

Performance Analysis of Multipliers in VLSI Design

Lunius Hepsiba P¹, Thangam T²

P.G. Student (ME - VLSI Design), PSNA College of Engineering and Technology, Dindigul, Tamilnadu, India¹

Associate Professor, Dept. of ECE, PSNA College of Engineering and Technology, Dindigul, Tamilnadu, India²

ABSTRACT: Multipliers play an important role in image Processing and some other applications. One of the basic arithmetic operations is Multiplication. It requires more hardware resources and process time than other arithmetic operations. This paper presents comparative study of different multipliers. The speed of the system depends upon speed of the multiplier, so that a high speed multiplier is necessary. Various techniques have been proposed to design high speed multipliers, which offer low power and lesser area. Thus making them suitable for various high speed, low power compact VLSI implementations.

KEYWORDS: Arithmetic operations, Multipliers, Low Power, High Speed.

I. INTRODUCTION

Now-a-days, the arithmetic operation is the basic operation for all the innovative processes. The arithmetic operation has addition, subtraction, multiplication and division. Among these multiplication and addition is the frequently using operations. Multiplication is to add an integer to a specified number of times by itself. Multiplication is done with two numbers, multiplicand and multiplier. The Multiplication is done by adding multiplicand to itself for a number of times, as specified by another number called Multiplier. The output generated is the product value of the two binary numbers. The multiplicand is multiplied by each digit of the multiplier beginning with the LSD. Intermediate results are called as the partial products. The final product is determined by summation of all the partial-products. Multiplication involves three main steps. They are Partial product generation, Partial product reduction and Final addition. For the multiplication of an N-bit multiplicand with an M-bit multiplier, M partial products are generated and product formed is N+ Mbits long.

Digital circuit design uses digital multipliers, which are fast, reliable and have efficient components. They can do many operations with less number of components. Depending upon the arrangement of the components, the various multipliers are available. Particular multiplier architecture is chosen based on the application. In many DSP algorithms, the multiplier has critical path and determines the performance of the algorithm. The speed of multiplication operation is of great importance in DSP as well as in general processor. In the past, multiplication was implemented generally with a sequence of addition and shifting operations. There are many algorithms proposed to perform multiplication, each offers different benefits and having trade off in terms of speed, circuit complexity, and area and power consumption.

The objective of good multiplier to provide a physically compact with high speed and low power consumption. Being a core part of arithmetic processing unit multipliers are in high demand on multipliers speed and low power consumption. To reduce significant power consumption of multiplier design, it is a good direction to reduce number of operations thereby reducing a dynamic power which is a major part of total power dissipation. In the past considerable effort were put into designing multiplier in VLSI design.

Paper describes as follows. Section II describes the implementation multipliers. The two different multipliers describes in Section III. Section IV, V, VI presents three types of multipliers. The experimental results are given in the section VII. Finally, Section VIII presents conclusion.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

II. RELATED WORK

The shift and add algorithm has been implemented in traditional to design the multiplier. However, this is not suitable for delay and for VLSI implementation. The low power and high speed multiplier can be implemented with different logic style. The three important parameters for VLSI design are power, area and delay. Each logic style has its own advantages in terms of speed, area and power. The demand for high speed processing has been increasing as a result of expanding image processing and signal processing applications.

Higher throughput arithmetic operations are important to achieve the desired performance in many real-time signal and image processing applications. Multiplication is one of the arithmetic operations. Reducing the time delay and power consumption are very essential requirements for many applications. This paper presents the study of different multipliers. There are number of techniques that to perform binary multiplication. The choice is based upon factors such as latency, throughput, area, and design complexity.

More efficient approach uses array/tree of adders to sum the partial products terms. Array multiplier, Booth Multiplier and Wallace Tree multipliers are the different types of binary multipliers which are implemented using VLSI design.

