ISSN: 2277-9655
Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Design of Wallace Tree Multiplier using Compressors

K.Gopi Krishna[*1], B.Santhosh[2], V.Sridhar[3]
gopikoleti@gmail.com

Abstract

A multiplier is one of the key hardware blocks in most digital and high performance systems such as FIR filters, digital signal processors and microprocessors etc. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following- high speed, low power consumption, regularity of layout and hence less area or even combination of them in multiplier. Thus making them suitable for various high speed, low power, and compact VLSI implementations. However area and speed are two conflicting constraints. So improving speed results always in larger areas. So here we try to find out the best trade off solution among the both of them.

Generally as we know multiplication goes in three basic steps. Partial product generation, reduction and final stage is addition. Hence in this paper we have first tried to design different adders and compare their speed and complexity of circuit i.e. the area occupied. And then we have designed Wallace tree multiplier then followed by Conventional, proposed Wallace multipliers and have compared the speed and Power consumption in both of them. While comparing the adders we found out that Ripple Carry Adder had a smaller area while having lesser speed, in contrast to which sklansky Adders are high speed but posses a larger area. After designing and comparing the adders we turned to multipliers. Initially we went for Parallel Multiplier and then Wallace Tree Multiplier. In the mean time we learned that delay amount was considerably reduced when sklansky adder were used in Wallace Tree applications.

Keywords:

Introduction

Risc Processors

The trend in the past shows the RISC processors clearly outsmarting the earlier CISC processor architectures. The reasons have been the advantages, such as simplicity, flexibility. paves for higher clock speed, by eliminating the need for microprogramming through fixed instruction format and hardwired control logic. The combined advantages of high speed, low power, area efficient and operation-specific design possibilities have made the RISC processor universal. The main feature of the RISC processor is its ability to support single cycle operation, meaning that the instruction is fetched from the instruction memory at the maximum speed from the memory.

RISC processors are designed to achieve this by pipelining, where there is a possibility of stalling of clock cycles due to wrong instruction fetch when jump type instructions are encountered. This reduces the efficiency of the processors. This paper describes a RISC architecture in which, single cycle operation is obtained without using a pipelined design. It averts possible stalling of clock cycles in effect. The development of CMOS technology provides very high density and high performance integrated circuits. The performance provided by the existing devices has created a never-ending greed for increasingly better performing devices. This predicts the use of a whole RISC processor as a basic device by the year 2020. However, as the density of IC increases, the power consumption becomes a major threatening issue along with the complexity of the circuits.

Basic Multipliers

The growing market for fast floating-point co-processors, digital signal processing chips, and graphics processors has created a demand for highspeed, area-efficient multipliers. Current architectures range from small, low-performance shift and add multipliers, to large, high-performance array and tree multipliers. Conventional linear array multipliers achieve high performance in a regular structure, but require large amounts of silicon. Tree structures achieve even higher performance than linear arrays but the tree interconnection is more complex and less regular, making them even larger than linear arrays. Ideally, one would want the speed benefits of a tree structure, the regularity of an array multiplier, and the small size of a shift and add multiplier.

This thesis presents a new tree multiplier architecture which is smaller and faster than linear array multipliers, and more regular than traditional multiplier trees. At the heart of the architecture is a new tree structure, the 4-2 tree. The regular structure of the 4-2 tree is the result of using a 4-2 adder as the basic building block. A row of 4-2 adders can be used to reduce four inputs to two outputs. In contrast, the carry-save adders used in Wallace trees reduce three inputs to two outputs. The 240-1 reduction of the 4-2 adders produces a binary tree structure which is much more regular than the, 3-to-2 structure found in Wallace trees. As such, 4-2 trees are better suited for VLSI implementations than traditional multiplier trees.

Wallace Tree Multipler

Wallace tree reduces the number of partial products to be added into 2 final intermediate results. The Wallace tree basically multiplies two unsigned integers,A Wallace tree is an efficient hardware implementation of a digital circuit that multiplies two integers, devised by an Australian Computer Scientist Chris in 1964.

