ISSN XXXX XXXX © 2017 IJESC

Research Article Volume 7 Issue No.12

Survey on Multipliers Utilized in VLSI

Prajna .K .B[1], Savithri Hande[2]
Assistant professor[1, 2]
Department of ECE
NMIT, Bangalore, India

Abstract:
Low Power VLSI circuit has become significant criterion for designing many energy efficient electronic designs for high performance and portable devices. Majority of electronic application especially DSP applications the critical operations are the multiplication. Multiplier plays a major role in many hardware computations. Multiplication is one of the basic arithmetic operations and it requires substantially more hardware resources and processing time than addition and subtraction. For proper functioning of a system, multiplier has to be fast and less area in hardware. 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 common multiplication method is “add and shift” algorithm. In multipliers number of partial products to be added is the main parameter that determines the performance of the multiplier. Among these, the Binary multiplier and Array multiplier are the basic one. To reduce the number of partial products to be added, 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. Vedic Multiplier reduces the steps and time consumed in computation of partial products. Baugh Wooley Multiplier is best suited multiplier for signed multiplication. In this paper, a brief study of different multipliers is done.

Keywords: Array Multiplier, Baugh Wooley Vedic Multiplier, Booth Multiplier, Braun Multiplier, Carry save, Dadda Multiplier, Wallace Tree Multiplier;

I. INTRODUCTION

Multipliers play an important role in digital signal processing and various other applications. In high performance systems such as microprocessor, DSP etc., addition and multiplication of two binary numbers is fundamental and most often used arithmetic operations. More than 70% instructions in microprocessor and most of DSP algorithms perform addition and multiplication. So, this operation dominates execution time.

The demand of high speed processing has been increasing as a result of expanding computer and signal processing applications. Low power consumption is an important issue in multiplier design. By reducing the number of operation we can reduce significant power consumption thereby reducing dynamic power which is a major part of total power consumption. The number of addition operation is performed by a multiplier concept.

II. VARIOUS MULTIPLIERS

There are various multipliers used for computation purpose, the multipliers discussed in this paper are Binary multiplier, Array multiplier, Vedic multiplier, Booth multiplier, Carry save Multiplier, Dadda Multiplier, Baugh Wooley Multiplier, Braun multiplier.

Binary Multiplier

A binary multiplier is employed in digital electronic circuits to perform the multiplication of two binary numbers. The binary numbers are 0 and 1, hence the binary multiplication will be easy. Referring to the multiplication done below in FIGURE 1, partial products are created for every bit during multiplication. All the partial products are summed up to obtain the final product. During the partial product multiplication, the partial product obtained will be zero when the multiplier bit is zero and the partial product obtained will be the multiplicand when the multiplier bit is 1.In every consecutive step the partial products are shifted left, then the summing of all the partial products is carried out. Therefore binary multiplication requires N – shifts and N – adds for the multiplication of N –bit binary number.

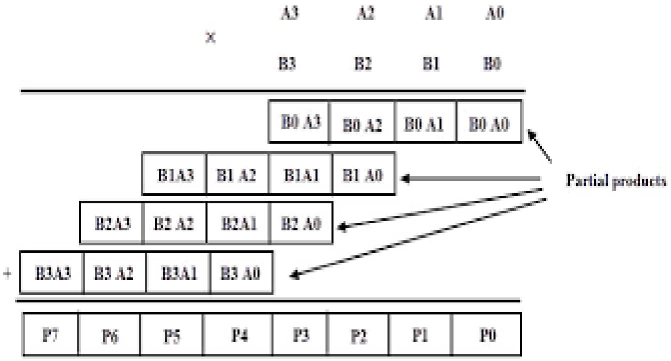
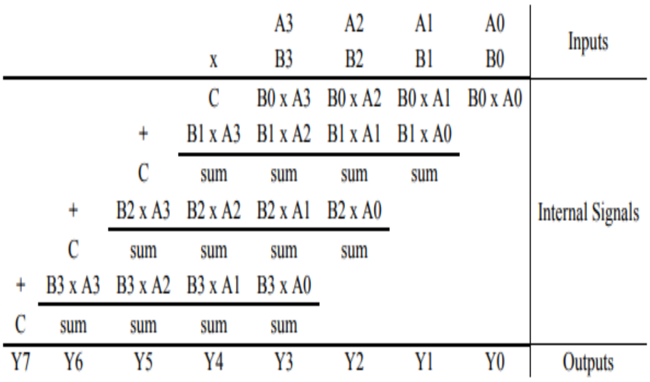


