International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 12
April 2018

Performance Analysis of Different Multipliers

B.Chennaiah[1], M.Omnamasivaya[2], V.Hemanth[3], M.Suresh[4]
CH.Harika[5]

[1,2,3,4] *ECE Department, QIS College of engineering and technology, ongole.*
[5]*Assistant professor of ECE Department, QIS College of engineering and technology, ongole.*

ABSTRACT

Multiplication is one of the most important fundamental operations in applications such as Digital Signal Processing and in ALU of the microprocessors. Since multiplication dominates the execution time and power consumption of most of the digital signal processing applications, high speed and low power multipliers are essential for next generation processors.

In this study we realized that, with proper optimization the performance of the multipliers can be increased significantly, irrespective of the type. We also realized that array multiplier occupies less area and it is moderate in various parameters when compared with other multipliers.

This paper presents a detailed study of performance comparison of different types of multipliers such as Array Multiplier, Carry Save Multiplier, Wallace Tree Multiplier and Baugh Woolley Multiplier. The simulations were performed using Xilinx 14.7 software tool.

Keywords: array multiplier, Braun multiplier, Wallace tree multiplier, bough-wooley multiplier

I.INTRODUCTION

Multiplication is the most basic and frequently used operations in CPU. Scaling of one number with another number is the operation of Multiplication. Multiplication also supports for complex operations such as convolution, Discrete Fourier Transform, Fast Fourier Transform etc. The advancement in technology there is a need of increasing faster clock frequency to have faster arithmetic unit. The fastness of the arithmetic unit depends mainly on the performance of multipliers and adders. To achieve efficient performance high speed multipliers of different types are available.

Multiplication can be considered as a series of repeated additions. the number to be added is called the multiplier and the result obtained is called the product. The basic operations involved in multiplication include generating and accumulating or adding the partial products. Consequently, to speed up the entire multiplication process, these two major steps must be optimized. The two main categories of binary arithmetic multiplication involve computing unsigned numbers and computing signed numbers.

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 12
April 2018

1.1 UNSIGNED MULTIPLICATION:

Real-time computer applications require fast multiplication. By utilizing AND gates and full adders, multiplication can be implemented on the processor much in the same way as it is done by hand: multiply each digit of the multiplier by the multiplicand, thereby generating partial products and then sum up the respective partial products in order to generate the final result.

Assume that X and Y are two n-bit unsigned numbers, where X is the multiplicand and Y is the multiplier. They can be expressed as follows

$$X=\sum^{n-1} X_i 2^i_{=0} \qquad (1)$$

$$Y=\sum^{n-1} Y_j 2^j_{=0} \qquad (2)$$

The product of X and Y is P and it can be written in the following form:

$$\boldsymbol{p} = \sum_{j=0}^{n-1} \sum_{i=0}^{n-1} X_i Y_j 2^{(i+j)}$$

To illustrate further, the multiplicand X and multiplier Y j can be represented as follows:

X multiplicand: $x_{n-1}x_{n-2}x_{n-3} \dots \dots . x_1x_0$

Y multiplier: $_{-1}y_{n-2}y_{n-3} \dots \dots . y_1y_0$

P product(X*Y): $p_{2n-1}p_{2n-2}p_{2n-3} \dots \dots . p_1p_0$

Each of the partial product terms $P_n=X_iY_j$ is called summand. Equation shows the process of multiplying two unsigned binary-coded decimals (BCDs) using the paper and pencil method. Each partial product then gets stored in the arithmetic logic unit register, where it occupies memory space until the final partial product is obtained. All the partial products then get added up to generate the final product.

1.2 MULTIPLICATION OF SIGNED NUMBERS

The above procedure for multiplication works well for unsigned integers for unsigned fixed point numbers. Generally, for the multiplication of signed numbers is first converted to its 2's complement representation, thus making sure all the partial products are positive.

