© 2018 IJSRSET | Volume 4 | Issue 2 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
National Conference on Advanced Research Trends in Information and
g Technologies (NCARTICT-2018), Department of IT, L. D. College of Engineering,
Ahmedabad, Gujarat, India In association with
International Journal of Scientific Research in Science, Engineering and Technology

A Review Paper on Multiplier Algorithms for VLSI Technology

Kajal Agrawal, Milind Shah, Gaurav Asari
Vishwakarma Government Engineering College, Chandkheda, Gujarat, India

ABSTRACT

In the era of digitalization, it is required to increase the speed of digital circuits while reducing area and power consumption. In any digital system, multiplication is a key element. One of the important parameter which affects the performance of entire system is performance of multiplier unit. Therefore, it is required to design efficient multiplier unit. To improve the efficiency of multiplier unit, it's needed to optimize various parameters such as speed and area. There are different multiplier algorithms discussed and compared in this paper for performance optimization.
Keywords: Array multiplier, Wallace tree multiplier, Booth algorithm, Karatsuba algorithm, Vedic multiplier.

I. INTRODUCTION

Multipliers play an important 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 the following design targets – 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.

The general multiplication method is performed by addition, subtraction and shifting operations. After each step of calculation partial product is generated, and this is the main factor that determines the performance of the multiplier. This repetitive addition requires more time to perform the operation [1].

To reduce the number of partial products to be added, Modified Booth algorithm is one of the most popular algorithms. To achieve speed improvements Wallace Tree algorithm can be used to reduce the number of sequential adding stages.

There are many types of multiplication algorithm such as Array multiplier, Wallace tree multiplier, Booth algorithm, Karatsuba algorithm, Vedic multiplier, etc.

The functionality of any algorithm is greatly dependent on functional parameters of multipliers. The parameters include delay, memory and power. For efficient working of any algorithm which includes multiplication the selection of multiplier plays a key role. The selection of multiplier is concerned by observing the delay and area of the multiplier [2].

The organization of this document is as follows. In Section 2(Multiplier Algorithms), detail of different algorithms for multiplication is discussed. In Section 3(Comparison and Discussion), comparison of different algorithms of multiplication is given. In Section 4(Conclusion)a conclusion is given based on comparison of different algorithms for FPGA implementation.

II. MULTIPLIER ALGORITHMS

A. Array Multiplier

The traditional method for multiplication is done by using array multiplier. Array multiplier is popular due to its regular structure. It is based on add and shift algorithm. In parallel multiplication operation, number of partial products to be added is the main parameter that determines the performance of the multiplier. Each partial product is generated by the multiplication of the multiplicand with one multiplier bit. The partial products are then shifted according to their bit order and then added. Addition can be performed with normal carry propagate adder[3].

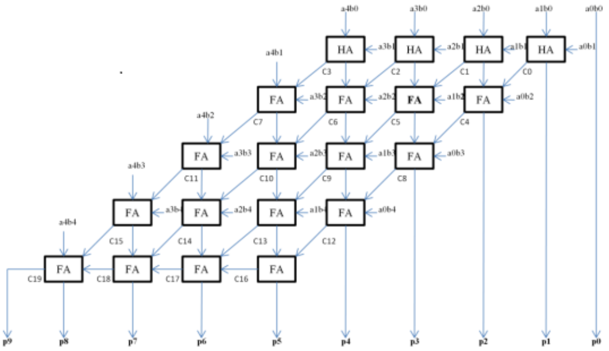

Figure 1. Structure of array multiplier

For n x n array multiplier, number of adders and gates needed are:

1. n(n-2) full adders
2. n half adders
3. n^2 AND gates

	1 0 1 0 1 0	Multiplicand
x	1 0 1 1	Multiplier
	1 0 1 0 1 0	Partial products
	1 0 1 0 1 0	
	0 0 0 0 0 0	
+	1 0 1 0 1 0	
	1 1 1 0 0 1 1 1 0	Result

Figure 2. Example of array multiplier

The advantage of array multiplier is that it has minimum complexity and regular structure. And Disadvantages are large number of logic gates, so more chip area. It has high power consumption and it is limited to 16-bits.

