COMPARATIVE ANALYSIS FOR HARDWARE CIRCUIT ARCHITECTURE OF WALLACE TREE MULTIPLIER

Dhaval R Gandhi
Electronics and Communication Department
Babaria Institute of Technology, BITS edu campus
Varnama, Vadodara, India
dhaval.rg@gmail.com

Nehal N Shah
Electronics and Communication Department
Sarvajanik College of Engineering and Technology
Surat, India
nehal.shah@scet.ac.in

***Abstract*— Multiplication is fundamental and significant operation of Electronic Circuits. Low power multipliers with high clock frequencies are widely used in today's digital signal processing. Currently demand is power efficient, high speed miniature system which leads to design circuits with transistor level optimization. Full adder circuit is basic block of multiplier. Transistor level optimization of basic building element directly results in reduction of delay and power. In this paper, the performance analysis of Wallace-tree multiplier architectures are carried out based on small size full adder circuits.**

***Index Terms*— Wallace tree multiplier, Transmission gate full adder, Zhuang full adder, GDI XNOR full adder**

I. INTRODUCTION

Fast multipliers are essential parts of digital signal processing systems. The speed of multiply operation is of great importance in digital signal processing as well as in the general purpose processors, especially since the media processing took off. In the past multiplication was generally implemented via a sequence of addition, subtraction, and shift operations. Multiplication can be considered as a series of repeated additions. The number to be added is the multiplicand, the number of times that it is added is the multiplier, and the result is the product. Each step of addition generates a partial product. In most computers, the operand usually contains the same number of bits. When the operands are interpreted as integers, the product is generally twice the length of operands in order to preserve the information content. This repeated addition method that is suggested by the arithmetic definition is slow that it is almost always replaced by an algorithm that makes use of positional representation. It is possible to decompose multipliers into two parts. The first part is dedicated to the generation of partial products, and the second one collects and adds them.

The basic multiplication principle is two folds i.e., evaluation of partial products and accumulation of the shifted partial products. It is performed by the successive additions of the columns of the shifted partial product matrix. The 'multiplier' is successfully shifted and gets the appropriate bit of the 'multiplicand'. The delayed, instance of the multiplicand must all be in the same column of the shifted partial product matrix. They are then added to form the product bit for the particular form. Multiplication is therefore a multi operand operation.

Power dissipation is recognized as a critical parameter in modern VLSI design field. To satisfy MOORE'S law and to produce consumer electronics goods power efficient, and reduce weight, low power VLSI design is necessary. Dynamic power dissipation which is the major part of total power dissipation is due to the charging and discharging capacitance in the circuit. The golden formula for calculation of dynamic power dissipation is $P_d = C_l V^2 f$. Power reduction can be achieved by various manners like reduction of output capacitance C_l, reduction of power supply voltage V, reduction of switching activity based on frequency f. In this paper concentration is on small size full adder circuits which leads to reduction in power and improve speed of operation. Section II describes motivation of work followed by Wallace tree multiplier and full adder circuits in section III. Section IV shows simulation results followed by conclusion in section V.

II. MOTIVATION

As the scale of integration keeps growing, more and more sophisticated signal processing systems are being implemented on a VLSI chip. These signal processing applications not only demand great computation capacity but also consume considerable amount of energy. While performance and Area remain to be the two major design tolls, power consumption has become a critical concern in today's VLSI system design. The need for low-power VLSI system arises from two main forces. First, with the steady growth of operating frequency and processing capacity per chip, large currents have to be delivered and the heat due to large power consumption must be removed by proper cooling techniques. Second, battery life in portable electronic devices is limited. Low power design directly leads to prolonged operation time in these portable devices.

Multiplication is a fundamental operation in most signal processing algorithms. Multipliers have large area, long latency and consume considerable power. Therefore low-power multiplier design has been an important era. There has

978-1-4799-0317-7/13/$31.00©2013 IEEE

been extensive work on low-power multipliers at technology, physical, circuit and logic levels. A system's performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest element in the system. Furthermore, it is generally the most area consuming. Hence, optimizing the speed and area of the multiplier is a major design issue. However, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas. As a result, a whole spectrum of multipliers with different area- speed constraints has been designed with fully parallel.