Consider the multiplication of an n-bit X and – Y. Converting the negative number of Y to its 2's complement format results in (2^N-Y). The product in the 2's complement context, based on direct multiplication, is

$$P'=X(-Y)=X(2^n-Y)=2^nX-XY$$

This product differs from the expected result

$$P=-XY=2^{2n}-XY$$

(2^{2n}-XY) Is the 2's complement representation of –XY. Therefore, P' deviates from the actual result P by

$$P-P' = 2^{2n}-2^nX = 2(2^n-X)$$

Where $2(2^n-X)$ is the correction factor to be added. This is done by taking the 2's

complement of X and then shifting it to the left by n positions. This corrective factor is then added to the pre computed product of P'.

If both the multiplicand and multiplier are negative, their 2's complements will be multiplied .Consider that an n-bit –X is to be multiplied with an n-bit –Y. The product P' is given by

$$P' = (2^n\text{-X})\,(2^n\text{-Y}) = 2^{2n} - 2^n X - 2^n Y + XY$$

However, the expected result is

P=XY

The difference is

$$P - P' = -2^{2n} + 2^n X + 2^n Y$$

The term 2^{2n} denotes a carry-out bit from the Most Significant Bit (MSB) that can be ignored. To get the correct result, correction factors for both multiplier and multiplicand should be added.

2.ARRAY MULTIPLIER

Array multiplier is an efficient layout of a combinational multiplier. Multiplication of two binary number can be obtained with one micro-operation by using a combinational circuit that forms the product bit all at once thus making it a fast way of multiplying two numbers since only delay is the time for the signals to propagate through the gates that forms the multiplication array.

Array multipliers are essentially regular structures and are simple to expand. The array multiplier works based on the principle of add and shift algorithm. A simple diagram of the 4*4 array multiplier is shown in the figure . Each partial product is generated by the multiplication of the multiplicand with one multiplier bit. The partial products are shifted according to their bit orders and the added. In array multiplication we need to add as many partial products as there are multiplier bits. In this multiplier, the product bits can be generated using AND gates and the summation of partial products can be performed using full adders and half adders. In an nxn array multiplier, nxn and gates compute the partial products and the addition of partial products can be performed by using n*(n-2) full adders and n half adders. The bit array multiplier is easy to design and uses a pipelined architecture. Since the worst case delay of the bit array multiplier is proportional to the width of the multiplier, the speed performance will be degraded for wide fan-in multipliers.

In order to perform signed multiplication, 2's complement number system is used to represent the multiplicand and multiplier. This implies that all the adders in a particular stage should be of equal bit length. To achieve this, the sign bits of the partial products in the initial row and the sum and carry signals of each adder stage are extended. The extension is carried out until the signals width matches the width of the largest absolute value signal in that stage.