B. Wallace-Tree Multiplier

A Wallace tree is an efficient hardware implementation of a digital circuit that multiplies two integers, devised by Australian Computer Scientist Chris Wallace in 1964.

The Wallace tree has three steps:

1. Multiply (that is – AND) each bit of one of the arguments, by each bit of the other, yielding n^2 results. Depending on position of the multiplied bits, the wires carry different weights.
2. Reduce the number of partial products to two by layers of full and half adders.
3. Group the wires in two numbers, and add them with a conventional adder.

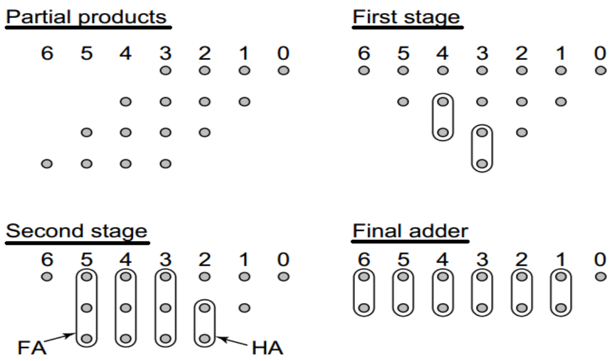

Figure 3. Example of Wallace tree multiplier

The advantage of Wallace-tree multiplier is that it becomes more pronounced for more than 16-bits. And Disadvantage is that a logarithmic depth reduction tree based CSA's has an irregular structure, therefore its design and layout is difficult.

C. Booth Algorithm

Booth's multiplication algorithm is a multiplication algorithm that multiplies two binary numbers in two's complement notation. The algorithm was invented by Andrew Donald Booth in 1950 while doing research on crystallographyatBirkbeck CollegeinBloomsbury, London. Booth's algorithm is of interest in the study of computer architecture. Booth multiplier is best for signed numbers. Booth used desk calculators that were faster at shifting than adding & created the algorithm to increase their speed.

Hence to reduce the iterations Booth's Algorithm isdeveloped for multiplying signed as well as unsignednumbers. It initiate with the ability to both add and subtract there are multiple ways to compute a product. Booth's algorithm is a multiplication algorithm that utilizes two's complement notation of signed binary numbers for multiplication. This multiplier can scan the three bits at a time hence the delay decreases. But the power consumption of this multiplier is more hence the efficiency of the system reduces [1].

Example: 3 x (-4).

m = 3 = 0011, r = (-4) = 1100.

Multiplication can be implemented by repeatedly adding one of two predetermined values A and S to a product P, then performing a rightward arithmetic shift on P.

Step:1 Let x and y represent the number of bits in m and r.

Step:2 Determine the values of A and S , and the initial value of P. All of these numbers should have a length equal to $(x + y + 1)$.

A = Fill the most significant (leftmost) bits with the value of m. Fill the remaining $(y + 1)$ bits with zeros.

A= 001100000

S= Fill the most significant bits with the value of (−m) in two's complement notation. Fill the remaining $(y + 1)$ bits with zeros.

S = 110100000

P = Fill the most significant x bits with zeros. To the right of this, append the value of r. Fill the least significant (right most) bit with a zero.

P = 000011000

Step:3 Determine the two least significant (rightmost) bits of P.

If they are 01, find the value of $P + A$. Ignore any overflow.

If they are 10, find the value of $P + S$. Ignore any overflow.

If they are 00, do nothing. Use P directly in the next step.

If they are 11, do nothing. Use P directly in the next step.

A = 0011 0000 0, S = 1101 0000 0, P = 0000 1100 0

Step:4 Perform the loop four times:

P = 0000 1100 0. The last two bits are 00, Arithmetic right shift.

P = 0000 0110 0. The last two bits are 00, Arithmetic right shift.

P = 0000 0011 0. The last two bits are 10, So $P = P + S$.

P = 1101 0011 0. Then Arithmetic right shift.

P = 1110 1001 1. The last two bits are 11, Arithmetic right shift.

P = 1111 0100 1. Arithmetic right shift.

Result: The product is 1111 0100, which is −12.

The advantages ofbooth algorithm are that it used to reduce number of stages in multiplication and it performs two bits of multiplication at once, so it requires half number of stages. And its Disadvantage is that its each stage is more complex than simple multiplier.

