International Journal of Electrical and Electronic Engineering & Telecommunications

ISSN 2319 – 2518 www.ijeetc.com
Vol. 3, No. 2, April 2014

Review Article

REVIEW ON OPTIMIZATION TECHNIQUES OF WALLACE TREE MULTIPLIER

K Padma Priya[1]*, P Sai Arun Kumar[1], B Mounika[1], P Lavanya[1],
D Sivananda Das[1] and B Buchi Babu[1]

*Corresponding Author: K Padma Priya, ✉kesaripadmapriya@yahoo.com

In present generation, VLSI systems and their design became so much important in the Electronic Engineering. As the VLSI design is the basis for electronic components, one should optimise the constraints. For high performance systems such as digital signal processors and microprocessors, Multiplier is the key hardware block. In designing VLSI systems, the main criteria of interest are high speed, low power, less area, low cost. 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 compact VLSI implementations. Improving speed results always in larger areas. So, this paper provides the techniques that optimise the area and speed up the multiplication process.

Keywords: Multiplier, Wallace tree structure, Compressors, Sklansky adder

INTRODUCTION

Most digital signal processing algorithms involves arithmetic operations which dominates the execution time. The dominating factor in determining instruction cycle time of Reduced Instruction Set Computers (RISC) and DSP chip is the time to execute multiplication operation (Lakshmanan *et al.*, 2002). Among the many methods of implementing parallel multipliers, there are two basic approaches namely Booth algorithm and Wallace tree compressors. In Wallace tree structure multipliers, the multiplier array becomes large and requires more number of logic gates and interconnecting wires. This makes the chip design complex and slow down the operating speed (Lakshmanan *et al.*, 2002). This paper describes about the pitfalls of conventional Wallace tree structure and designing techniques of high speed Wallace tree multiplier. It will describe the speeding techniques of conventional Wallace Tree multiplication structure which focus mainly on the acceleration of the process (Wallace,

[1] ECE Department, JNTUKUCEV, Vizianagaram, Andhra Pradesh, India.

1964). Acceleration of the process based purely on the following aspects of normal multiplication:

- Reduction in the number of summands;
- Acceleration of formation of summands;
- Acceleration of the addition of summands (Wallace, 1964).

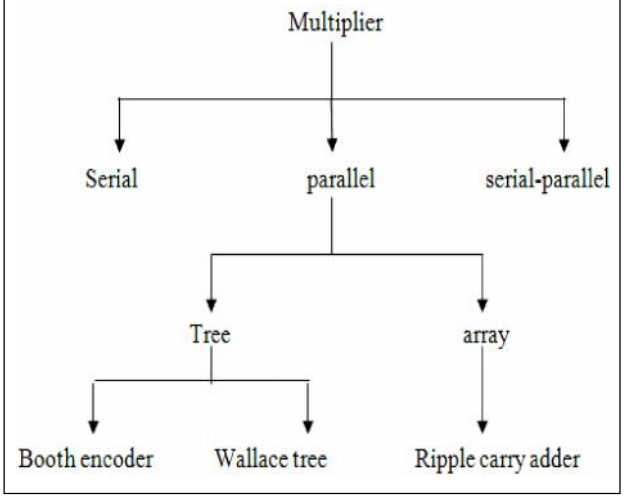
It consists of sections which describe about multiplier, Wallace Tree multiplier, pitfalls of conventional Wallace Tree multiplier and designing techniques of high speed Wallace Tree multiplier.

MULTIPLIER

Multiplication is hardware intensive and fast multipliers are essential for digital signal processing systems. General purpose DSPs can afford to have customized high performance multipliers. Application specific DSPs implement algorithms as direct realization of their floe graphs, thus having several multipliers, each one with different specifications (Jalil Fadavi-Ardekani, 1993). Multiplication is essential for computers, process controllers, microprocessors, digital signal processing and graphic engines. System performance is determined by performance of multiplier because it is slowest element and area consuming. Multiplication operation is basically a shift and add operation. Single bit multiplication is equivalent to logical AND operation. Multi bit multiplication is of AND, ADD, SHIFT operations. There are number of techniques that can be used to perform multiplication. In general, the factors such as latency, throughput, area and design complexity are the basis of the choice (Neil Weste *et al.*, 2005).

Digital multipliers can be classified into serial, parallel and serial-parallel multipliers. Parallel multipliers are again of two types, Array multipliers and tree multipliers. The selection of multiplier depends upon the application. The below Figure 1 describes the classification of multipliers.