Figure.1. Structure of 4x4 binary multiplier

Array Multiplier

Multiplier circuit is based on add and shift algorithm. Each partial product is generated by the multiplication of the multiplicand with one multiplier bit. The partial product are shifted according to their bit orders and then added. The addition can be performed with normal half adder and full adder or carry propagate adder. N-1 adders are required where N is the multiplier length.



				A3	A2	A1	A0	Inputs
			x	B3	B2	B1	B0	
			C	B0 x A3	B0 x A2	B0 x A1	B0 x A0	Internal Signals
		+	B1 x A3	B1 x A2	B1 x A1	B1 x A0		
		C	sum	sum	sum	sum		
	+	B2 x A3	B2 x A2	B2 x A1	B2 x A0			
	C	sum	sum	sum	sum			
+	B3 x A3	B3 x A2	B3 x A1	B3 x A0				
C	sum	sum	sum	sum				
Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	Outputs

Figure.2. Structure of 4 x 4 array multiplier

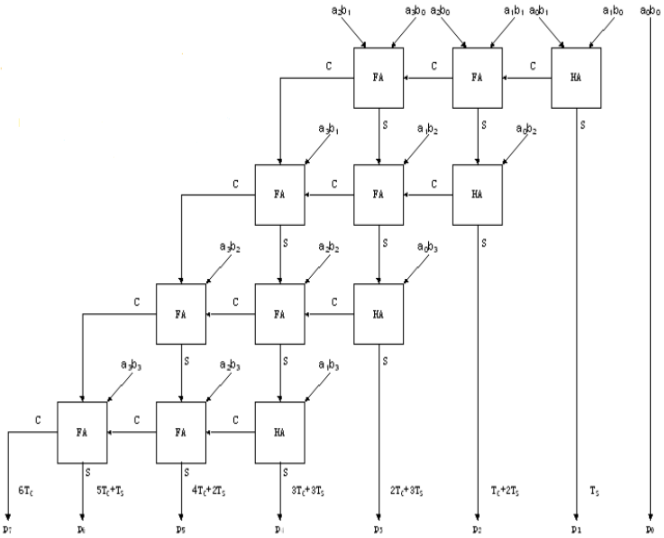


Figure.3. 4 x 4 array multiplier

FIGURE 2 and FIGURE 3 show Array Multiplier shows 4x4 multiplier. For 4 x 4 Multiplier it requires 16 AND gates, 4 HA's, 8 FA's (total 12 adders). Major limitation of array multiplier is its size. As operand sizes increase, arrays grow in size at a rate equal to the square of the operand size, hence speed of multiplier reduces.

Vedic Multiplier

Vedic Mathematics hails from the ancient Indian scriptures called "*Vedas"* or the source of knowledge. This system of computation covers all forms of mathematics, be it geometry, trigonometry or algebra. Vedic mathematics is part of four Vedas (books of wisdom). It is part of Sthapatya- Veda (book on civil engineering and architecture), which is an upa-veda (supplement) of Atharva Veda.

It covers explanation of several modern mathematical terms including arithmetic, geometry (plane, co-ordinate), tri gonometry, quadratic equations, factorization and even calculus. In Vedic mathematics there are 16 sutras (formulae) and 16 Upa sutras (sub formulae). Among sutras three are used for multiplication. Urdhava Tiryakbhyam is a Sanskrit word which means vertically and crosswire in English. The method is a general multiplication formula applicable to all cases of multiplication. It is based on a novel concept through which all partial products are generated concurrently. FIGURE 4 demonstrates a 4 x 4 binary multiplication using this method. The method can be generalized for any N x N bit multiplication