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 12
April 2018

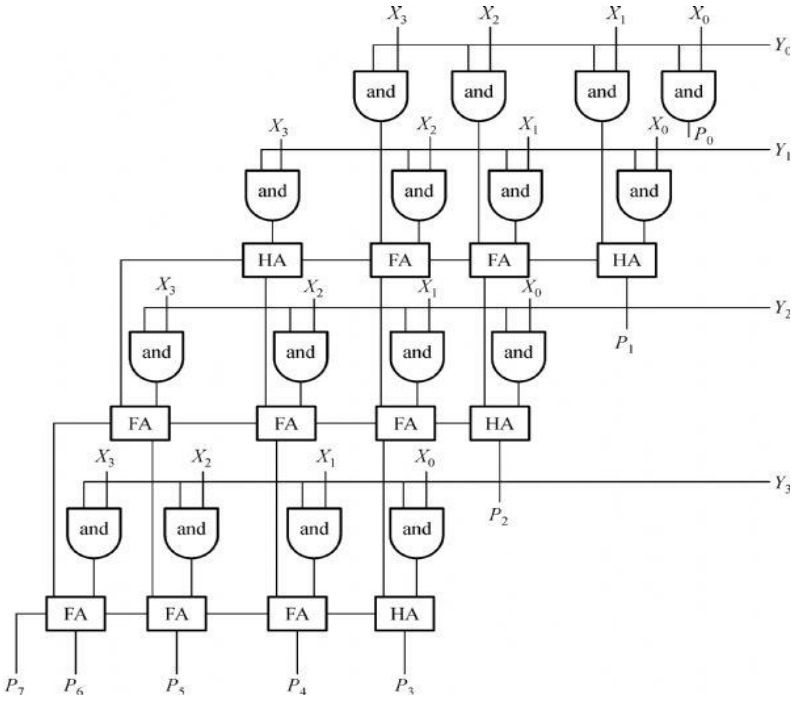

Fig : 4*4 array multiplier

3.BRAUN MULTIPLIER

Braun multiplier is a simple parallel multiplier that is commonly known as the carry save array multiplier. This multiplier is restricted to performing multiplication of two unsigned numbers. It consists of an array of AND gates and adders arranged in an iterative structure that does not require logic registers. This is also known as the non-additive multiplier since it does not add an additional operand to the result of the multiplication.

An n*n nit Braun multiplier requires n(n-1) adders and n2 AND gates. One efficient implementation of the Braun multiplier is the regular layout of the adder array. The internal structure of the full adder used in the Braun multiplier is depicted. This makes Braun multipliers ideal for very large scale integration (VLSI) and application specific integrated circuit (ASIC) realization.

Each of the X_iY_j product bits is generated in parallel with the AND gates. Each partial product can be added to the previous sum of partial products by using a row of adders. The carry out signals are shifted one bit to the left and are then added to the sums of the first adder and the new partial product. The shifting of the carry out bits to the left is done by a carry save adder (CSA). As the carry bits are passed diagonally downward to the next adder stage, there is no horizontal carry propagation for the first four rows. Instead, the respective carry bit is “saved” for the subsequent adder stage. Ripple carry adders (RCA) are used at the final stage of the array to output of the final result.

The Braun multiplier performs well for unsigned operands that are less than 16 bits, in

terms of speed, power and area. Besides it has a simple and regular structure as compared to other multiplier schemes. However, the number of components required in building the Braun multiplier increases quadratically with the number of bits. This makes the Braun multiplier inefficient and so it is rarely employed while handling large operands. Another pitfall of the Braun multiplier is its potential susceptibility to glitching problems at the last stage of the full adders due to the exploitation of the ripple carry adders (RCA).

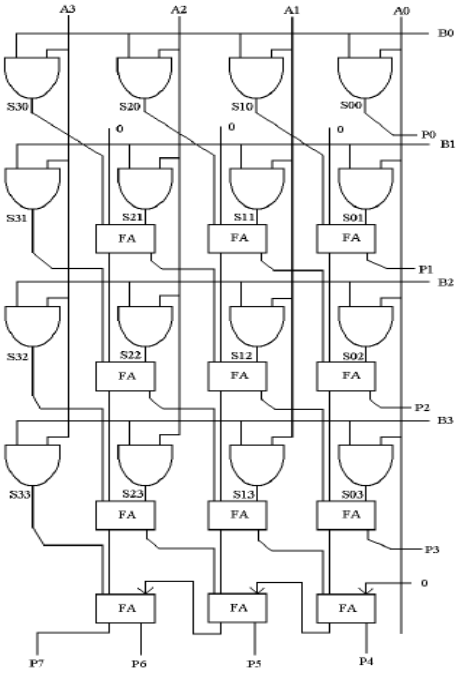

Fig :4*4 Braun multiplier

4. WALLACE TREE MULTIPLIER

Wallace tree multiplier is based on a tree structure. The Wallace tree multiplier minimizes the latency by using carry save addition algorithm. This Wallace tree structure is suitable for multipliers whose computation time increases as the logarithm of the number of bits to be multiplied increases.