Figure 1: Classification of Multipliers



WALLACE TREE MULTIPLIER

Professor Christopher Stewart Wallace devised the Wallace Tree Multiplier algorithm in 1964. The Wallace tree multiplier is considerably faster than a simple array multiplier and is an efficient implementation of a digital circuit that multiplies two integers. The Figure 2 shows the Wallace proposed tree architecture (Wallace, 1964) and Figure 3 shows structural representation of 4 x 4 multiplier.

Features of Wallace Tree Multiplier

- Optimized column adder tree.
- Combines all partial products into two vectors (carry and sum).
- Carry and sum outputs combined using a conventional adder.

Figure 2: Wallace Tree Structure

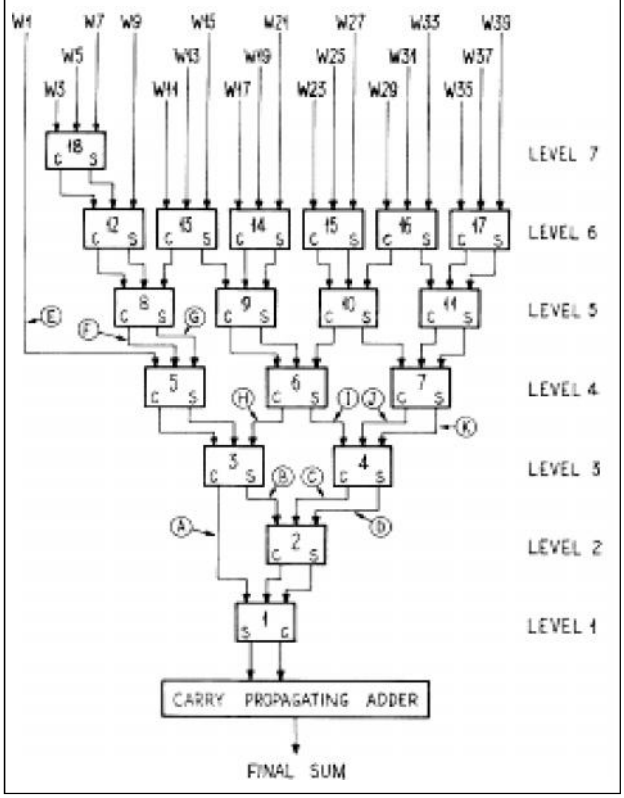
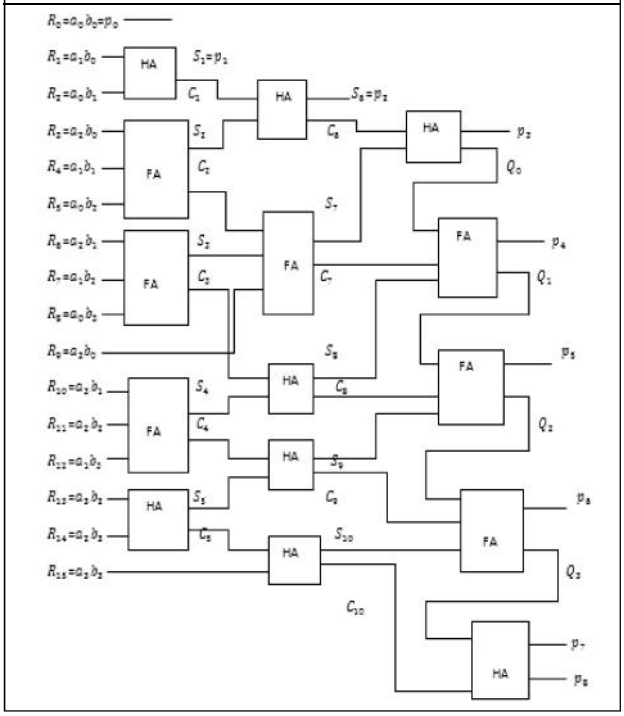


Figure 3: 4 x 4 Multiplier-Structural Representation



- Wallace tree multiplier uses Wallace tree to combine 1 X M partial products.
- Irregular routing.

Components

- Half adder
- Full adder
- Ripple carry adder
- Carry propagating adder

Operation Used in Wallace Tree Multiplier

As an example for the description of operation and logic used in the Wallace Tree manipulation, the below Figure 4 shows 8-bit multiplier constructed using Wallace Tree architecture. Partial product are added in 6 steps. In the Wallace Tree Method shown below, the stages are reduced by using components like half adders and full adders (Naveen Gahlan *et al.*, 2012).