The Wallace tree has three steps:

1. Partial Product Generation Stage
2. Partial Product Reduction Stage
3. Partial Product Addition Stage

Partial Product Generation Stage :

Partial product generation is the very first step in binary multiplier. These are the intermediate terms which are generated based on the value of multiplier. If the multiplier bit is ‘0’, then partial product row is also zero, and if it is ‘1’, then the multiplicand is copied as it is. From the 2nd bit multiplication onwards, each partial product row is shifted one unit to the left as shown in the above mentioned example. In signed multiplication, the sign bit is also extended to the left. Partial product generators for a conventional multiplier consist of a series of logic AND gates as shown in Figure.

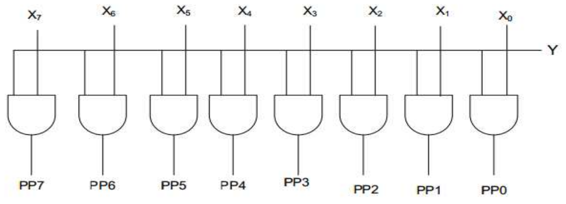


Figure3.1 : Partial product selection logic for simple multiplication.

The main operation in the process of multiplication of two numbers is addition of the partial products. Therefore, the performance and speed of the multiplier depends on the performance of the adder that forms the core of the multiplier. To achieve higher performance, the multiplier must be pipelined..

Partial Product Reduction Stage:

The design analysis starts with the analysis of the elementary algorithm for multiplication by Wallace Tree multiplier. Figure 3.1 shows the algorithm for 8-bits x 8-bits multiplication performs by Wallace Tree multiplier. There are five stages to go through, to complete the multiplication process. Each stage used half adders and full adders that are denoted by the red circle for the 1 bit half adder and the blue circle for the 1-bit full adder. Firstly, we have to reduce the partial products using half adders and full adders that are combined to build a carry-save adder (CSA) until there were just two rows of partial products left. Next, we add the remaining two rows by using a fast carry-propagate adder. For this project, ripple-carry adder (RCA) is used, to get the final product of the two operands multiplication. Secondly, the schematic of the conventional 8-bits x 8-bits high speed Wallace Tree multiplier is design by referring to the algorithm. Figure 3.2 shows the block diagram for the conventional high speed 8-bits x 8-bits Wallace Tree multiplier. Reduce the number of partial products to two by layers of full and half adders.

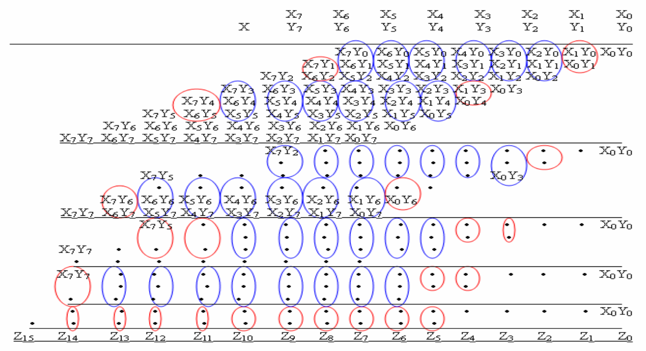

X
X7 Y7
X6 Y6
X5 Y5
X4 Y4
X3 Y3
X2 Y2
X1 Y1
X0 Y0
X7Y0 X6Y0 X5Y0 X4Y0 X3Y0 X2Y0 X1Y0 X0Y0
X7Y1 X6Y1 X5Y1 X4Y1 X3Y1 X2Y1 X1Y1 X0Y1
X7Y7
X0Y0
Z15 Z14 Z13 Z12 Z11 Z10 Z9 Z8 Z7 Z6 Z5 Z4 Z3 Z2 Z1 Z0

Fig3.2 8*8 Multipication

when the Verilog source code of the multiplier has been design, we must simulate and check its functionality. If it is functioning correctly, we could proceed to the next step, which is to determine the maximum speed and time that the multiplier takes to complete a single multiplication process.

Proposed Wallace Tree Multipler

The proposed architecture aims to reduce the overall latency. This leads to increased speed and reduced power consumption. The design makes use of compressors in place of full adders, and the final carry propagate stage is replaced by a Sklansky tree adder.