III. TYPES OF MULTIPLIER

A. Serial Multiplier

Along with area and power, delay can also be tolerated by the serial multiplier. The serial multiplier circuit uses one adder to add the $M \times N$ partial products terms. For M - Multiplicand bits and N - Multiplier bits which are inputs have to be arranged in a specified manner which will synchronize with the circuit. The length of multiplicand and multiplier used to arrange the inputs. The two clocks are used, one to clock the data and one for the reset. A first order approximation of the delay is $O(m,n)$. With this circuit arrangement the delay is given as $D = [(m+1)n + 1]$. The below is the serial multiplier architecture diagram Fig.1.

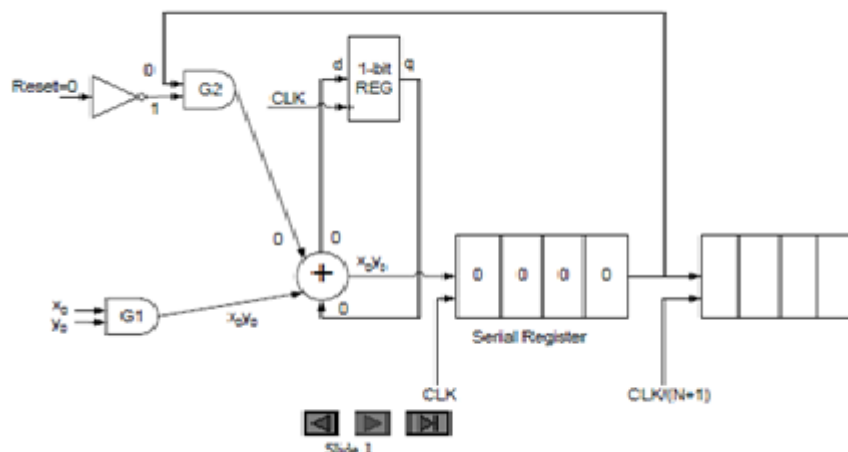


FIG.1. SERIAL MULTIPLIER

B. Serial/Parallel Multiplier

Another type of multiplier is the parallel multiplier. The general architecture of the serial/parallel multiplier is given below figure Fig.2. Among the two inputs, one input is given to the circuit in parallel and the other input is given in serial. For each cycle, n partial products are formed. Each cycle does the addition of one column of the multiplication table of $M \times N$ partial product terms. The final results are stored in the output register after $N+M$ cycles. While the area required is $N-1$ for $M=N$.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

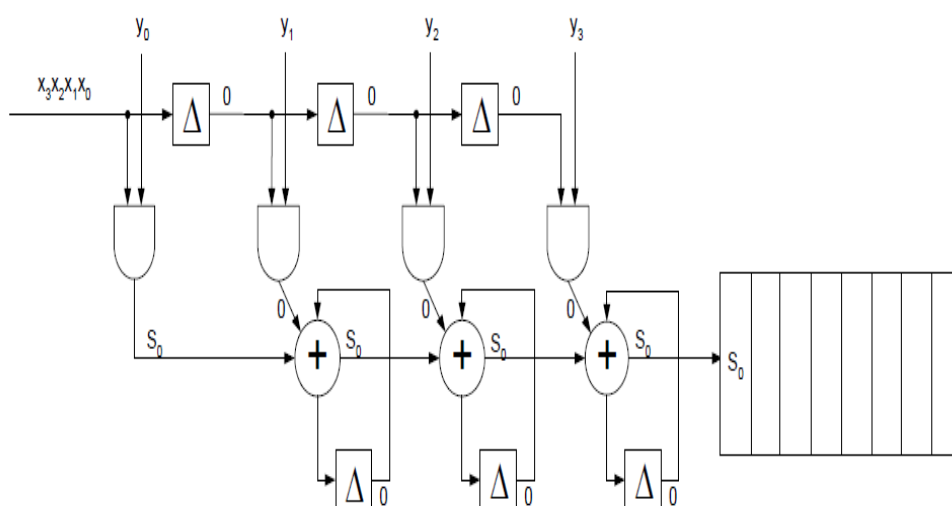


FIG.2. PARALLEL MULTIPLIER

IV. ARRAY MULTIPLIER

An array multiplier is a combinational circuit which is well-known for its regular structure. This multiplier is used for the multiplication of M bit multiplicand and N bit multiplier. The multiplicand and multiplier are binary numbers. This is done by employing an array of full adders and half adders.