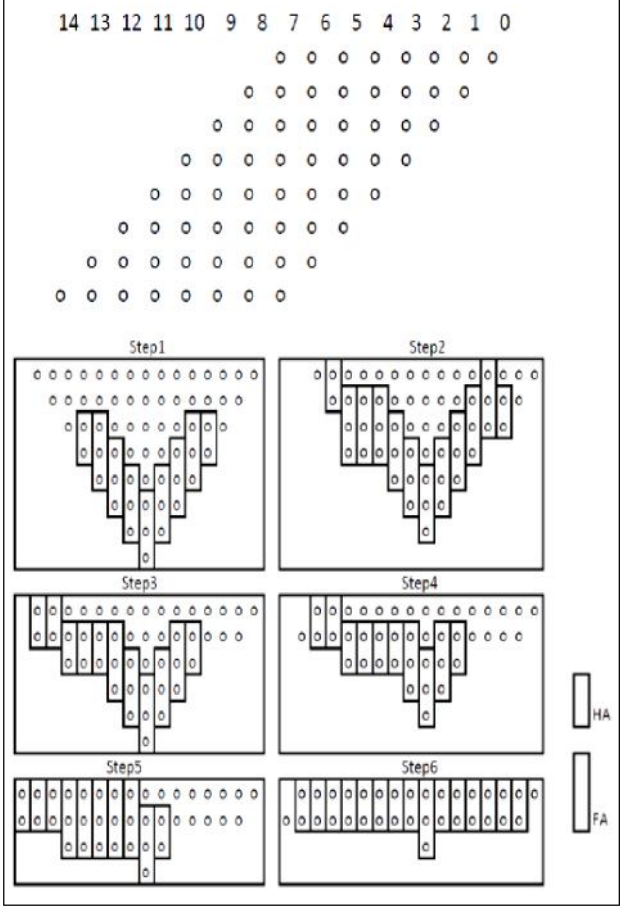
PITFALLS IN CONVENTIONAL WALLACE TREE MULTIPLIER

- Wallace trees do not provide any advantage over ripple adder trees in many FPGAs.
- Due to the irregular routing, they may actually be slower and are certainly more difficult to route.
- Adder structure increases for increased bit multiplication.

DESIGNING TECHNIQUES OF HIGH SPEED WALLACE TREE MULTIPLIER

Realizing high speed multiplier can be done by enhancing the parallelism so that the

Figure 4: Wallace Tree Multiplier Reduction Stages for 8 x 8 Multiplication



subsequent calculation stages will decrease. To improve the functionality of the conventional Wallace tree multiplier, we here studied and described some methods to reduce power and increase the speed to the maximum extent and one can implement the below methods and can experience the improvement in the constraints related to the conventional Wallace tree multiplier. They can be done in two ways mainly. They are:

1. Design by using compressors and sklansky adder (Novel Architecture) (Vinoth *et al.*, 2011).
2. Design using pipelined architecture (Tajul Hamimi Harun, 2007).

Design of High Speed and Low Power Wallace Tree Structure Using Compressors and Sklansky Adder (Novel Structure)

Conventional Wallace Tree Structure vs. Novel Structure

In conventional 8 bit Wallace tree multiplier design, more number of addition operations is required. Three partial product terms can be added at a time to form the carry and sum using the carry save adder. At the next level, the sum signal is used by the full adder. The carry signal is used by the adder involved in the next output bit generation with a resulting overall delay which is proportional to log 3/2 n, for n number of rows. In the first and second stages of the Wallace structure, the partial products do not depend upon any values other than the inputs obtained from the AND array. However, the final value depends on the carry out value of previous stage for the intermediate higher stages. This operation is repeated sequentially for the consecutive stages. Finally, it is observed that major cause of delay is propagation of the carry out from the previous stage to the next stage. In conventional Wallace Tree multiplier structure, the total number of stages in the critical path sums up to 13. Each full adder used accounts for a latency of 2. Total latency of the whole structure gives a total of 26. The latency count gets increased by 1, when considered the and array, which results the total latency.