The architecture of a 4*4 Wallace tree multiplier is shown in the figure. It consists of 16 AND gates to generate the partial products. The partial products generated are added using half adders and full adders. The addition process of the multiplication is shown above. The 4*4 Wallace tree multiplier consists of two half adders. It also consists of ten full adders. Generally, for any adder there will be two outputs. They are sum and carry. In the Wallace architecture shown in figure .the adders which are present in the first stages forward the carry to the adders in the next stage. The adders which are present in the last stage forward the carry to the adder present in the same stage.

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 12
April 2018

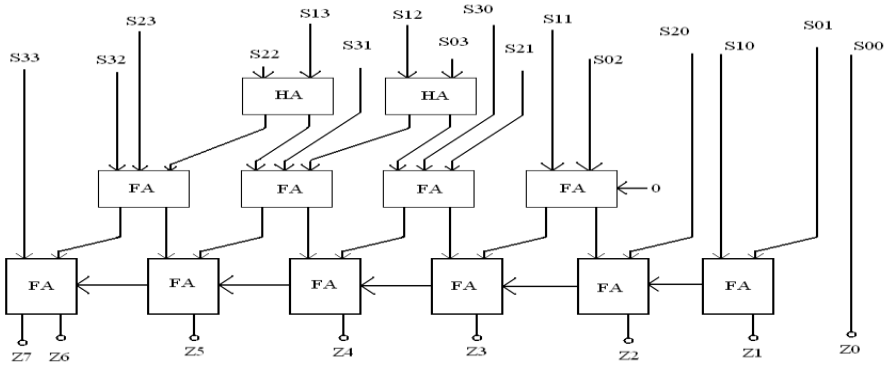

Fig : Wallace tree multiplier

5. BAUGH-WOOLEY MULTIPLIER

The Baugh Wooley multiplier is an enhanced version of the Braun multiplier. It is designed to cater to multiplication of both signed and unsigned operands which are represented in the 2's complement number system. The partial products are adjusted so that the negative signs are moved to the last steps, which in turn maximize the regularity of the multiplication array.

The architecture of the Baugh Wooley multiplier is also based on the carry save algorithm. It inherits the regular and repeating structure of the array multiplier. Each partial product bit is the result of an AND gate of a multiplier. The signs of all partial product bits are positive.

The multiplication result of a 4*4-bit multiplier is an 8-bit output, hence there are eight vertical columns shown in table. Each of these product terms is made up of one AND gate. Each column represents the addition in accordance with the respective weight of the product term. In this scheme a total of n(n-1)+3 full adders are required. Hence for the case of n=4, the array requires 15 adders.