III. STANDARD MULTIPLIERS

Multiplication of two fixed point binary numbers is done with paper and pencil by a process of successive Shift and Add method. Simple multiplication [1] process can be illustrated with the numerical example as shown in equation 1.

$$P = \left(\sum_{j=0}^{M-1} y_j 2^j\right)\left(\sum_{i=0}^{N-1} x_i 2^i\right) = \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} x_i y_j 2^{i+j} \quad (1)$$

where 'x' is multiplier, 'y' is multiplicand and 'P' is final product. For 8 bit multiplicand represented by (y_7 - y_0) and 8 bit multiplier (x7 - x_0) process is shown in figure 1, where each multiplier bit is multiplying with every multiplicand bit which generates 8 rows of partial products.

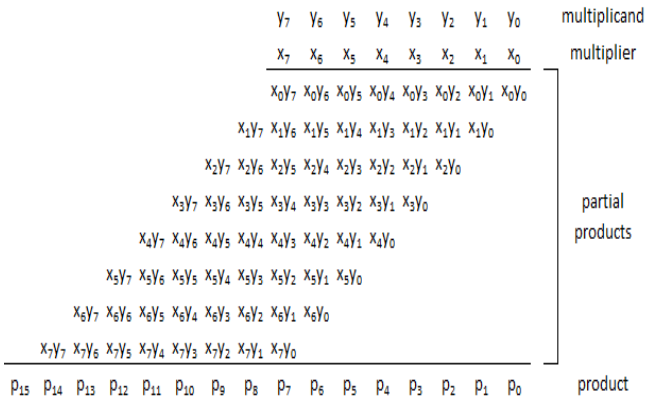

Fig. 1 simple multiplication method

The process consists of looking at successive bits of multiplier, starting from least significant bit. If the multiplier is a 1, the, multiplicand copied down; otherwise, zeros are copied down. The numbers copied down in successive lines are shifted one position to the left from the previous numbers and finally their sum forms the product. Multiplication can be performed by different methods like Array Multiplier[1], Shift and Add multiplier[2], Booth multiplier[3] and Wallace tree multiplier[4][5].

This paper discussed about architecture of Wallace tree multiplier by using diverse adder techniques like Transmission gate based full adder[7], Zhuang full adder[1] and GDI XNOR full adder[8].

A. *Wallace Tree Multiplier*

A fast process for a multiplication of two numbers was developed by Wallace. The structure of this method is looks like a tree that's why the method is known as Wallace Tree multipliers. The block diagram of Wallace tree multiplier is shown in figure 2 indicate the addition process for one column of partial products. A three-step process is used to multiply two numbers by using this method [2][3] is described as: (1)Formation of bit products (2) Reduction of the bit product matrix into a two row matrix by means of a carry save adder. (3) Summation of remaining two rows using a faster Carry Propagation Adder (CPA).

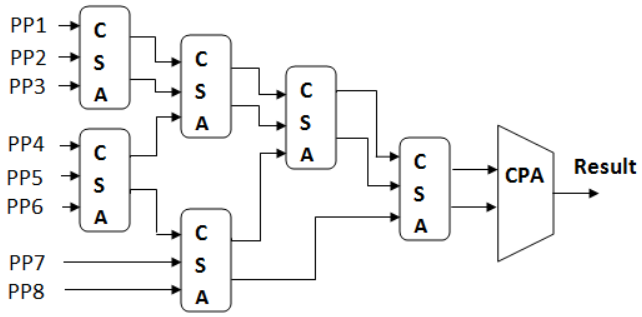

Fig. 2 Block diagram of Wallace tree multiplier

In order to design an 8-bit Wallace tree Multiplier an algorithm is derived from the figure 2. The intermediate state reductions of the multipliers are being done by Carry save adders while the final step addition is being done by a Carry Propagation Adder. In simple multiplication method as shown in figure 1, one column has maximum 8 partial product elements which are denoted as PP1 to PP8 in figure 2. For lesser number of partial products less number of CSAs are needed.

In Array multiplier the carry-out is passed to the next more significant column, while a corresponding carry-in is received from the previous column. Hence a carry is represented as being passed directly down the column.

