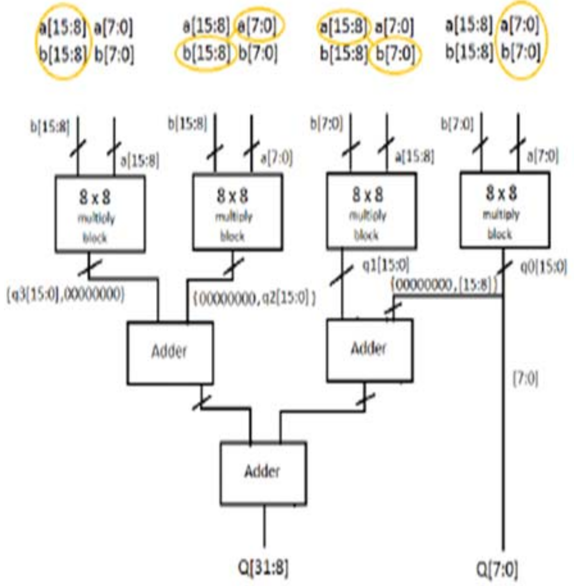


Fig 3.1:block diagram of 16×16 Vedic multiplier.

Adder tree diagram for above design is given below where we can clearly understand the process of addition.

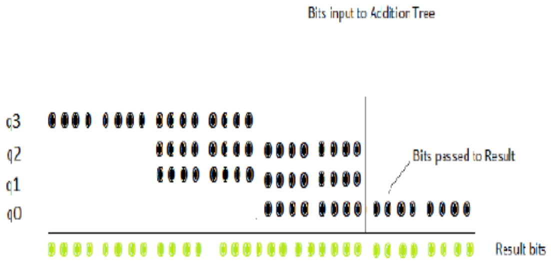


Fig 3.2.2 : adder tree of 16×16 Vedic multiplier.

3.3. Carry save adder:

After completion of multiplication of two 64-bits binary numbers we obtained 128-bit product which is given to carry save adder which give parallel out. The Carry Save Adder (CSA) is a type of Digital adder, used to compute the sum of three or more number of bits in binary form. CSA (carry save adder) gives less propagation delay and the Glitching problem in RCA is also avoided [8]. Since, the Representation of 128 bit CSA is very difficult, A Typical example of 8 bit CSA is shown below.

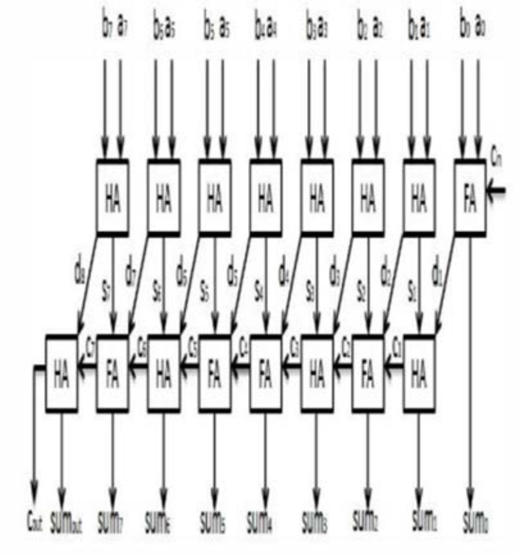


Fig 3.3: carry save adder.

Here, we compute the sum of two 128 bit binary numbers so 128 half adders at the first stage is required instead of 128 full adders. Since, we add bits of two binary numbers only [9].If, P and Q are two 128 bit numbers then i t produces the partial products and carry S_i and C_i respectively. Where,

$S_i = P \oplus Q_i$

$C_i = P_i * Q_i$

However, a CSA Produces all the output values in parallel. so that, the computation time is reduced compared to RCA. Also, Parallel in Parallel out (PIPO) is used in Accumulator Stage. From accumulator we get final output.