Figure depicts the first stage consisting of a full adder. In the second stage, two full adders have been grouped and implemented using a 4:2 compressor. Similarly, the third stage consists of a 5:2 compressor, which is a combination of 3 full adders and so on. In this manner, the individual full adder blocks in the original structure are grouped and implemented using compressors. The number of interconnections is taken care of, since they play a vital role in the flow of carry from one stage to the next in the tree.

we can see that the longest delay path of our design is the one consisting of two 5:2 compressors, which produces a reduced latency of 8 (four per compressor) only. The use of the Sklansky adder in the structure further results in a reduced latency of 6 with a latency of 1 for the AND array. Hence, this novel structure brings down the overall latency count to 15. Thus, a significant latency reduction of 44.4% than the conventional counterpart is realized. The symbolic arrangement of the proposed structure is depicted in Fig. 13 for elaboration.

Partial Product Generation Stage:

The Wallace tree basically multiplies two unsigned integers. The Proposed Wallace tree multiplier architecture comprises of an AND array for computing the partial products, an adder for adding the partial products so obtained and a sklansky adder in the final stage of addition.

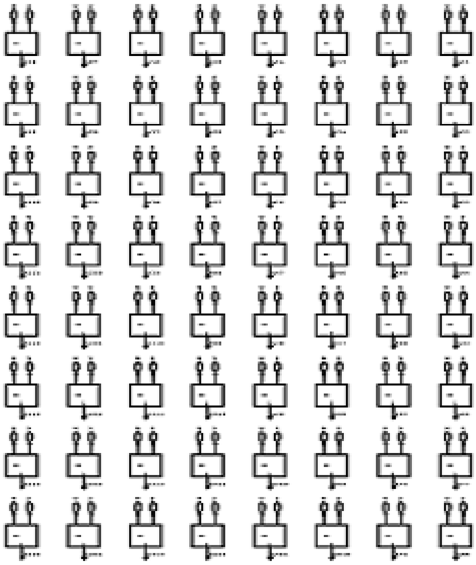


FIG: PP Generation Using And Array

Compressors for Partial Product Reduction:

In the proposed architecture, partial product reduction is accomplished by the use of 4:2, 5:2 compressor structures and the final stage of addition is performed by a Sklansky adder.

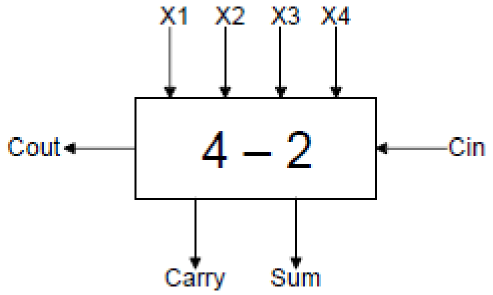
This multiplier architecture comprises of a partial product generation stage, partial product reduction stage and the final addition stage. The latency in the Wallace tree multiplier can be reduced by decreasing the number of adders in the partial products reduction stage. In the proposed architecture, multi bit compressors are used for realizing the reduction in the number of partial product addition stages. The combined factors of low power, low transistor count and minimum delay makes the 5:2 and 4:2compressors, the appropriate choice. In these compressors, the outputs generated at each stage are efficiently used by replacing the XOR blocks with multiplexer blocks . The select bits to the multiplexers are available much ahead of the inputs so that the critical path delay is minimized. The various adder structures in the conventional architecture are replaced by compressors.

In high-speed designs, the Wallace tree construction method is usually used to add the partial products in a tree-like fashion in order to produce two rows of partial products that can be added in the last stage. The Wallace tree is fast since the critical path delay is proportional to the logarithm of the number of bits in the multiplier. There exist a handful of ways to construct the Wallace Tree. The prominent method considers all the bits in each column at a time and compresses them into two bits (a sum and a carry).

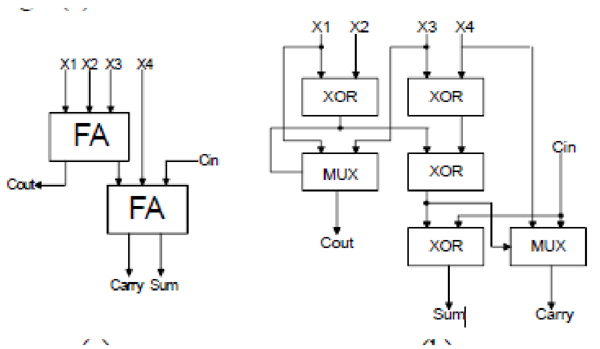
The Wallace tree is constructed by considering all the bits in each fours row at a time and compressing them in an appropriate manner. Thus, compressors form the essential requirement of high speed multipliers. The speed, area and power consumption of the multipliers will be in direct proportion to the efficiency of the compressors. Thus, in order to satisfy the requirement of small area low power high throughput circuitries, this paper provides novel designs of 4:2 and 5:2 compressors with minimum number of transistors. The proposed designs are highly efficient in terms of small area low power.