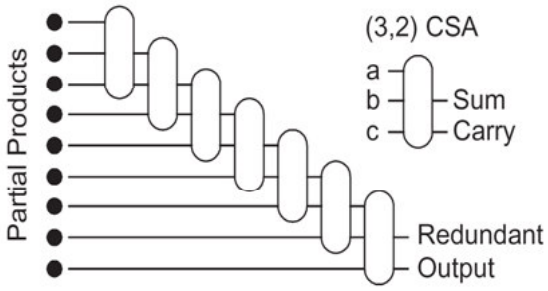

Fig. 3 Array Multiplier [1]

Figure 3 shows a dot diagram of an array multiplier column that sums 9 partial products sequentially using 7 CSAs in seven stages. The column addition is slower because only one CSA is active at a time. Another way to speed the column addition is to sum partial products in parallel rather than

sequentially. The concept is used in Wallace tree multiplier as shown in figure 4.

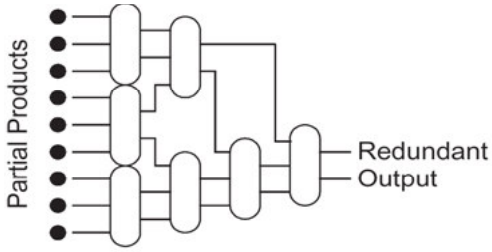

Fig. 4 Wallace tree multiplier [1]

The Wallace tree multiplier is considerably faster than a simple array multiplier due to reduction in number of stages but its wiring is much less regular and more complicated [4][5]. The Wallace tree requires N-2 CSA to reduce N inputs down to 2 carry-save redundant form outputs. Unfortunately, the routing between levels becomes much more complicated. The longer wire capacitance and the irregular tree is difficult to layout [1]. The comparison between Wallace tree multiplier and array multiplier are given below in table 1.

TABLE I COMPARISON OF ARRAY MULTIPLIER AND WALLACE TREE MULTIPLIER

Array Multiplier	Wallace Tree Multiplier
Only one CSA is active at a time	More than one CSA is active at a time
Column addition is slower	Column addition is faster
Sum of partial products is sequentially	Sum of partial products is parallel
Layout is easy	Layout is complex

As we know, in every multiplier block diagram or circuit, adder is an essential part. In multiplication partial product has been generated, and for adding purpose adder circuit is required. The different types of adder methods are discussed here.

B. Variance of Full Adder Circuits

1) Transmission Gate based Full Adder

Transmission gate based full adder (TGA) circuit is shown in figure 5 Transmission gate is made by combining NMOS and PMOS transistors in parallel. Normally conventional static CMOS full adder circuit consists of 28 numbers of transistors whereas by using transmission gate in full adder circuit, it required only 14 transistors. Transmission gate gives strong 1 and strong 0 outputs in any suitable condition.

Transmission gates are used to act as a low pass filter that suppresses glitches and reduces the capacitance and increases the speed. In this architecture of full adder A, B and Cin are the inputs and Sum and Cout are the outputs for this adder circuit. Here NAND circuit is used rather than NOR because it occupies less area and less power consumption [7].

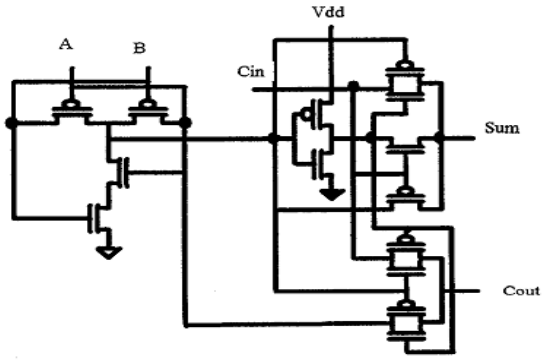

Fig. 5 Transmission gate based full adder circuit [7]

2) Zhuang full adder

Zhuang full adder (ZFA) is also made by transmission gate. In this design uses transmission gate to form a multiplexer and XORs. Figure 6 shows the transistor-level schematic using 22 transistors. The design can be understood by parsing the transmission gate structure in to multiplexer and an "invertible inverter" XOR structure as drawn in figure 7.