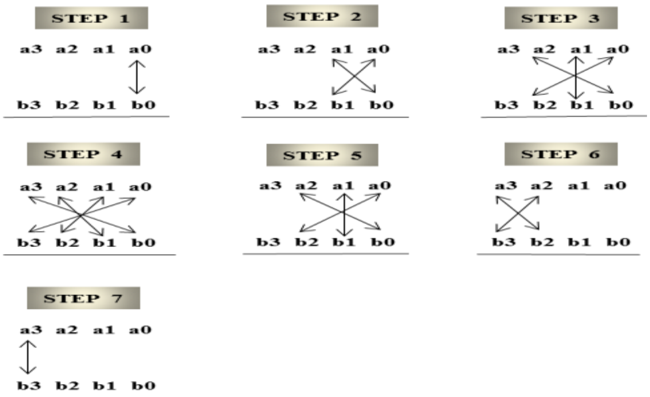


Figure.4. Multiplication of two 4 bit numbers using urdhava tiryakbhyam method

The line diagram in FIGURE 4, illustrates the algorithm for multiplying two 4-bit binary numbers. The procedure is divided into 7 steps and each step generates partial products. Initially as shown in step 1the least significant bit (LSB) of the multiplier is multiplied with least significant bit of the multiplicand (vertical multiplication). This result forms the LSB of the product. In step 2 next higher bit of the multiplier is multiplied with the LSB of the multiplicand and the LSB of the multiplier is multiplied with the next higher bit of the multiplicand (crosswire multiplication). These two partial products are added and the LSB of the sum is the next higher bit of the final product and the remaining bits are carried to the next step.

In general, taking 2 numbers $a_3a_2a_1a_0$ *and* $b_3b_2b_1b_0$, the steps can be explained as

r0=a0b0. (1)
c1r1=a1b0+a0b1. (2)
c2r2=c1+a2b0+a1b1 + a0b2. (3)
c3r3=c2+a3b0+a2b1 + a1b2 + a0b3. (4)
c4r4=c3+a3b1+a2b2 + a1b3. (5)
c5r5=c4+a3b2+a2b3. (6)
c6r6=c5+a3b3 (7)

The hardware architecture of 4×4 Vedic multiplier is as shown in FIGURE 5

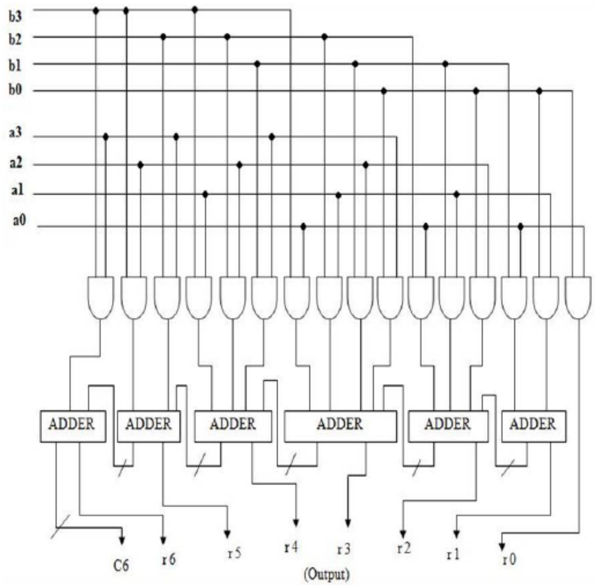


Figure.5. Hardware architecture of 4 x 4 multiplier

This type of multiplier is independent of the clock frequency of the processor because the partial products and their sums are calculated in parallel. The net advantage is that it reduces the need of microprocessors to operate at increasingly higher clock frequencies. Booth Multiplier

Booth Multiplier

The Booth algorithm enhances the speed of multiplication and it overcomes the limitations of array multiplier. The number of addition is reduced and positive numbers and negative numbers are treated uniformly. The number of partial products obtained from multiplication is reduced by adapting booth algorithm. The multiplication of two signed binary numbers is accomplished by adapting 2's complement method. The booth algorithm is represented in the flowchart given below.