Basic Multiplier circuit is based on adding and shifting algorithm. Each partial product is generated by the multiplication of the multiplicand with the one multiplier bit. According to their bit orders, the partial products are shifted and then those partial products are added. The addition can be performed with normal carry propagate adder. $N-1$ adders are required where N is the multiplier length. This array is used for the simultaneous addition of the various generated product terms. To form the various product terms, an array of gates is used before the adder array. These formed product terms are called partial product terms.

Let us consider 2×2 bit multiplication with A and B being the multiplicand and the multiplier respectively. Assuming $A = a(1) \times a(0)$ and $B = b(1) \times b(0)$, the various bits of the final product term P can be written as:-

$$P(0) = a(0) \times b(0)$$

$$P(1) = \text{Sum1} = a(1) \times b(0) + b(1) \times a(0)$$

$$P(2) = \text{Sum2} = a(1) \times b(1) + C1 \text{ where } C1 \text{ is the carry of the } P(1) \text{ term.}$$

$$P(3) = C2 \text{ where } C2 \text{ is the carry of the } P(2) \text{ term.}$$

For the above multiplication, four AND gates are required to form the various product terms like $a(0) \times b(0)$, $a(1) \times b(0)$, $b(1) \times a(0)$, $a(1) \times b(1)$. The Adders are used to calculate the product terms and carry as shown in the equation.

The hardware requirement for an $M \times N$ bit array multiplier contains $(M \times N)$ AND gates, $(M-1) \times N$ Adders in which N HA (Half Adders) and $(M-2) \times N$ FA (full adders).

Consider 4×4 multiplications, say $A = A_3 A_2 A_1 A_0$ and $B = B_3 B_2 B_1 B_0$. The output line for this multiplication is $P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0$. Using the fundamental of Array Multiplication, taking partial product addition is carried out in carry save form. The circuit diagram of the array multiplier is shown in the below figure Fig.3.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

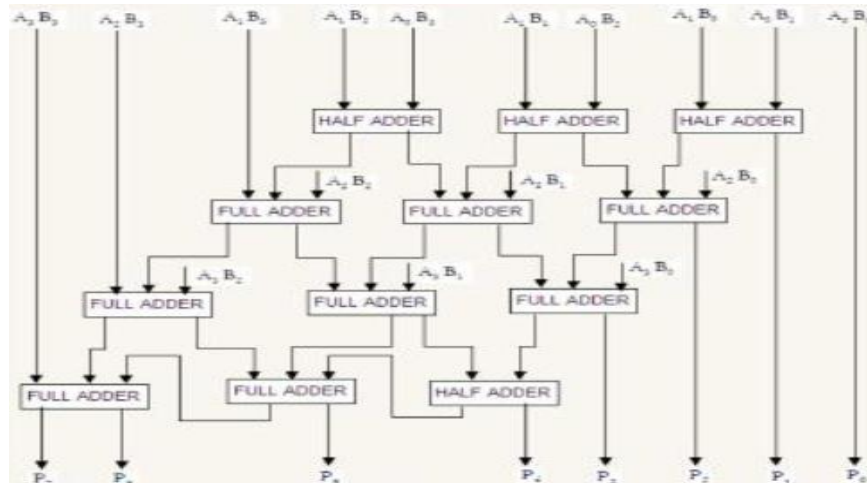


FIG.3. ARRAY MULTIPLIER

V. WALLACE TREE MULTIPLIER

Use several popular and well-known schemes, with the objective of improving the speed of the parallel multiplier. Wallace introduced a very important iterative realization of parallel multiplier. This advantage is that this multipliers with larger than 16 bits.