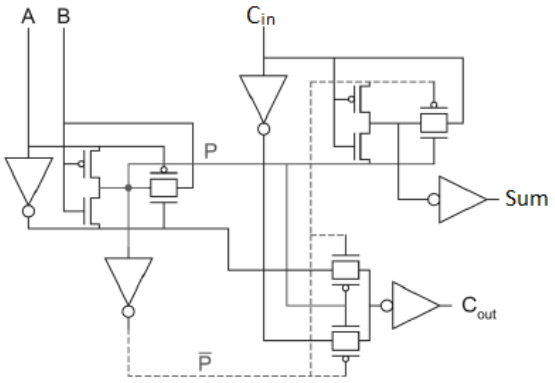

Fig. 6 Zhuang full adder [1]

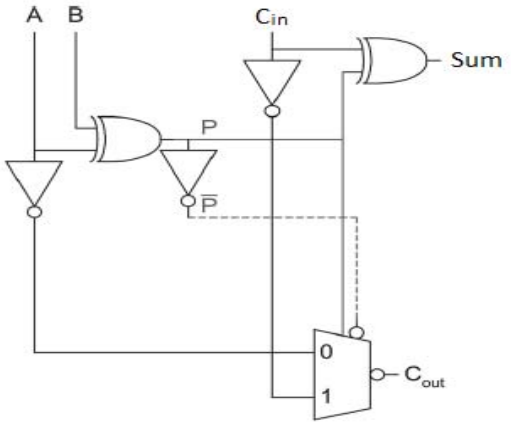

Fig. 7 Zhuang full adder (equivalent circuit) [1]

3) The Gate-Diffusion-Input (GDI) XNOR full adder

Gate Diffusion Input (GDI) method is based on the use of a simple cell. One may be reminded of the standard CMOS inverter at the first glance of this circuit, but there are some important differences: (1) The GDI cell contains three inputs—G (common gate input of NMOS and PMOS), P (input to the source/drain of PMOS), and N (input to the source/drain of NMOS). (2) Bulks of both NMOS and PMOS are connected to N or P (respectively), so it can be arbitrarily biased in contrast to CMOS inverter. The basic GDI cell is shown in Figure 8[8].

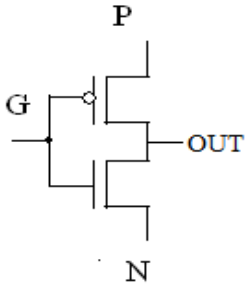

Fig. 8 Basic Gate-Diffusion-Input cell [8]

The XOR and XNOR gates based on GDI cells are applications of the GDI technique. As can be seen in Figure 9, each of them requires only four transistors. Obviously, the proposed GDI XOR and XNOR gates use less transistors compared with the conventional CMOS counterparts.

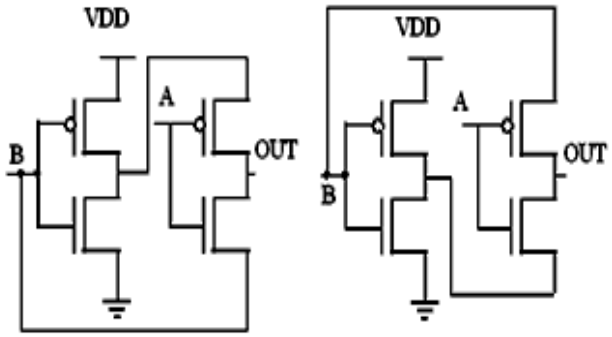

Fig. 9 (a) GDI XOR gate (b) GDI XNOR gate[8]

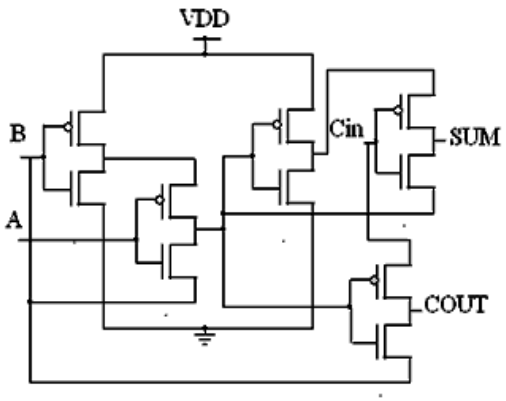

Fig. 10 GDI XNOR full adder circuit [8]

The transistor level implementation of GDI XNOR full adder (GXFA) is shown in Figure 10. This full adder consists of three modules, two GDI XNOR gates and a multiplexer. The Sum and Cout can be calculated using SUM and COUT equation. In the worst case, Sum has 4-T delay while Cout has 3-T delay. However, due to the advantages of GDI cell, this circuit still can achieve its benefit of low power consumption.