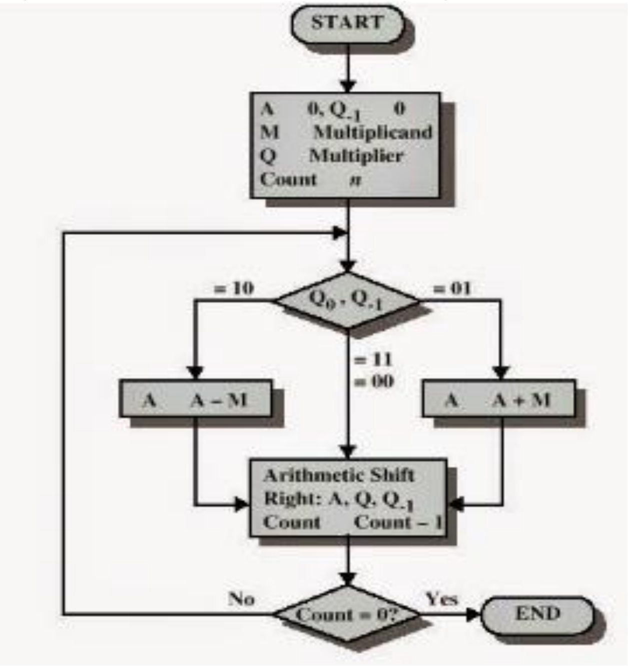


Figure.6. Flow diagram of booth multiplier

Booth multiplier is illustrated by considering Multiplicand (M)= 0101 and Multiplier (Q)=0100.

Table.1. Booth algorithm table

Multiplicand (M) = 0101 (5) Multiplier (Q)= 0100 (4)				
Steps	A	Q	Q-1	Operation
	0000	0100	0	Initial
Step 1	0000	0010	0	Shift right
Step 2	0000	0001	0	Shift right
Step 3	1011	0001	0	A-M and shift it to A
	1101	1000	1	Shift right
Step 4	0010	1000	1	A+M and shift it to A
	0001	0100	0	Shift right
00010100=20				

Booth multiplier delay is depended on the number of additions. The problem of reduced speed of array multiplier is overcome by using booth multiplier. Due to its low power consumption, booth multiplier is adapted in digital circuits.

Carry save Multiplier

FIGURE 7, shows a 4x4 multiplier implemented using Carry-Save method. Unlike the normal array multiplier, in Carry-Save multiplier the output carry bits are propagated diagonally downwards, instead of to the right. This design will require an extra adder called the vector-merging adder in order to achieve the final result. It is named so because the carry bits of each stage are saved to be propagated to the next adder rather than immediate sideways propagation. For this multiplier there is an increase in the number of transistors and hence the area occupied also increases.

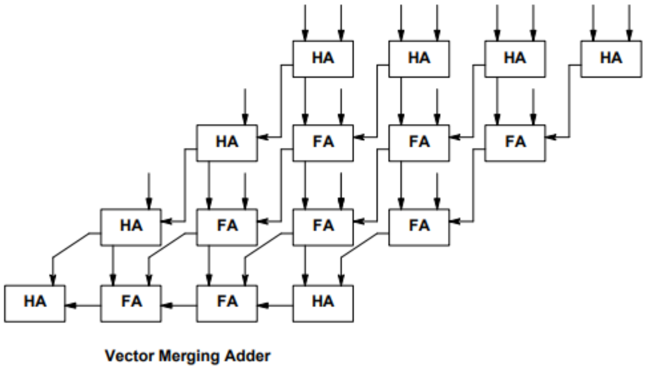


Figure.7. Structure of 4x 4 carry save multiplier

Wallace Tree Multiplier

Wallace is a tree multiplier designed for minimum propagation delay. It is implemented by adders using parallel multiplication resulting in less delay. Wallace tree sums up same weight of three bits and produces output which is said to be compressors.