4. RESULTS:

The Design is developed using Verilog - HDL and Synthesized using Xilinx 14.3 ISE. As a previous work different MAC Units were developed using different combination of multipliers and adders. Here in this implementation we selected modified Wallace multiplier with carry save adder to compare with our efficient method that is Vedic multiplier using carry save adder and measure the performance parameters of two MAC units. The parameters are area, delay where area is measured in terms of number of slice and delay is measured in nanoseconds respectively. When code is checked our multiplier is shown in schematic way as follow

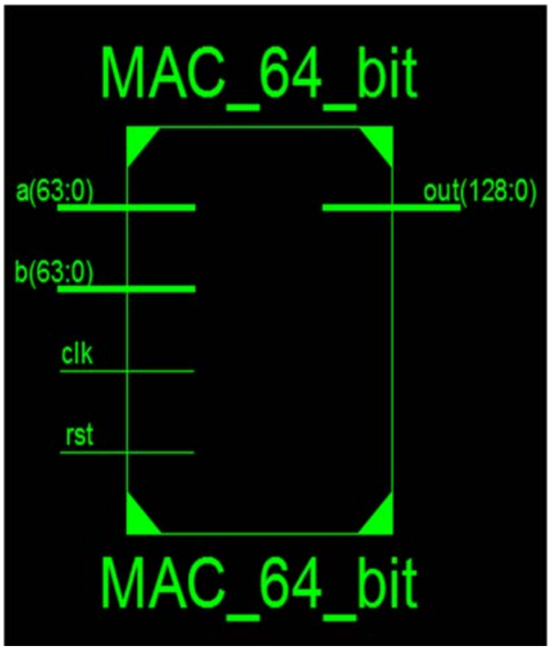
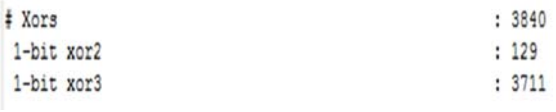


Fig 4: Block diagram of 64-bit MAC unit.

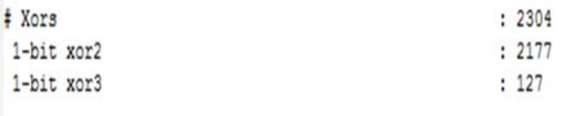
4.1 Area:

It is measured in the terms of space occupied by the hardware components which are used in the design. Let us consider our both designs where in our existing MAC unit the modified Wallace multiplier uses more number of XOR gates then that of our implemented MAC unit which uses Vedic multiplier which can be showed in synthesis report.



```
# Xors                                 : 3840
 1-bit xor2                            : 129
 1-bit xor3                            : 3711
```

Fig 4.1.1: synthesis report of 64-bits Wallace multiplier.

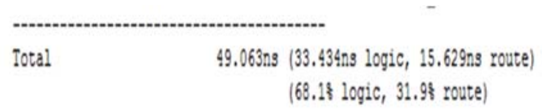


```
# Xors                                 : 2304
 1-bit xor2                            : 2177
 1-bit xor3                            : 127
```

Fig 4.1.2: synthesis report of 64-bits Vedic multiplier.

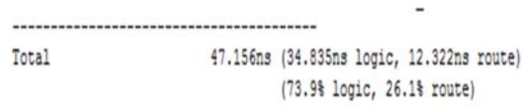
4.2 Delay:

Delay is measured in terms of signal length from input to output. To measure the delay parameter in our two MAC units we have to consider the length of the signal that travels via components from input to output end which is shown in schematic diagrams. Lets us first consider our existing 64-bits MAC unit with modified Wallace multiplier and carry save adder (CSA). Here we take the core block that is 8×8 Wallace multiplier block in this block the signal length is more from input AND gate through FULL ADDER to output end. When we consider our implemented MAC unit with Vedic multiplier and CSA the signal length in the basic block 2×2 multiplier is very less when compared with modified Wallace multiplier and also in this multiplier signal to all Vedic block will travel parallel. That's why it takes less time. Therefore delay is also less for vedic multiplier.



```
                                            -
----------------------------------------
Total                     49.063ns (33.434ns logic, 15.629ns route)
                                   (68.1% logic, 31.9% route)
```

Fig 4.2.1: Synthesis delay report of 64-bit Wallace multiplier base MAC unit.



```
                                            -
----------------------------------------
Total                     47.156ns (34.835ns logic, 12.322ns route)
                                   (73.9% logic, 26.1% route)
```

Fig 4.2.2: Synthesis delay report of 64-bit Vedic multiplier base MAC unit.

Now we go for schematic diagrams comparison of both MAC units as below

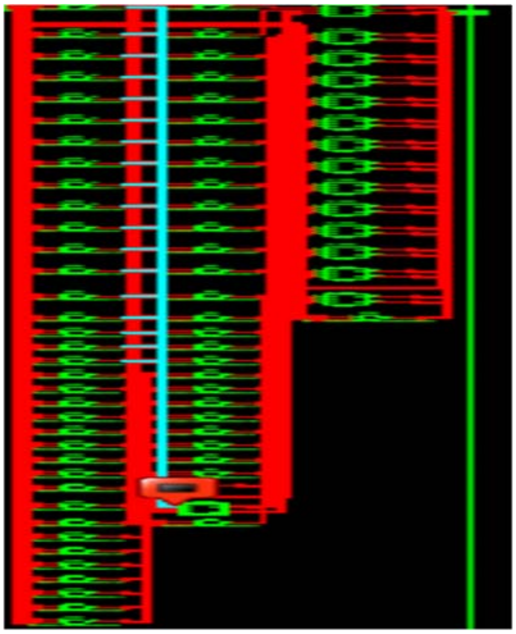


Fig 4.2.3: Schematic diagram of 16-bits Wallace multiplier.