IV. SIMULATION AND RESULT ANALYSIS

A. Result analysis for adder circuits

In this section simulation result of three different types of adders are discussed. Based on these three adders, Wallace tree multiplier is implemented. Schematic and layout are created using 180nm technology in electricbinary 9.00v and spice simulation is done in LT-Spice. TGA, ZFA and GXFA requires 14, 22, 10 transistors respectively for 1 bit adder circuit. Delay analysis for all this adder circuits are shown in table 2 in comparative manner.

From the table 2, GXFA circuit gives the maximum worst case as well as average delay. ZFA gives minimum average delay and for worst case in comparison to other two adder circuits.

TABLE II COMPARISON IN TERMS OF DELAY ANALYSIS BETWEEN DIFFERENT ADDER CIRCUITS

Input			Transmission gate full adder		Zhuang full adder		GDI XNOR full adder	
A	**B**	**Cin**	**Sum**	**Cout**	**Sum**	**Cout**	**Sum**	**Cout**
0	0	0	0.4	0	0	0.26	0.61	0.38
0	0	1	0.87	0.95	0.42	0.42	0.43	0.44
0	1	0	0.15	0.02	0.31	0	0.6	0.13
0	1	1	0.2	0.16	0.4	0	1.25	0.41
1	0	0	0.1	0	0	0.45	0.06	0.1
1	0	1	0.15	0.02	0	0.2	0	0.32
1	1	0	0.01	0.18	0.71	0.63	0.43	0
1	1	1	0.5	0.1	0.31	0.23	0.26	0
Average Delay (in ns)			0.30	0.18	0.27	0.27	0.45	0.22
Worst case delay (in ns)			0.87	095	0.71	0.63	1.25	0.44
Area (in μm^2)			126.26		249.6		85.9	

Figure 11 and 12 shows layout of TGA and GXFA and corresponding layout area is indicated in table 2. Less number of transistors results in less area. Figure 13, 14 and 15 indicates waveform for 1 bit addition using TGA, GXFA and ZFA respectively.