In Wallace tree architecture, the bits of all partial products are added together by a counters in parallel without propagating any carries. Then some counters reduces this matrix, until a two-row matrix is generated. The most common counter used is the 3 to 2 counter which is also act as a Full Adder. The final results are added using usually carry propagate adder. The advantage of Wallace tree is speed because the addition of partial products is now $O(\log N)$. As seen from the block diagram partial products are added in Wallace tree block. The addition result is the final product. The sum and carry are added in the final fast adder (CRA).

Using this method, a three step process is used to multiply two numbers; the bit products are formed, the product matrix is reduced to a two row matrix where sum of the row equals the sum of bit products, and the resulting rows are added with a adder to produce a final product. Three bits are passed to a one bit adder which is called a three input Wallace tree circuit, and the output signal is supplied to the next stage full adder of the same bit, and the carry output signal thereof is passed to the next stage with the same number of bit, and the carry output signal thereof is supplied to the next stage of the full adder located at a one bit in higher position. Wallace tree is of a carry-save adders arranged as shown in figure FIG.3. A carry save adder consists of full adders as like the ripple carry adders, but the carry output from each bit is brought to form second result rather than wired to the next Most Significant Bit (MSB). The carry is saved to be combined with the sum later. In the Wallace tree method, the circuit diagram is more complex but speed is high.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

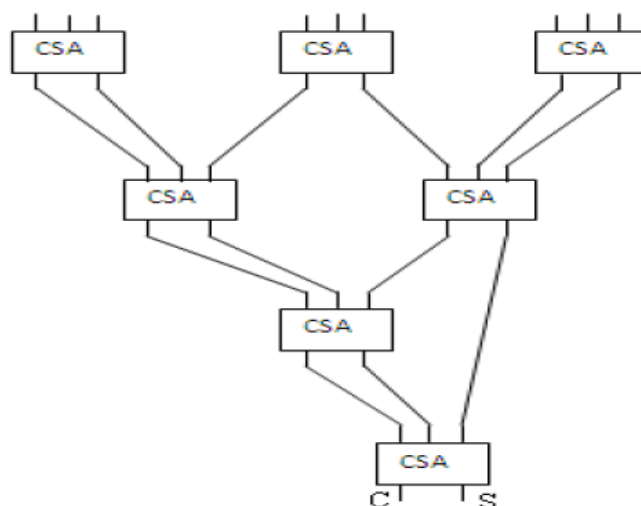


FIG.4. WALLACE TREE MULTIPLIER

VI. BOOTH MULTIPLIER

Another type of multiplier is done by reducing the number of partial products. The Booth multiplier is one such multiplier; it fetches three bits at a time to reduce the number of partial products. The three bits are taken, among them the first two bits are from the present pair and the third bit is from the higher order bit of an adjacent lower order. The three bits are converted by Booth logic into a set of five control signals used in the adder to control the operations performed by the adder.

It is an efficient algorithm for multiplication of unsigned/signed-numbers, which treats both positive and negative numbers uniformly. Each multiplier bit generates multiple of the multiplicand to be added to the partial product. The large number of multiplicands have to be added to do the multiplication for large numbers. In this case the delay of multiplier is determined by the additions to be performed. If there is a way to reduce the number of the additions, the performance will be good.

Booth algorithm is a method that will reduce the multiplication of multiplicand. To represent a given numbers, a higher radix leads to fewer digits. A K-bit number can be interpreted as K/2-digit radix-4 number, a K/3-digit radix-8 number, and so on, it can deal with more than a bit of the multiplier for each cycle by high radix multiplication.