Where the novel structure aims to reduce the overall latency. This increases the speed and reduces power consumption. This design makes use of compressors instead of the full adders and the final stage is replaced by a sklansky tree adder. In first stage, consists of

full adder. In the second stage, two full adders are grouped using 4:2 compressors and implemented. Then, in third stage, using a 5:2 compressor, which is a combination of 3 full adders and so on. In this way, the individual blocks of full adders in the original structure are grouped and implemented with the help of compressors. The use of sklansky adder in the structure further results in a reduced latency of 6 with a latency of 1 for the AND array. This novel structure, at a whole bring down the latency count to 15. Thus, a significant latency reduction of 44.4% is realized than conventional structure. The figures (Figure 5 shows critical path of conventional tree structure (Vinoth *et al.*, 2011) and Figure 6 shows novel architecture (Vinoth *et al.*, 2011)).

Figure 5: Critical Path in Conventional Wallace Tree

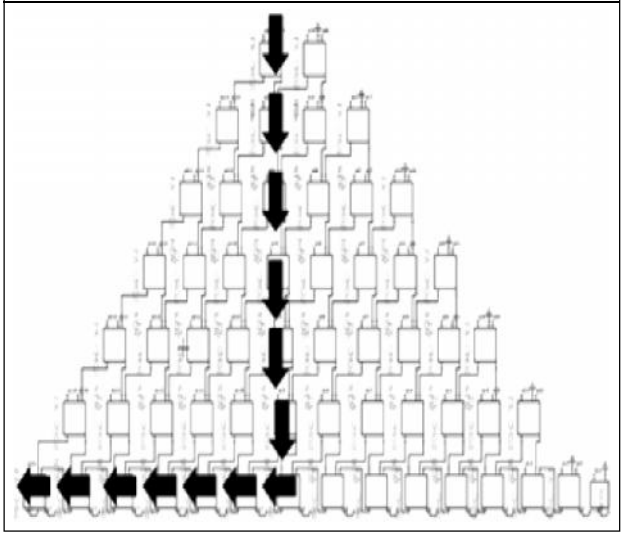
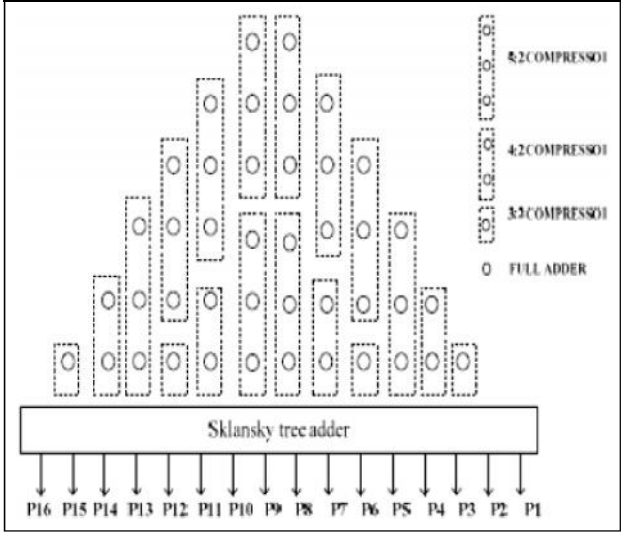


Figure 6: Proposed Wallace Tree Under Novel Architecture



Compressors

The half adder and full adder contributes more delay in circuit and this total delay is linear to the number of addition stages. Now a days, for high speed multipliers, compressors are widely used which lower the latency of partial product accumulation stage. The proposed architecture of Wallace tree multiplier is focused on minimization of delay in partial product reduction stage using compressors.

A compressor is single bit adder circuit. A high speed compressor can perform more than three bit addition in parallel. Realization and reduction in number of partial product addition stages can be done using multi bit compressors. 3:2 is combinational circuit which sum up three binary inputs of one bit and gives sum and carry output of one bit. It is equivalent to full adder.

$$Sum = X1 \oplus X2 \oplus X3$$

$$Carry = X1X2 + X2X3 + X3X1$$

4:2 compressor has 4 inputs (*X*1, *X*2, *X*3 and *X*4) and 2 outputs (*sum* and *carry*) along with carry-in and carry-out. It consists of two 3:2 compressors cascaded in series. This involves a critical path delay of three X-OR gates and hence reducing the critical path by one X-OR delay. 5:2 compressor is widely used building block in high speed and high precision multipliers. For an example of the

compressors, the figures Figure 7 shows the block diagram (Karthick *et al.*, 2012) and Figure 8 shows architecture of 5:2 compressor (Karthick *et al.*, 2012).