4-2 Compressor:

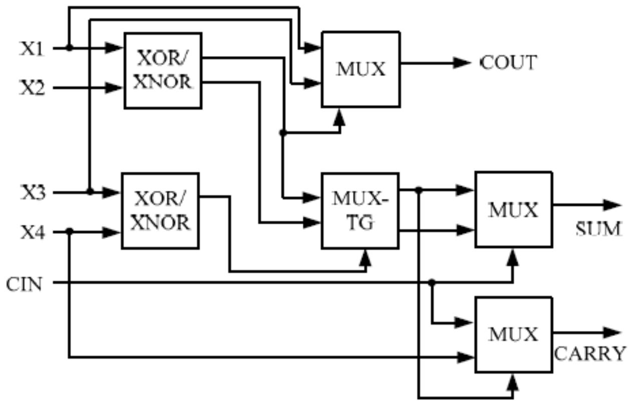
The 4-2 compressor has 4 inputs X1, X2, X3 and X4 and 2 outputs Sum and Carry along with a Carry-in (Cin) and a Carry-out (Cout) as shown in Fig 5. The input Cin is the output from the previous lower significant compressor. The Cout is the output to the compressor in the next significant stage.



The standard implementation of the 4-2 compressor is done using 2 Full Adder cells as shown in fig.



Thus replacing some XOR blocks with multiplexers results in a significant improvement in delay below



The equations governing the outputs in the proposed architecture are shown below

$$Sum=(x1\oplus x2)\bullet\overline{x3\oplus x4}+\overline{(x1\oplus x2)}\bullet(x3\oplus x4)\bullet\overline{Cin}+$$

$$\overline{(x1\oplus x2)\bullet\overline{x3\oplus x4}+\overline{(x1\oplus x2)}\bullet(x3\oplus x4)}\bullet Cin$$

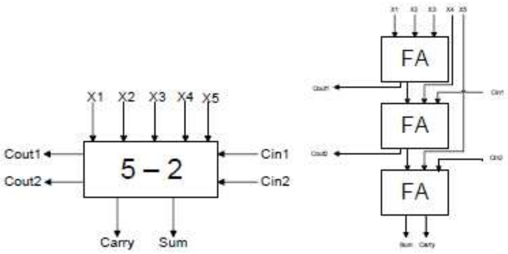
$$Cout=(x1\oplus x2)\bullet x3+\overline{(x1\oplus x2)}\bullet x1$$

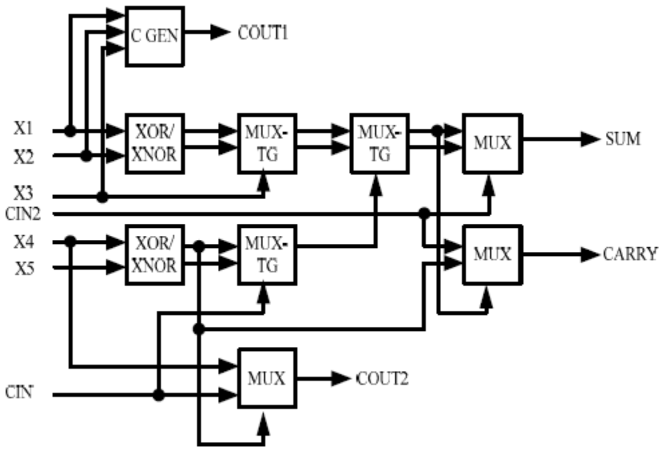
$$Carry=(x1\oplus x2\oplus x3\oplus x4)\bullet Cin+$$

$$\overline{(x1\oplus x2\oplus x3\oplus x4)}\bullet x4$$

5-2 Compressor:

The 5-2 Compressor block has 5 inputsX1,X2,X3,X4,X5 and 2 outputs, Sum and Carry, along with 2 input carry bits (Cin1, Cin2) and 2 output carry bits (Cout1,Cout2) as shown in Fig.a. The input carry bits are the outputs from the previous lesser significant compressor block and the output carry are passed on to the next higher significant compressor block.