Booth algorithm takes advantage of the fact that an adder subtractor is nearly as fast and small as a simple adder. From the basics, it can be proved that the addition or subtraction operation can be skipped. If the consecutive bits are same then addition or subtraction can be skipped. Thus in most of the cases the delay with booth multiplication are smaller than that with array multiplier. However the performance of booth multiplier is input data dependent. The method of booth algorithm reduces the numbers of adders and hence the delay required to produce the partial product by three bits at a time. The high performance of booth multiplier comes with the drawback of power consumption.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

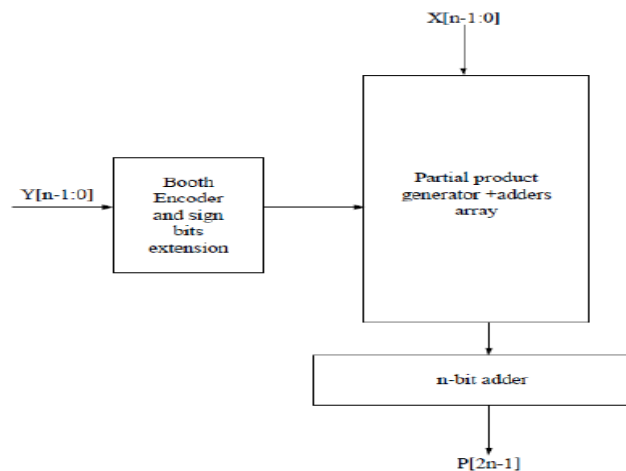
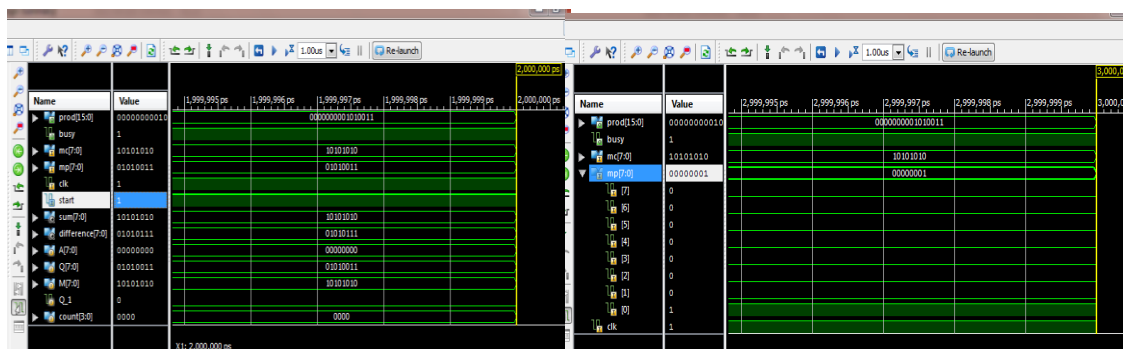


FIG.5. BOOTH MULTIPLIER

VII. SIMULATION RESULTS

Different multipliers architecture for the required functionality were discussed in the previous chapters. Now this chapter deals with the simulation and synthesis results of the different multiplier architecture. Here **ISim** tool is used to simulate the design and checks the functionality of the design. All multipliers are designed in **VHDL** (Hardware Description Language). Logic synthesis and simulation was done using EDA (Electronic Design Automation) tool in **XilinxISE14.2i**. The below simulation results are tested for 8×8 Array Multiplier, Wallace Tree Multiplier and Booth Multiplier.



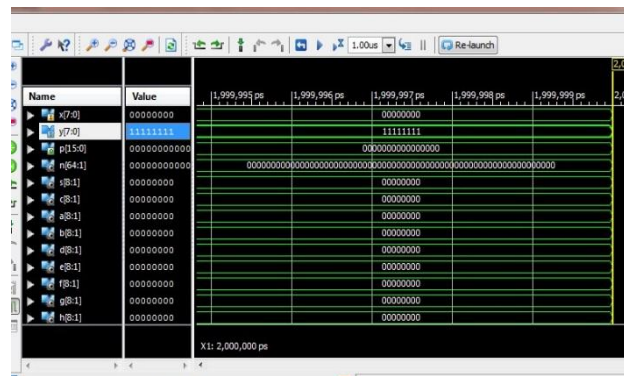
A

B

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015



C

FIG. 6. ABOVE ARE THE SIMULATION RESULTS OF THE VARIOUS MULTIPLIERS. (A) SIMULATED RESULT OF ARRAY MULTIPLIER, (B). SIMULATED RESULT OF BOOTH MULTIPLIER AND (C). SIMULATED RESULT OF WALLACE TREE MULTIPLIER