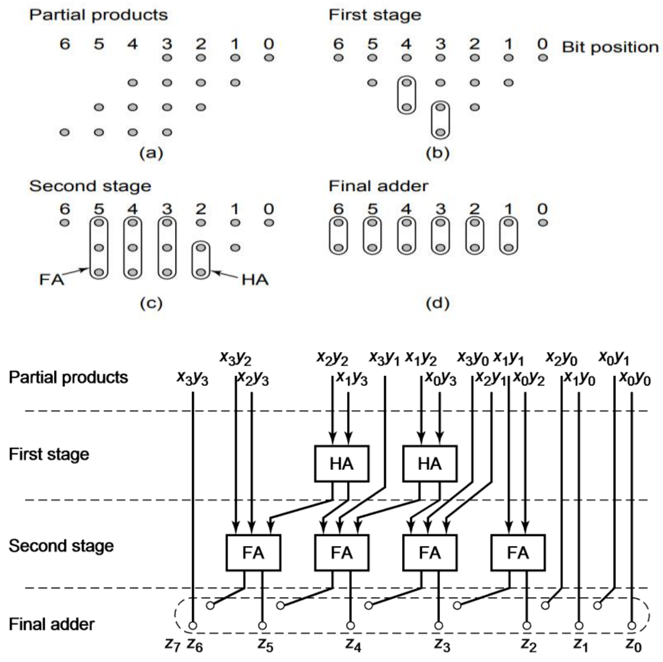

Partial products
First stage
6 5 4 3 2 1 0
Bit position
(a)
(b)
Second stage
Final adder
FA
HA
(c)
(d)
Partial products
First stage
Second stage
Final adder
x_3y_3 x_3y_2 x_2y_3 x_2y_2 x_1y_3 x_3y_1 x_1y_2 x_0y_3 x_3y_0 x_1y_1 x_2y_1 x_0y_2 x_2y_0 x_1y_0 x_0y_1 x_0y_0
HA
FA
z_7 z_6 z_5 z_4 z_3 z_2 z_1 z_0

Figure.8. Structure of wallece tree multiplier

Dadda Multiplier

Dadda multipliers utilize less of number of gates. Dadda multipliers mainly focus on lessening of partial product stage, yet it will accomplish a more ideal final product. The DADDA multiplier minimizes the number of adder stages required to perform the summation of partial products. This is accomplished by using full and half adders to reduce the number of rows in the matrix number of bits at each summation stage. The basic idea of DADDA multiplier depends on the underneath framework shape appeared in

FIGURE 9 A. The partial product is framed in the principal organize by AND stages which is delineated in FIGURE 9 B