In the proposed architecture these outputs are utilized efficiently by using multiplexers at select stages in the circuit. Also additional inverter stages are eliminated. This in turn contributes to the reduction of delay, power consumption and transistor count (area). The equations governing the outputs are shown below.

$$SUM=X1\oplus X2\oplus X3\oplus X4\oplus X5\oplus CIN\oplus CIN2$$

$$COUT1=(X1+X2)\bullet X3+X1\bullet X2$$

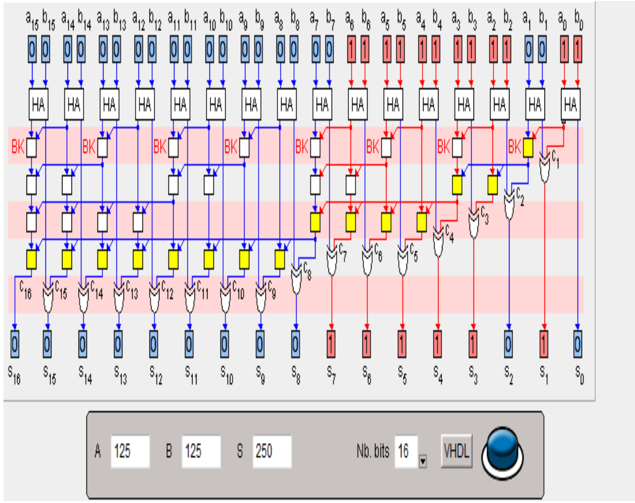
$$COUT2=(X4\oplus X5)\bullet CIN+\overline{(X4\oplus X5)}\bullet X4$$

$$CARRY=((X1\oplus X2\oplus X3)\oplus(X4\oplus X5\oplus CIN))\bullet CIN2+$$

$$\overline{((X1\oplus X2\oplus X3)\oplus(X4\oplus X5\oplus CIN))}\bullet(X1\oplus X2\oplus X3)$$

Sklansky Tree Adder

To design fast adders, binary trees of "BK" cells will first generate simultaneously all the carries ci. The "Sklansky's adder" builds recursively 2-bit adders then 4-bit adders, 8-bit adders, 16-bit adder and so on by abutting each time two smaller adders. The architecture is simple and regular, but suffers from fan-out problems. Besides in some cases it is possible to use less "BK" cells with the same addition delay.

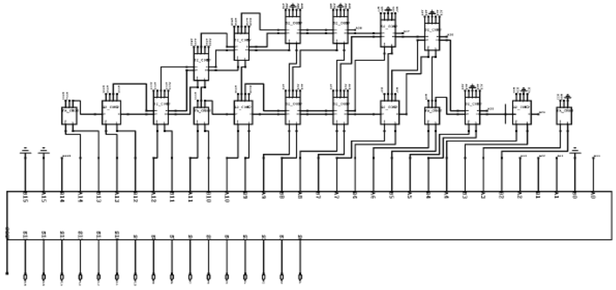

a15 b15 a14 b14 a13 b13 a12 b12 a11 b11 a10 b10 a9 b9 a8 b8 a7 b7 a6 b6 a5 b5 a4 b4 a3 b3 a2 b2 a1 b1 a0 b0
HA
BK
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16
s16 s15 s14 s13 s12 s11 s10 s9 s8 s7 s6 s5 s4 s3 s2 s1 s0
VHDL

A 125 B 125 S 250 Nb. bits 16

The output bits s_i = a_i Å b_i Å c_i. Now a_i Å b_i = '1' if the "HA" cell output equals 'P' . Thus the "HA" cell computes a_i Å b_i and subsequently s_i is given by one "XOR" gate. The "BK" cells that output the carries c_i never output the value 'P', consequently they can be simplified. Those "BK" cells are in yellow.the BK cell architecture.