Figure 7: Block Diagram of 5:2 Compressor

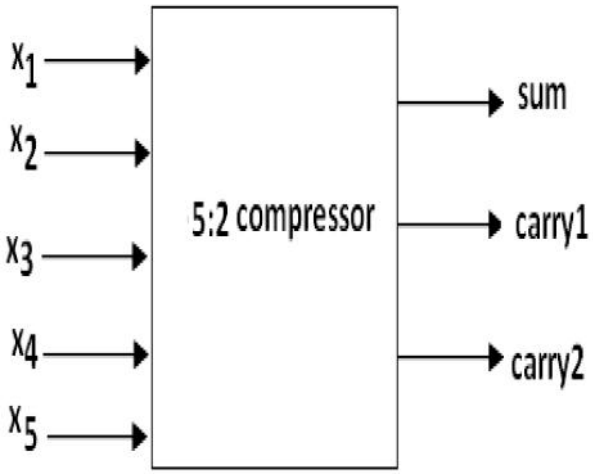
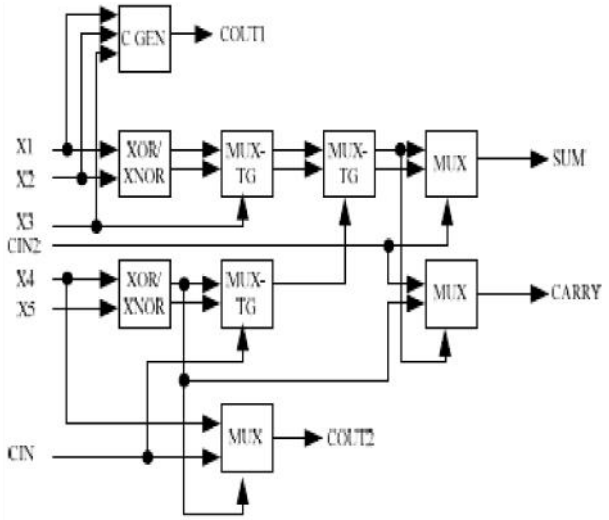


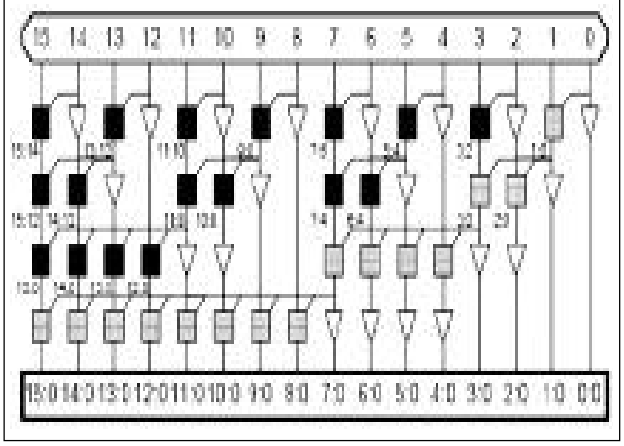
Figure 8: High Speed 5:2 Compressor



Sklansky Adder

Binary addition is one of the most primitive and most commonly used applications in computer arithmetic. A large variety of algorithms and implementations have been proposed for binary addition. When high speed operation is required, tree structures like parallel prefix adders such as Kogge-stone, Sklansky, Brent-kung, Han-carlson and Kogge-stone with ling damadders (Dakupati Ravi Sankar and Shaik Ashraf Ali, 2013). The final stage for the summation of the partial products can be further minimised by the usage of tree adders in the Wallace tree multiplier, which reduces the power consumption preliminarily and also increases the speed of the operation. The fundamental concept of tree adders extend from the idea of carry look-ahead computation and the class of parallel carry look-ahead schemes. These architectures give so many high-performance applications. Because of the lower power consumption than incurred by the other tree structures, here the sklansky adder tree is preferred (Vinoth *et al.*, 2011). The Figure 9 shows sklansky tree adder structure (Vinoth *et al.*, 2011).