The parameters such as speed, circuit complexity, and area and power consumption are taken into consideration for comparison of different multipliers.

	Array Multiplier	Booth Multiplier	Wallace Tree Multiplier
Speed	LOW	HIGH	HIGH
Circuit Complexity	SIMPLE	MORE	MOST
Area	LARGE	SMALL	MEDIUM
Power Consumption	MOST	MORE	LESS

TABLE.1. COMPARISON TABLE OF VARIOUS MULTIPLIERS SUCH AS ARRAY, BOOTH AND WALLACE TREE MULTIPLIERS BY ITS PERFORMANCE LIKE SPEED, CIRCUIT COMPLEXITY, AREA AND POWER CONSUMPTION.

VIII. CONCLUSION

Thus it is concluded that among the above discussed multipliers, Booth Multiplier is superior. This is discussed with respect to area, power consumption, and delay and circuit complexity. The Array Multiplier consumes more power consumption and more number of components are required. But for the array multiplier delay is larger than Wallace Tree Multiplier. Hence for low power and less delay requirement Booth multiplier is suggested. These results were obtained after synthesizing individual multipliers in Xilinx ISE 14.2. All comparisons are based on the synthesis reports keeping one common base for comparison.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2015

REFERENCES

- [1] Byung-Gook Park, Jong Duk Lee, Kyong Hwa Leez andYong Jin Yoony, "Application of Dynamic Pass Transistor Logic to 8 bit Multiplier," *Journal of Korean Physical Society*, Vol. 38, Issue No. 3, pp. 220-223, March. 2001.
- [2] Law C. F, Rofail S.S andYeo K. S, "A Low Power 16×16 bit Parallel Multiplier Utilizing Pass Transistor Logic," *IEEE Journal of Solid State Circuits.*, Vol. 34, Issue No. 10, pp. 1395-1399, October 1999.
- [3] Deepak Dandekar, Sumit Vaidya,"Delay – Power Performance Comparison of Multipliers in VLSI Circuit Design", *International Journal of Computer Science and Communication*, Vol. 2, Issue No. 4, July, 2010.
- [4] Chandrasekhran, Nikolic, andRabey, "Digital Integrated Circuits: A Design prespective," 2nd Edition, *Prentice Hall*, pp. 586-594, July 2003.
- [5] Oscar T. C. Chen, Sandy Wang, and Yi-Wen Wu, "Minimization of Switching Activities of Partial Products for Designing Low Power Multipliers," *IEEE Transtactions On VLSI Systems*, Vol. 11, Issue No. 3, pp. 418-433, June 2003.
- [6] Jyoti Sharma and Sachin Kumar, "Speed Comparison of Various Multipliers", *International Journal of Engineering and Management Research*, Vol.4, Issue No. 3, pp. 244-247, June 2014.
- [7] L. Sriraman, T.N. Prabakar, "Design and Implementation of Two Variable Multiplier using KCM and Vedic Mathematics," *2012 first International Conference on Recent Advances in Information Technology IEEE*, pp. 782-787, March, 2012.
- [8] Anil Kumar Sahani and Mohammed Hasmat Ali, "Study, Implementation and Comparison of Different Multipliers Based on Array, KCM, Vedic Mathematics using EDA Tools", *International Journal of Scientific Research Publications*, Vol. 3, Issue No.6, pp. 2250-3153, June 2013.