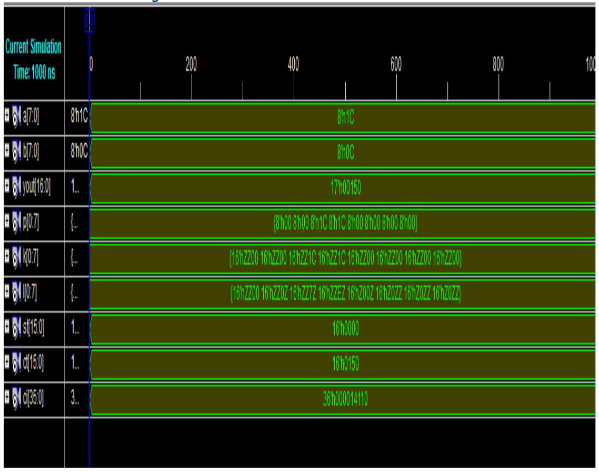
Architecture of Proposed Wallace Tree Multiplier

Our proposed architecture aims to reduce the overall latency. This leads to increased speed and reduced power consumption. The design makes use of compressors in place of full adders, and the final carry propagate stage is replaced by a Sklansky tree adder.



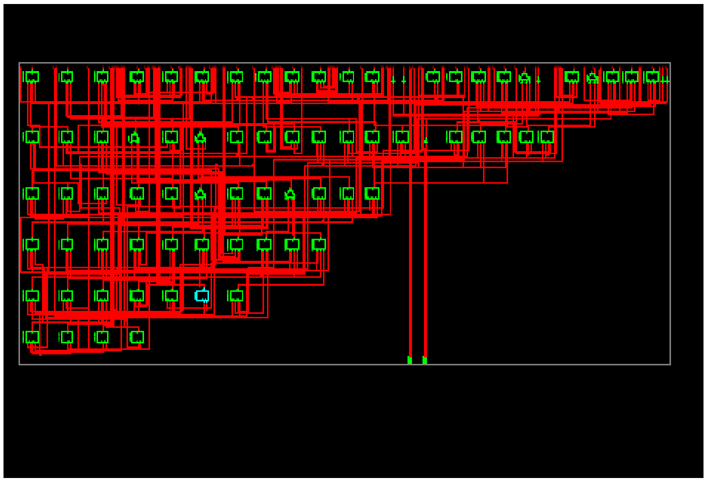
The first stage consisting of a full adder. In the second stage, two full adders have been grouped and implemented using a 4:2 compressor. Similarly, the third stage consists of a 5:2 compressor, which is a combination of 3 full adders and so on. In this manner, the individual full adder blocks in the original structure are grouped and implemented using compressors. The number of interconnections is taken care of, since they play a vital role in the flow of carry from one stage to the next in the tree. From Fig. 12, we can see that the longest delay path of our design is the one consisting of two 5:2 compressors, which produces a reduced latency of 8 (four per compressor) only. The use of the Sklansky adder in the structure further results in a reduced latency of 6 with a latency of 1 for the AND array. Hence, this novel structure brings down the overall latency count to 15. Thus, a significant latency reduction of 44.4% than the conventional counterpart is realized. The symbolic arrangement of the proposed structure is depicted in Fig. 13 for elaboration.

Result Analysis

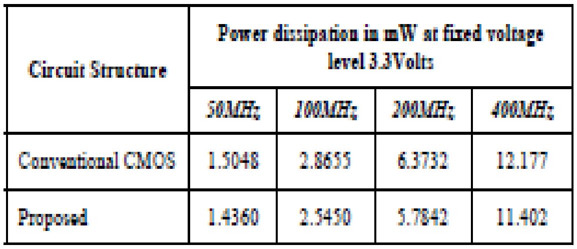


```
a[7:0]      8'h1C   8'h1C
b[7:0]      8'h0C   8'h0C
yout[16:0]  1...    17'h00150
p[0:7]      {...    {8'h00 8'h00 8'h1C 8'h1C 8'h00 8'h00 8'h00 8'h00}
k[0:7]      {...    {16'hZZ00 16'hZZ00 16'hZZ1C 16'hZZ1C 16'hZZ00 16'hZZ00 16'hZZ00 16'hZZ00}
l[0:7]      {...    {16'hZZ00 16'hZZ0Z 16'hZZ7Z 16'hZZEZ 16'hZ00Z 16'hZ0ZZ 16'hZ0ZZ 16'hZ0ZZ}
st[15:0]    1...    16'h0000
ct[15:0]    1...    16'h0150
ci[35:0]    3...    36'h000014110
```