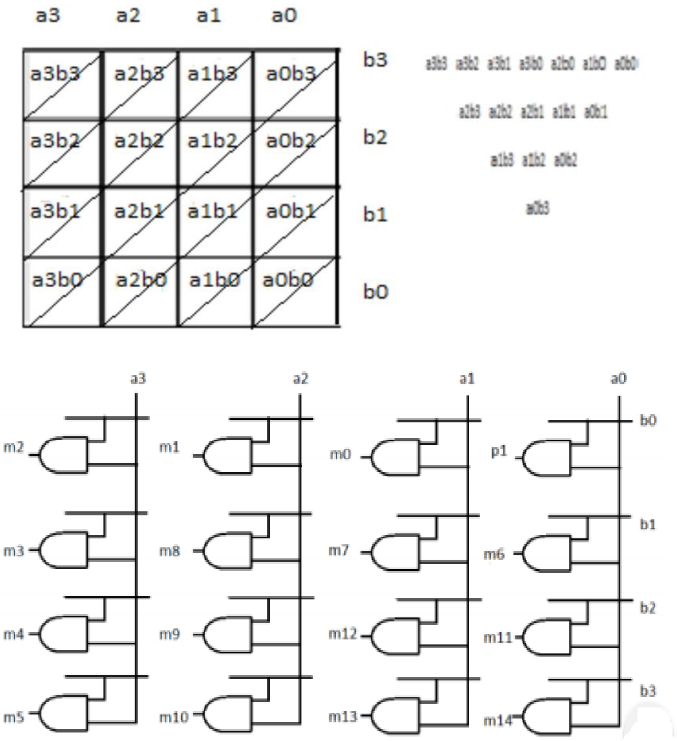


Figure .9. a&b Frame Work & And Stage of Dadda Multiplier

The general procedure can be separated into three stages. The initial step is to create the partial product matrix. A case of 4*4 partial product matrixes is given in Figure 10. Every halfway item is produced with an AND gate. Thus, 16 AND gates are required in a 4 by 4 multiplier. As a rule for N by N multiplier we require N*N AND gates. The second step is otherwise called the "decrease" step is to lessen the N lines of partial product bits to 2 bits that have a proportionate esteem. This progression has the most of the delay in a multiplier and here we concentrated on enhancing this deferral. The third step is to utilize adder (full adder or carry skip adder (CSA or carry propagate adder) to include the 2 rows and acquire their aggregate which is the result of the two information operands.

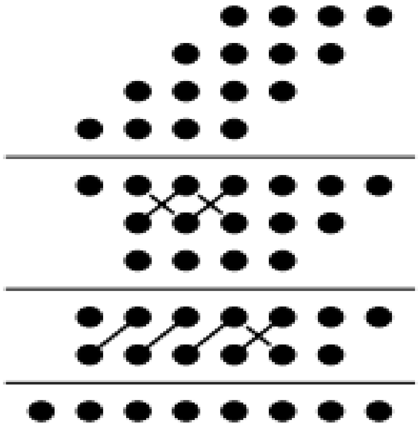


Figure.10. General Structure of 4 x4 multiplier

Baugh Wooley Multiplier

In signed multiplication the length of the partial products and the number of partial products will be very high. So an algorithm was introduced for signed multiplication called as Baugh Wooley algorithm. The Baugh-Wooley multiplication is one amongst the cost-effective ways to handle the sign bits. This method has been developed so as to style regular multipliers, suited to 2's compliment numbers. It uses parallel products to complement multiplication and adjusts the partial products to maximize the regularity of multiplication array when number is represented in two's complement form; sign of the number is embedded in Baugh-Wooley multiplier. This algorithm has the advantage that the sign of the partial product bits are always kept positive so that array addition techniques can be directly employed. In the two's complement multiplication, each partial product bit is the AND of a multiplier bit and a multiplicand bit, and the sign of the partial product bits are positive.

A signed number is represented by

$$X=-x_{n-1}2^{n-1}+\sum_{i=0}^{n-2}x_i2^i \qquad (8)$$

Baugh Wooley principle

$$XY=x_{n-1}y_{n-1}2^{2n-2}+\sum_{i=0}^{n-2}\sum_{j=0}^{n-2}x_iy_j2^{i+j}-\sum_{i=0}^{n-2}(x_{n-1}y_i+y_{n-1}x_i)2^{i+n-1}$$

$$x_i=1-\overline{x_i} \qquad y_i=1-\overline{y_i} \qquad (9)$$

$$XY=(x_{n-1}y_{n-1}-x_{n-1}-y_{n-1})2^{2n-2}+(x_{n-1}+y_{n-1})2^{n-1}+$$

$$\sum_{i=0}^{n-2}\sum_{j=0}^{n-2}x_iy_j2^{i+j}+\sum_{i=0}^{n-2}(x_{n-1}\overline{y_i}+y_{n-1}\overline{x_i})2^{i+n-1} \qquad (10)$$

$$XY=-2^{2n-1}+(x_{n-1}y_{n-1}+\overline{x_{n-1}}+\overline{y_{n-1}})2^{2n-2}+(x_{n-1}+y_{n-1})2^{n-1}+$$

$$\sum_{i=0}^{n-2}\sum_{j=0}^{n-2}x_iy_j2^{i+j}+\sum_{i=0}^{n-2}(x_{n-1}\overline{y_i}+y_{n-1}\overline{x_i})2^{i+n-1} \qquad (11)$$