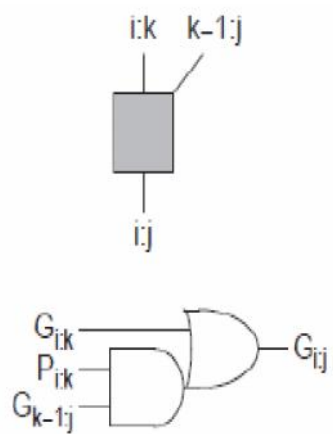
Figure 9: Sklansky Tree Adder Structure



In the above figure, the binary trees of cells first generate all the carry input bits simultaneously. This architecture follows a regular pattern. The delay is ultimately reduced to log2N by computing intermediate prefixes along with the large group prefixes. Here, N represents the number of bits. This also results

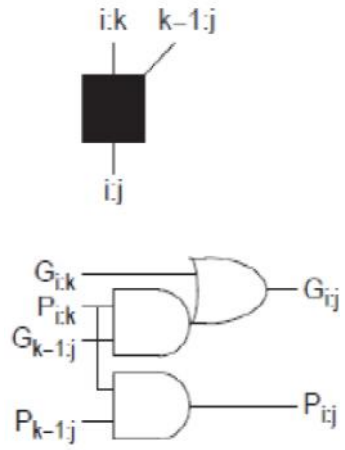
in an increased number of fan-outs at each level. In Figure 10 the black squares correspond to AND-OR-AND logic and in the Figure 11 grey squares correspond to AND-OR logic. The triangular box represents buffers (Vinoth *et al.*, 2011).

Figure 10: Gray Cell



Source: Dakupati Ravi Sankar and Shaik Ashraf Ali (2013)

Figure 11: Blackcell



Source: Dakupati Ravi Sankar and Shaik Ashraf Ali (2013)

Design Using Pipelined Structure

The Wallace tree architecture under pipelined structure is studied and compared with the conventional. As results are observed, after complete analysis of the Wallace Tree multiplier, it has been proven that the pipelining method could increased the speed of the multiplier. However, the area of the multiplier designed is also increased because of the 4-stages D flip-flop that has been implemented in the design. Basically, the objective of this project is mainly focus on the speed of the Wallace Tree multiplier, so, the increased area of the multiplier does not take into count. Otherwise, the high speed 8-bits x 8-bits (Tajul Hamimi Harun, 2007).

The conventional circuit produces the maximum speed of 14.99 MHz or maximum delay of 66.7 nanoseconds to complete one process of 8-bits x 8-bits multiplication. However, after upgrading the conventional Wallace Tree into pipeline Wallace Tree, by adding D flip-flop stages at every input of the

Figure 12: Conventional 8 x 8 WT Architecture

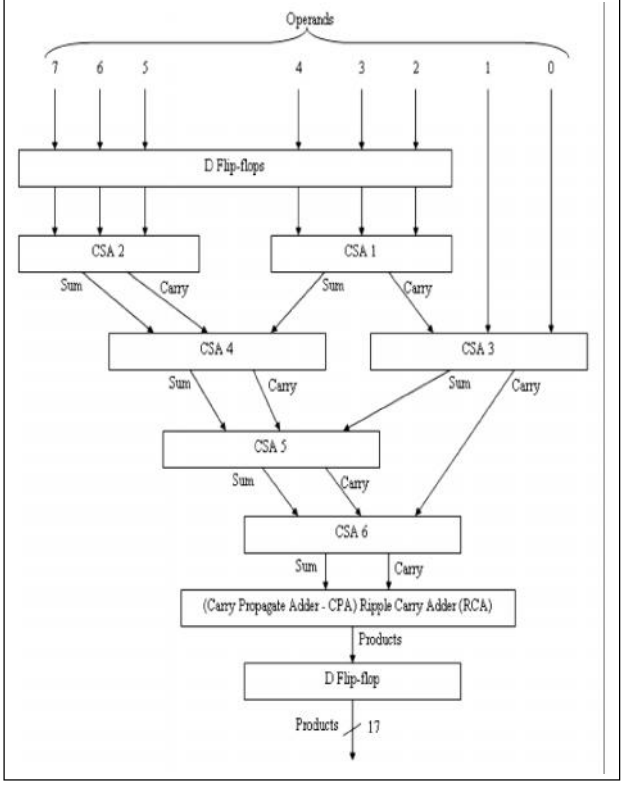
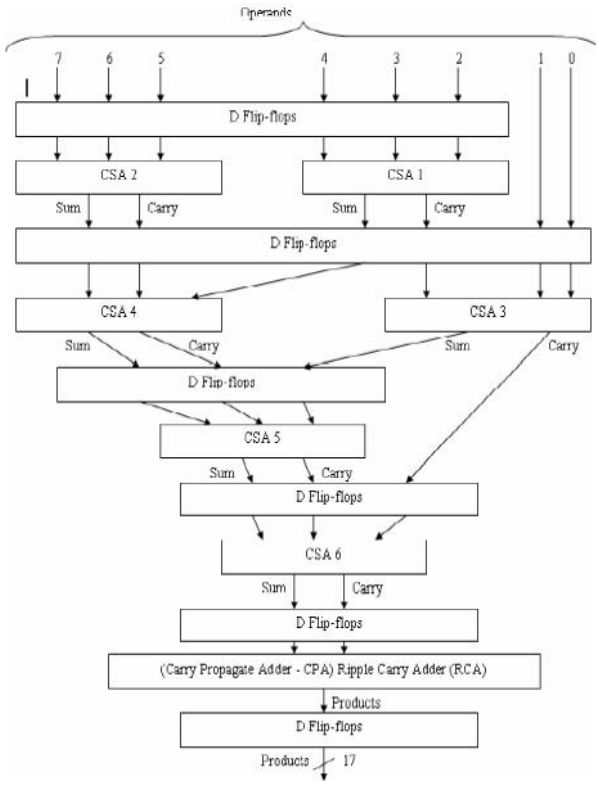