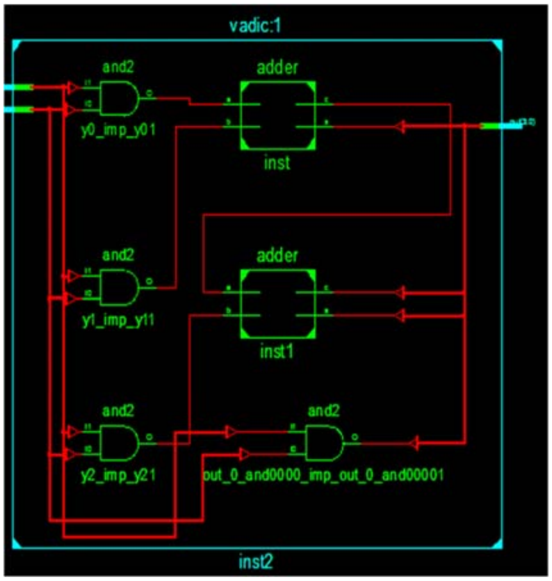


Fig 4.2.4: Schematic diagram of 2×2 Vedic multiplier.

As 2-bits Vedic multiplier signal length is shown above bit when we consider 64-bits Vedic multiplier it also have less signal path when compared with 64-bits Wallace multiplier.

Table 1: Comparison table of Wallace and Vedic MAC unit's parameters:

NO.	MAC UNITS	AREA(no. of slices)	DELAY(ns)
1	**Wallace multiplier using carry save adder**	**5770**	**49.063ns**
2	**Vedic multiplier using carry save adder**	**5547**	**47.156ns**

5. CONCLUSION:

Hence, a High Performance 64 bit MAC Unit is designed and implemented using Vedic Multiplier and Carry Save Adder. When compared with modified Wallace multiplier with carry save adder MAC Unit which is developed and more efficient than earlier MAC units using different combinations of multipliers and adders the designed Vedic Multiplier offers High Performance with Less Area, and Less Propagation Delay, which further increases the overall speed of MAC Unit. This MAC Unit is designed using Verilog - HDL and Synthesized using Xilinx 14.3 ISE.

REFERENCES:

1. "*Implementation of high performance 64 bit MAC unit by modified wallace multiplier*" IEEE CONFERENCE ON VERY LARGE SCALE INTEGRATED CIRCUITS AND SYSTEMS, JANUARY 2013.(Base paper).
2. Young-Ho seo and Dong wook kim, "*New vlsi architecture of parallel multiplier-accumulator based radix 2 modified booth algorithm*"IEEE Trans. Very Large Scale Integr. Syst., vol. 18, no. 2, pp. february 2010.
3. A. R. Cooper, "Parallel architecture modified Booth multiplier,"Proc.Inst. Electr. Eng. G, vol. 135, pp. 125–128, 1988.
4. D. Mohapatra, G. Karakonstantis, and K. Roy, "Reduced complexity wallace multiplier reduction" inProc. IEEE/ACM Int. Symp. Low Power Electron. Design, Aug. 2009, pp. 195–200.
5.] Jagadguru Swami Sri Bharath, Krsna Tirathji, "Vedic Mathematics or Sixteen Simple Sutras From The Vedas", Motilal Banarsidas, Varanasi(India),1986.
6. A.P. Nicholas, K.R Williams, J. Pickles, "Application of Urdhava Sutra", Spiritual Study Group, Roorkee (India),1984.
7.] Devika, K. Sethi and R.Panda, Vedic Mathematics Based Multiply Accumulate Unit, International Conference on Computational Intelligence and Communication Systems, CICN 2011, pp.754-757, Nov. 2011.
8. B.Ramkumar, Harish M Kittur, P.Mahesh Kannan, "ASIC Implementation of Modified Faster Carry Save Adder", European Journal of Scientific Research ISSN 1450-216X Vol.42 No.1, pp.53-58,2010.

.