Figure.11. Shows Baugh Wooley Multiplier Structure

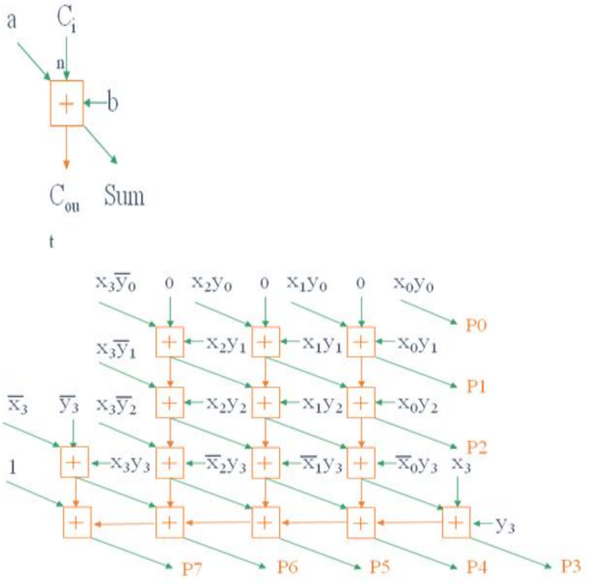


Figure.12. Baugh wooley multiplier structure

Braun Multiplier

Braun multiplier is a one of the parallel multipliers. It is also referred to as Carry Save Multiplier. Full adders and AND gates are present in the architecture of this multiplier. AND gates are connected in parallel, the partial products are obtained as result of it. Each partial product is summed up with the sum of previously produced partial products utilizing adders. Figure 12 shows the architecture of standard, 4 X 4 Braun multiplier. Generally, N X N Braun multipliers consists of n^2 AND gates and N (N-1) full adders. The drawback of this multiplier is that the number of components used in building the multiplier increases four times with the increase in the number of bits.

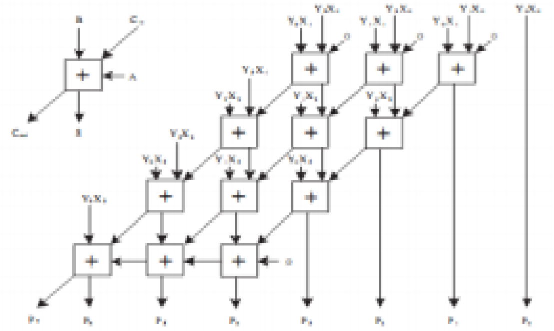


Figure.13. Structure of 4x4 braun multiplier

III. CONCLUSION

Various multipliers which are used in VLSI are discussed here. Each Multiplier is advantageous in specific field of application. Vedic Multiplier is suitable for all kinds of multiplication. We can choose the best multiplier according to our requirements.

IV. REFERENCES

[1]. S.P.Pohokar, R.S.Sisal, K.M.Gaikwad, M.M.Patil, Rushikesh Borse (2015) “Design and Implementation of 16x16 Multiplier using vedic mathematics” International Conference on Industrial Instrumentation and control (ICIC) College of Engineering Pune, India, May 28-30.

[2]. Ruchi Sharma (October 2012) “Analysis of Different Multiplier with Digital Filters Using VHDL Language” International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-2, Issue-1.

[3]. Vidhi Gupta, J. S. Ubhi “Analysis and Comparison of Various Parameters For Different Multiplier Designs” International Journal Of Scientific Research And Management Studies (IJSRMS) ISSN: 2349-3771Volume 1 Issue 4, Pg: 136-147.

[4]. M.Rohini, M.Madhumetha, S.Ranjith”An Survey on Adders & Multiplier For High Speed & Low Power Application” International Conference on Futuristic Trends in Computing and Communication (ICFTCC-2015) ISSN: 2348 – 8549, Page 55

[5]. Song, Leilei, and Keshab K. Parhi. (1998) "Low-energy digit-serial/parallel finite field multipliers." Journal of VLSI signal processing systems for signal, image and video technology 19, no. 2 149-166.

[6]. Baugh, Charles R., and Bruce A. Wooley. (1973) "A two's complement parallel array multiplication algorithm." IEEE Transactions on Computers 100, no. 12 ,1045-1047.

V. ACKNOWLEDGEMENT

We thank the library resources available in NMIT. We acknowledge our colleagues who helped us in providing information and guidelines in writing this paper. Finally we thank all our family members, friends and well wishers who have guided us.