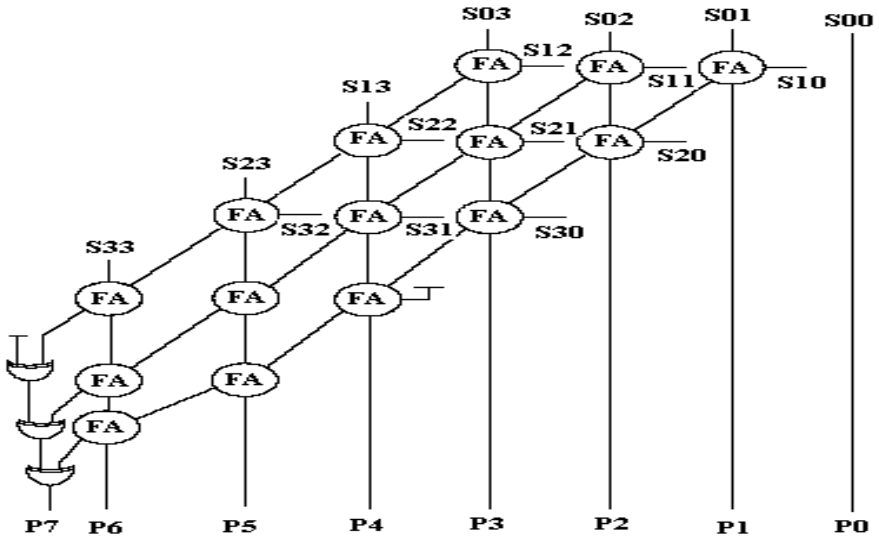

Fig: 4*4 Baugh wooley multiplier

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 12
April 2018

6. SIMULATION RESULTS

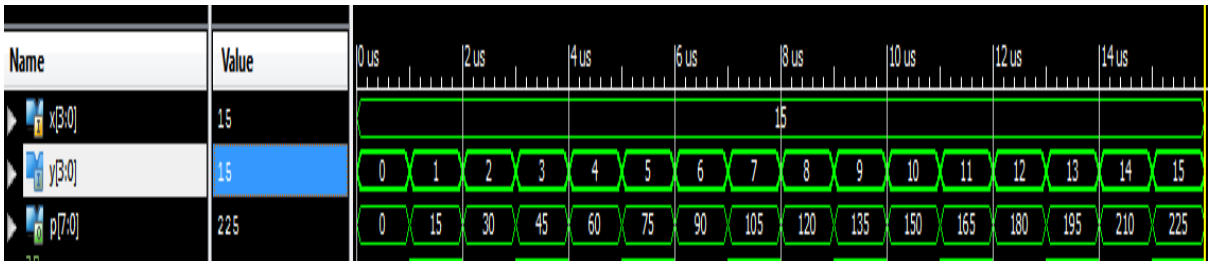

Fig 6..1: structural simulation result of 4X4 Array multiplier

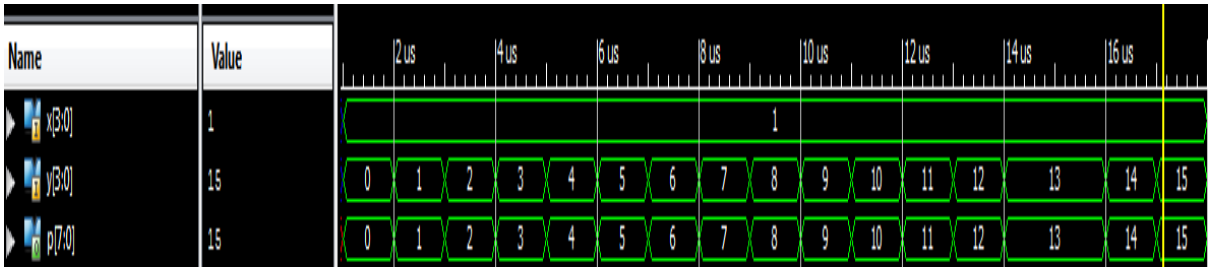

Fig-6.2: structural simulation result of 4X4 Braun multiplier

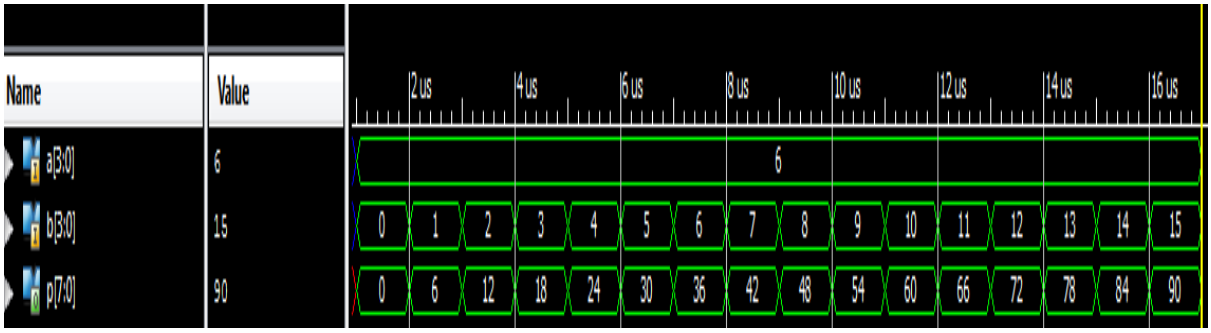

Fig-6.3: structural simulation result of 4X4 Wallace multiplier

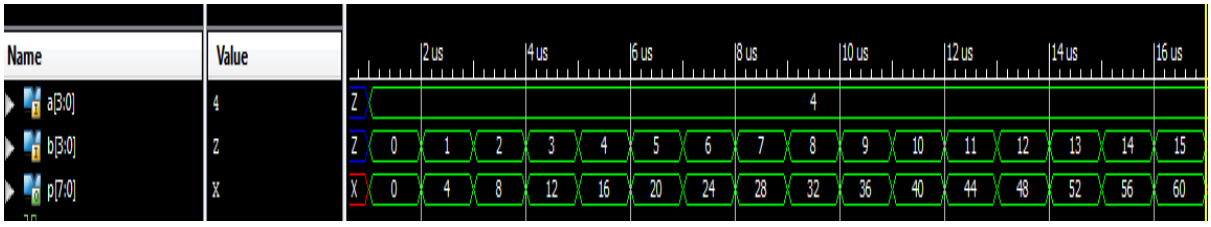

Fig-6.4: structural simulation result of 4X4 Baugh wooley multiplier

7. COMPARISION OF DIFFERENT MULTIPLIERS

PARAMETER	ARRAY MULTIPLIER	BRAUN MULTIPLIER	BOUGH WOOLEY MULTIPLIER	WALLACE MULTIPLIER	AVAILABILITY
NO OF SLICES	17	18	20	18	4656
SLICE INPUT LUTs	30	31	35	32	9312
TIME DELAY	13.726ns	11.676ns	10.734ns	11.701ns	-
POWER	80.98	80.78	80.98	80.98	-
LOGIC	32	31	35	32	9312
SIGNALS	38	39	42	40	-

Table-7.1: Comparison of different multipliers

8. CONCLUSION

Extensive comparisons with four different multipliers is performed and compared in area, power consumption and timing and physical design is implemented. From table 7.1, we can conclude that

1) Array multiplier is preferred when area is critical factor.

2) When speed is required, Bough-Woolley multiplier is preferred.

3) Braun multiplier is preferred when power consumption is critical factor

9. REFERENCES