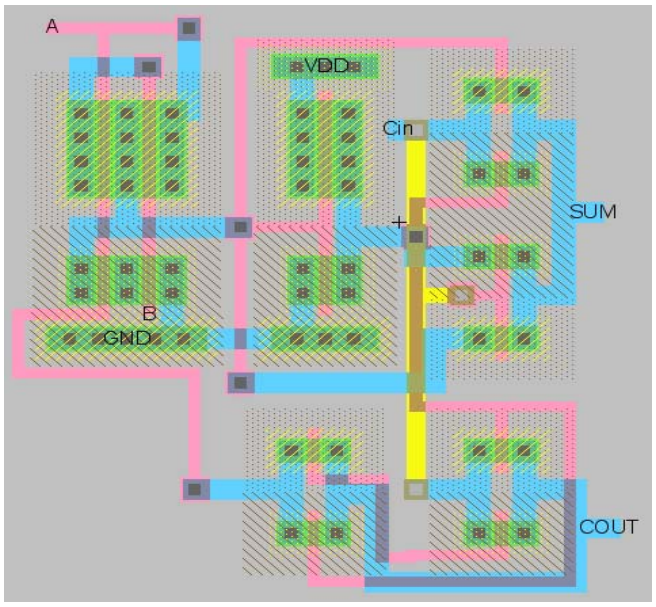

Fig. 11 Layout of 1 bit TGA

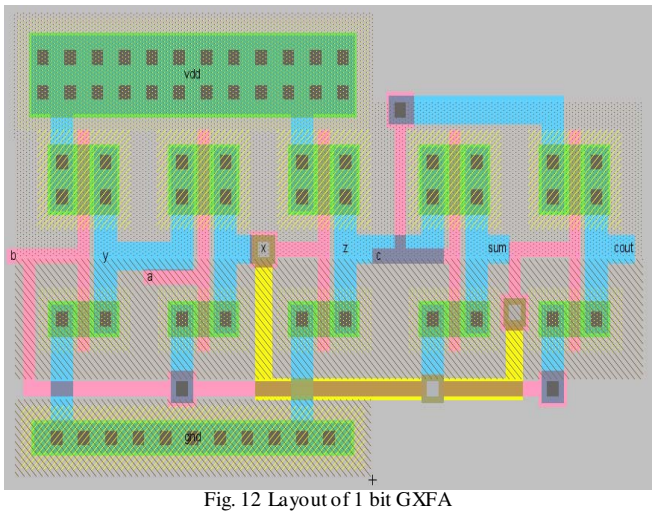

Fig. 12 Layout of 1 bit GXFA

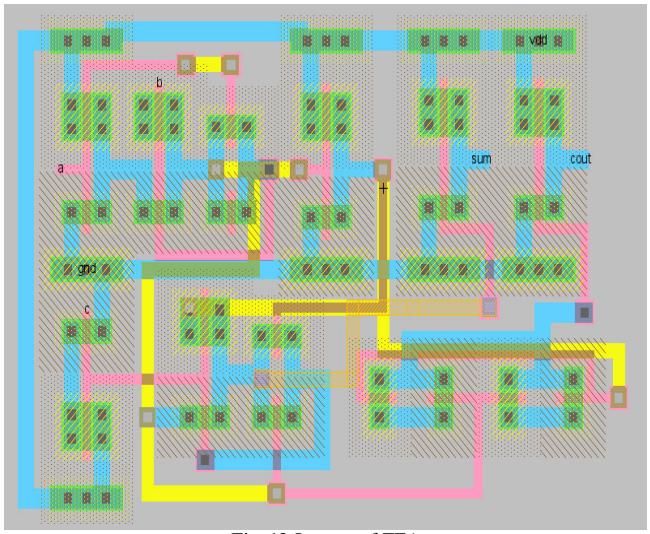

Fig. 13 Layout of ZFA

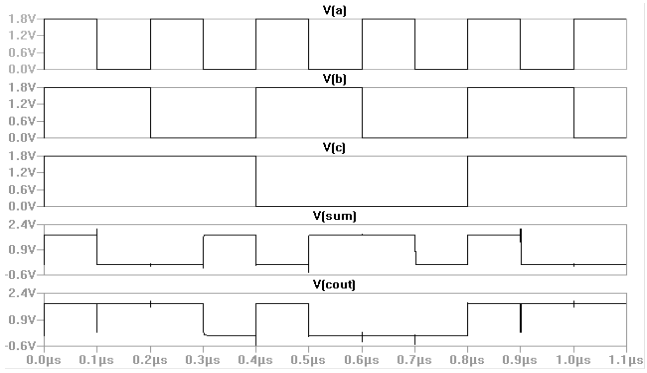

Fig. 14 waveform of TGA

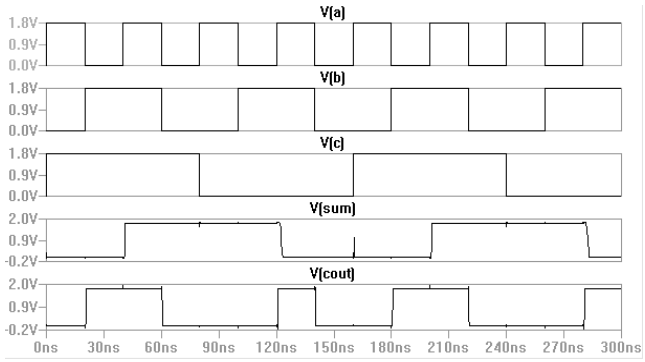

Fig. 15 waveform of GXFA

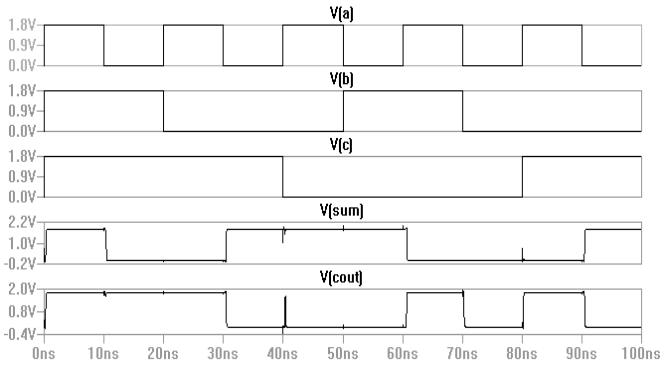

Fig. 16 waveform of ZFA

B. Result analysis for Multiplier circuits

Three different types of architecture for Wallace tree multiplier has been simulated by using three different adder circuits. Comparison in terms of delay, area, power and number of transistors used are done between these three types of architectures which are shown in table 3. From area point of view multiplier based on GXFA is small in size due to less number of transistors but offers maximum delay. ZFA based multiplier is fast at the cost of excess numbers of transistors and power. Overall multiplier based on TGA gives bit more delay then ZFA and number of transistors bit more then GXFA but consumes very less power.

TABLE III COMPARISON BETWEEN THREE TYPES OF ARCHITECTURE OF WALLACE TREE MULTIPLIER

Wallace Tree Multiplier	Average delay (in ns)	No. of Transistors used	Power (in mW)
Using TGA	2.82	1280	0.14
Using ZFA	2.62	1792	1.69
Using GXFA	431.44	1024	0.408

V. CONCLUSION

After comparative analysis of various full adders, GDI XNOR full adder, transmission gate full adder and Zhuang full adders are selected for Wallace tree multiplier. In this paper implementation of Wallace tree multiplier is made using TGA, GXFA and ZFA. After designing Wallace tree multiplier by using three different adder circuits, now it is a tread of situation. Where area and power are the main criteria, GXFA is a best suited and where speed is the only criteria, ZFA is best suited as an adder circuit. Whereas Wallace tree multiplier using TGA gives good result in all the criteria. It consumes little more area than GXFA but very less than ZFA. Results can be compared with [9] and we can observe, 42% transistors are saved, results in reduction of area, and reduction in power at improved speed of operation.