Figure 13: 8 x 8 High Speed WT Under Pipelined Structure



half adder and the full adder, the speed has improve much, increased to 54.05 MHz and the maximum delay has decreased to 18.3 nanoseconds. It was three times faster than the conventional design. The Figures 12 and 13 shows conventional and high speed architectures.

REFERENCES

1. Chip-Hong Chang, Jiangmin Gu and Mingyan Zhang (2004), "Ultra Low Voltage Low Power CMOS 4-2 and 5-2 Compressors for Fast Arithmetic Circuits", *IEEE Transactions on Circuits and Systems-I: Regular Papers*, Vol. 51, No. 10.
2. Dakupati Ravi Sankar and Shaik Ashraf Ali (2013), "Design of Wallace Tree Multiplier by Sklansky Adder", *International Journal of Engineering Research and Applications (IJERA)*, Vol. 3, No. 1, pp. 1036-1040.
3. "Implementation of 8 x 8 Wallace Tree Multiplier Using VHDL", A Project Report of Students of Bapatla Engineering College (available on internet).
4. Jalil Fadavi-Ardekani (1993), "Mxn Booth Encode Multiplier Generator Using Optimised Wallace Trees", *IEEE Transactions on Very Large Scale Integrated Systems*, Vol. 1, No. 2.
5. Jasbir Kaur and Kavita (2013), "Structural VHDL Implementation of Wallace Tree Multiplier", Vol. 4, No. 4, ISSN: 2229-5518.
6. Karthick S, Karthika S and Valarmathy S (2012), "Design and Analysis of Low Power Compressors", Vol. 1, No. 6, ISSN: 2278-8875.
7. Lakshmanan, Masuri Othman and Mohamad Alauddin Mohd. Ali (2002), "High Performance Parallel Multiplier Using Wallace-Booth Algorithm".
8. Naveen Kr. Gahlan *et al.* (2012), "Implementation of Wallace Tree Multiplier Using Compressor", *Int. J. Computer Technology & Applications*, Vol. 3, No. 3, pp. 1194-1199.
9. Neil H E Weste, David Harris and Ayan Banerjee (2005), "CMOS VLSI Design", *A Circuits and Systems Perspective*, 3rd Edition.
10. Paul F Stelling, Vojin G Oklobdzija and Ravi R (1996), "Optimal Circuits for Parallel Multiplier", *IEEE Transactions on Computers*, Vol. 47, No. 3.

11. Tajul Hamimi Harun (2007), “High Speed 8 x 8 Wallace Tree Multiplier”, A Report Submitted in 2007 (from internet).

12. Vinoth C, Kanchana Bhaaskaran V S, Brindha B, Sakthikumaran S, Kavinilavu V, Bhaskar B, Kanagasabapathy M and Sharath B (2011), “A Novel Low Power and High Speed Wallace Tree Multiplier for RISC Processor”, 978-1-4244-8679-3/11.

13. Wallace C S (1964), “A Suggestion for Fast Multipliers”, *IEEE Transactions on Electronic Computers*, Vol. EC-13, February, pp. 14-17.