[1] A study performance comparison of digital multipliers using 22nm strained silicon technology by S.Sabeetha, J.Ajayam, S, Shriram. K.Vivek, V.Rajesh.

[2]D. Kudithipudi, Eugene John, "Implementation of Low Power Digital Multipliers Using 10 Transistor Adder Blocks", Journal of Low Power Electronics, vol.1, pp.1-11, 2005.

[3] I.S.Ab u-Khater, A .Bellaouar , and M.I.Elmasry , Circuit techniques for CMOS low power high-performance multipliers. IEEE Journal of Solid State Circuits, Vol.31, pp.1535–1546. 1996.

[4]G.-K.Ma and F.J.T aylor, Multiplier policies for digital signal processing. IEEE ASSP Magazine (1990), pp.6–19.

[5] T.K.Calla way and E.E.Swartzlander, Jr., Power delay characteristics of CMOS multipliers. Proceedings of the 13th International Symposium on Computer Arithmetic (1997), pp.26–32.

[6] Y. C. Lim, "Single-precision multiplier with reduced circuit complexity for signal processing applications," IEEE Trans. Comput., vol. 41, no.10, pp. 1333–1336, Oct. 1992.

[7] S. S. Kidambi, F. El-Guibaly, and A. Antoniou, "Area- efficient multipliers for digital signal processing applications," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 43, no. 2, pp. 90–95, Feb. 1996.

[8] Sumita Vaidya and Deepak Dandekar, "Delay - Power Performance comparison of Multipliers in VLSI Circuit Design", International Journal of Computer Networks & Communications (IJCNC), Vol.2, No.4, pp. 47-56, July 2010.