REFERENCES

[1] Neil H. E. Weste & David Harris, "CMOS VLSI Design- A circuit and Systems Perspective", 4th edition, Addison Wesley, 2010

[2] C. N.Marimuthu, Dr. P. Thangaraj, Aswathy Ramesan, " Low power shift and add multiplier design", International Journal of Computer Science and Information Technology, June 2010, Vol. 2, Number 3.

[3] Marc Hunger, Daniel Marienfeld, "New Self-Checking Booth Multipliers", International Journal of Applied Mathematics Computer Sci., 2008, Vol. 18, No. 3, 319–328

[4] C. Jaya Kumar, R. Saravanan, "VLSI Design for Low Power Multiplier using Full Adder", European Journal of Scientific Research, ISSN 1450-216X Vol.72 No.1 (2012), pp. 5-16

[5] Ravi Nirlakalla, Thota Subba Rao, Talari Jayachandra Prasad, "Performance Evaluation of High Speed Compressors for High Speed Multipliers", Serbian Journal of Electrical Engineering, Vol. 8, No. 3, November 2011, 293-306

[6] C. H. Chang, J. Gu, M. Zhang, "A review of 0.18μm full adder performances for tree structured arithmetic circuits", IEEE Transactions Very Large Scale Integration System., Vol. 13, No. 6, June 2005, pp. 686 - 695.

[7] E. Abu-Sharma, M.B. Mazz, M.A Bayoumi, "A Fast and Low Power Multiplier Architecture", Circuits and Systems, IEEE 39th Midwest symposium, 18-21 August, 1996, Vol. 1, pp. 53-56.

[8] Dan Wang, Maofeng Yang, Wu Cheng, Xuguang Guan, Zhangming Zhu, Yintang Yang," Novel Low Power Full Adder Cells in 180nm CMOS Technology", National Natural Science Foundation of China, 2009 IEEE, pp. 430 – 433.

[9] Dr. P.T.Vanathi, P. Ramanathan, Amresh Chaubey and N. Senthil Raja, "Decomposition Algorithm for Power delay product optimization in Wallace multiplier", Control, Automation, Communication and Energy Conservation, 2009. INCACEC 2009. 2009 International Conference, 2009, pp. 1-6, 4-6 June.