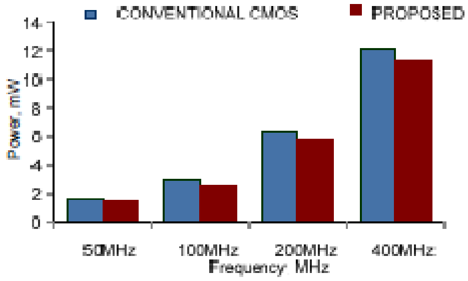
Simulation Result



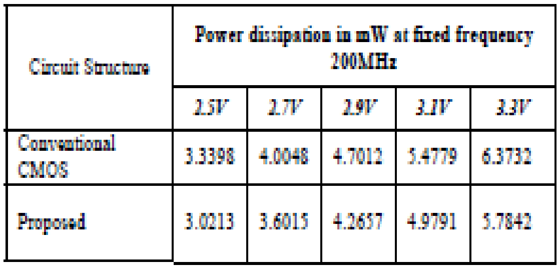
Comparision of Conventional & Proposed Wallace Tree Multipliers Power Comparision at Different Frequencies



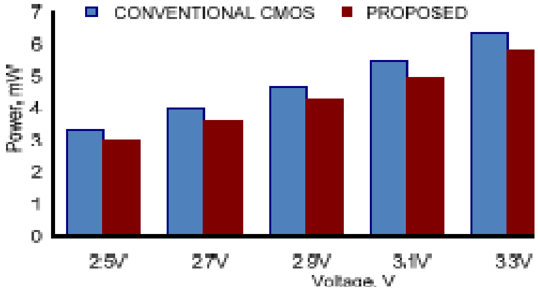
Circuit Structure	Power dissipation in mW at fixed voltage level 3.3Volts			
	50MHz	100MHz	200MHz	400MHz
Conventional CMOS	1.5048	2.8655	6.3732	12.177
Proposed	1.4360	2.5450	5.7842	11.402



Power Comparision At Different Voltages



Circuit Structure	Power dissipation in mW at fixed frequency 200MHz				
	2.5V	2.7V	2.9V	3.1V	3.3V
Conventional CMOS	3.3398	4.0048	4.7012	5.4779	6.3732
Proposed	3.0213	3.6015	4.2657	4.9791	5.7842



Conclusion

In this paper, the implementation and analysis of a novel Wallace tree architecture is proposed. The latency of existing Wallace tree multiplier which is found to be 27 has been reduced to 15.The comparison result also shows that a significant reduction of power is achieved. At an operating frequency of 50 MHz at 3.3V, the power is found to be 1.436mW. It is a realization of 4.57% of power reduction than the conventional Wallace tree multiplier. At 400MHz, the power consumed is found to be 11.402mW, which is a 6.36% reduction of that obtained from the existing architecture. The results prove that the proposed architecture is more efficient than the conventional one in terms of power consumption and latency.

The advantage of high speed becomes an enhanced feature for multipliers having operand of greater than 16 bits. For real-time signal processing, a high speed and throughput Multipliers-Accumulator (MAC) is always a key to achieve high performance in the digital signal processing system.

References

[1] List I. Abdellatif, E. Mohamed, “Low-Power Digital VLSI Design, Circuits and Systems,” Kluwer Academic Publishers, 1995.

[2] H. Neil. Weste and Kamran Eshraghian, “Principles of CMOS VLSIdesign-A Systems Perspective,” Pearson Edition Pvt Ltd. 3rd edition, 2005.

[3] Sreehari Veeramachaneni, Kirthi M, Krishna Lingamneni Avinash Sreekanth Reddy Puppala M.B. Srinivas, “Novel Architectures for High-Speed and Low-Power 3-2, 4-2 and 5-2 Compressors,” 20th International Conference on VLSI Design, Jan 2007, Pp. 324-329.

[4] K. Prasad and K. K. Parhi, “Low-power 4-2 and 5-2 compressors,” inProc. of the 35th Asilomar Conf. on Signals, Systems and Computers, 2001, Vol. 1, pp. 129–133.

[5] Perneti Balasreekanth Reddy and V. S. Kanchana Bhaaskaran, Design of Adiabatic Tree Adder Structures for Low Power, International Conference on Embedded Systems (ICES 2010) organized by CIT, Coimbatore and Oklohoma State University, 14-16 July 2010