D. Karatsuba Algorithm

The Karatsuba algorithm is a fast multiplication algorithm. It was discovered by Anatoly Karatsuba in 1960 and published in 1962. Karatsuba algorithm uses a divide and conquers approach. Where it breaks down the inputs into Most Significant half and Least Significant half.

Recursive application of Karatsuba Algorithm

If n are four or more, the three multiplications in Karatsuba's basic step involve operands with fewer than n digits. Therefore, those products can be computed by recursive calls of the Karatsuba algorithm. The recursion can be applied until the numbers are so small that they can (or must) be computed directly.

In a computer with a full 32-bit by 32-bit multiplier, for example, one could choose $B = 2^{31} = 2{,}147{,}483{,}648$, and store each digit as a separate 32-bit binary word. Then the sums $x_1 + x_0$ and $y_1 + y_0$ will not need an extra binary word for storing the carry-over digit (as in carry save adder), and the Karatsuba recursion can be applied until the numbers to multiply are only one digit long.

Karatsuba algorithm uses divide and conquer approach where it breaks down the inputs into Most significant half and Least significant half. Karatsuba algorithm is best suited for operands of higher bit length [4].

For multiplication, break down the input into two such as x_H and x_L.

Then apply following equations:

$$a = X_H Y_H$$

$$d = X_L Y_L$$

$$e = (X_H + X_L)(Y_H + Y_L) - a - d$$

$XY = a\,r^n + e\,r^{n/2} + d$

Where, r = 10

n = bit size

Example: 1234 x 4321

Ans:

a = 12 x 43

d = 34 x 21

e = (12+34) (43+21) – a – d

= (46 x 64) – a – d

Sub-problem: 1 a = 12 x 43

a_1 = 1 x 4 = 4

d_1 = 2 x 3 = 6

$e_1 = (1+2)(4+3) - 4 - 6$
$= 11$
$X_1Y_1 = (4 \times 10^2) + (11 \times 10) + 6$
$= 516$

Sub-problem: 2 $d = 34 \times 21$
$a_2 = 3 \times 2 = 6$
$d_2 = 4 \times 1 = 4$
$e_2 = (3+4)(2+1) - 6 - 4$
$= 11$
$X_2Y_2 = (6 \times 10^2) + (11 \times 10) + 4$
$= 714$
Sub-problem: 3 46×64
$a_3 = 4 \times 6 = 24$
$d_3 = 6 \times 4 = 24$
$e_3 = (4+6)(6+4) - 24 - 24$
$= 52$
$X_3Y_3 = (24 \times 10^2) + (52 \times 10) + 24$
$= 1714$
Ans:
$a = 12 \times 43 = 516$
$d = 34 \times 21 = 714$
$e = (12+34)(43+21) - a - d$
$= (46 \times 64) - a - d$
$= 1714 - 516 - 714$
$= 484$

So, 1234×4321 is,
$XY = (516 \times 10^4) + (484 \times 10^2) + 714$
$= 5332114$

The advantage of Karatsuba Algorithm is that it reduces number of multipliers by replacing them with adders.And its Disadvantage is that it is optimal if width of input is less than 16-bit.

E. Vedic Multiplier

Vedic Mathematics is a book written by the Indian monk Swami Bharati Krishna Tirtha and first published in 1965. It contains a list of mental calculation techniques claimed to be based on the Vedas. The mental calculation system mentioned in the book is also known by the same name or as "Vedic Maths". Its characterization as "Vedic" mathematics has been criticized by academics, who have also opposed its inclusion in the Indian school curriculum. Ancient mathematics has 16 different sutras, which are taken from Atharva Ved [3]. For multiplication, there are two sutras. First is Urdhva Tiryakbhyam sutra. Urdhva-Tiryagbhyam is one of the sutra from 16-Vedic sutras which performs the multiplication operation of two decimal numbers. Urdhva-Tiryagbhyam is the general formula applicable to all cases of multiplication of a large number by another large number. "Urdhva" means vertically and "Tiryagbhyam" means crosswise therefore it is also called as vertically and Crosswise Algorithm [3, 5]. It means "Vertical & Cross-wise".

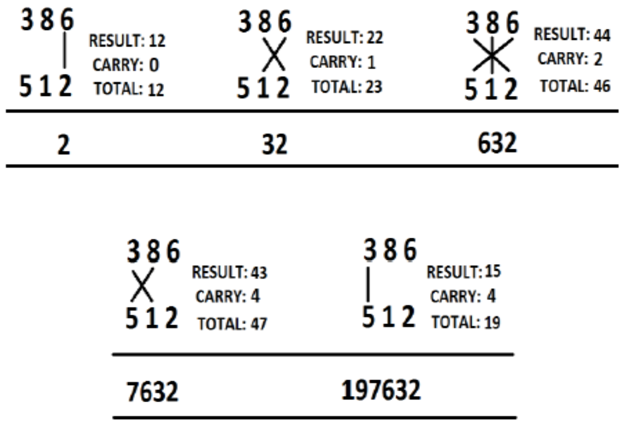

Figure 4. Example of Vedic multiplier [5]

The other one is Nikhilam Navatashcaramam Dashatah sutra. It means "All from 9 and last from 10".The sutra basically means start from the left most digit and begin subtracting 9 from each of the digits; but subtract 10 from the last digit [1].

Its advantage is that it has Minimum Delay. And its Disadvantage is that as number of bits increases, multiplication process becomes tedious.

III. COMPARISON AND DISCUSSION

Delay is very high in booth algorithm compared to other algorithms. Vedic multiplier has low delay in comparison with Array, Wallace and Karatsuba algorithm. Total Power consumption is very low in Vedic multiplier and in Booth algorithm total power is very high. Memory requirement is very high in booth algorithm and Wallace multiplier and low in Vedic multiplier. Vedic multiplier consumes very less number of LUTs for FPGA implementation. Wallace multiplier consumes higher number of LUTs. Thus, Vedic multiplier gives efficient performance for FPGA implementation.

Comparison between above different algorithm is presented Table 1.

Table 1. Comparison Of Multiplier Algorithms For 16-Bit Multiplication

	Array	**Wallace**	**booth**	**Karatsuba**	**Vedic**
Delay (ns)	High	Moderate	Very High	Moderate	Very Low
Total Power	High	High	Very High	High	Low
Memory (kb)	Moderate	Very High	Very High	High	Very Low
Num-ber of LUTs	Moderate	Very High	High	Moderate	Very Low

IV. CONCLUSION

This Paper presents different multiplier algorithms. From the comparison, Vedic multiplier is efficient among all the multiplication algorithms for 16-bit.Vedic multiplier shows the improved speed among all multipliers and it also reduces the memory requirement of the system. Therefore, Vedic multiplier is efficient and can be used in the DSP applications and Generic processors for faster computations.

V. REFERENCES

[1]. Hemangi P.Patil, S.D.Sawant. "FPGA Implementation of Conventional and Vedic Algorithm for Energy Efficient Multiplier" IEEE International Conference on Energy Systems and Applications (ICESA) Dr. D. Y. Patil Institute of Engineering and Technology, Pune, India 30 Oct-01 Nov, 2015.

[2]. G.Challa Ram, D.Sudha Rani, R.Balasaikesava, K.Bala Sindhuri. "Design of Delay Efficient Modified 16 bit Wallace Multiplier" IEEE International Conference On Recent Trends In Electronics Information Communication Technology, May 20-21, 2016, India.

[3]. S.P.Pohokar, R.S.Sisal, K.M.Gaikwad, M.M.Patil, Rushikesh Borse. "Design and Implementation of 16 x 16 Multiplier using Vedic Mathematics" International Conference on Industrial Instrumentation and Control (ICIC) College of Engineering Pune, India. May 28-30, 2015.

[4]. Arish S, R.K.Sharma. "An efficient floating point multiplier design for high speed applications using Karatsuba algorithm and Urdhva-Tiryagbhyam algorithm" IEEE International Conference on Signal Processing and Communication, March 16-18, 2015.

[5]. Paras Gulati, Harsh Yadav and Manoj Kumar Taleja. "Implementation of an Efficient Multiplier Using the Vedic Multiplication Algorithm"International Conference on Computing, Communication and Automation (ICCCA2016).