An Optimized Wallace Tree Multiplier using Parallel Prefix Han-Carlson Adder for DSP Processors

T.N.Priyatharshne
ME – VLSI DESIGN
Angel College of Engg and Tech
Tirupur , TN, India

Prof. L . Raja, M.E , (Ph.D)
Professor, ECE DEPT
Angel College of Engg and Tech
Tirupur , TN, India

A . Vinodhini
ME – VLSI DESIGN
Angel College of Engg and Tech
Tirupur , TN, India

Abstract-- **Multipliers play a key role in the high performance digital systems and DSP applications. Many attempts have been made to reduce the number of partial products generated to increase the speed in a multiplication process one of them is Wallace tree multiplier. It is an improved version of tree based multiplier. The parallel multipliers do the computations using lesser iterative steps and reduce the complexity as compared to the serial multipliers. It uses Han–Carlson adder algorithm to reduce the latency. It is constructed with the help of 4:2 and 5:2 compressors. The proposed method is faster than the conventional CMOS method, and power consumption realization at 200MHz. The simulations have been carried out using the pyxis v10.1 EDA tool.**

Index Terms-- **Wallace Tree, Han-Carlson adder, Low power VLSI, Compressors, Multiplier.**

I. INTRODUCTION

High performance multiplier is the important part of the CPU and DSP. The multiplier's speed usually determines the processor's speed. The multiplier is one of the key hardware blocks in most of the digital and high performance systems such as digital signal processors and microprocessors. With the recent advances in technology, many researchers have worked on the design of increasingly more efficient multipliers. They aim at offering higher speed and lower power consumption even while occupying reduced silicon area. However, the fact remains that the area and speed are two conflicting performance constraints. Hence, innovating increased speed always results in larger area. In this paper, we arrive at a better trade-off between the two, by realizing a marginally increased speed performance through a small rise in the number of transistors. The new architecture enhances the speed performance of the widely acknowledged Wallace tree multiplier. The structural optimization is performed on the conventional Wallace multiplier, in such a way that the latency of the total circuit reduces considerably. The Wallace tree basically multiplies two unsigned integers. The conventional Wallace tree multiplier architecture comprises of an AND array for computing the partial products, a carry save adder for adding the partial products so obtained and a carry propagate adder in the final stage of addition.

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 Han - Carlson adder.

II. WALLACE TREE MULTIPLIER

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:

A. Partial Product Generation Stage
B. Partial Product Reduction Stage
C. Partial Product Addition Stage

A. 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. 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. 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.

B. Partial Product Reduction Stage

The design analysis starts with the analysis of the elementary algorithm for multiplication by Wallace Tree multiplier. Figure 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. Fig. 1. shows the 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. 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.

C. Partial Product Addition Stage

Partial product generation stage is obtained using AND array, partial product reduction is accomplished by the use of 3:2, 4:2, 5:2 compressor structures and the final stage of addition is performed by a Han-Carlson adder. Partial product addition stage is the important stage for Wallace tree multiplier. It reduces the complexity and latency. It has Log N+1 stages. It has less fanout. It will increase the performance and speed of the Wallace tree multiplier.

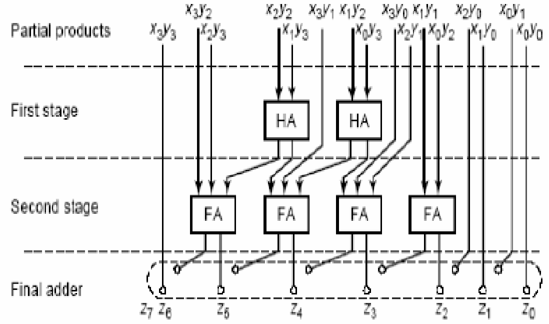

Fig. 1. Wallace Tree Multiplier

III. COMPRESSOR FOR PARTIAL PRODUCT REDUCTION

The 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, 4:2 and 3:2 compressors, 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.

A. *3:2 Compressor*

A 3-2 compressor takes 3 inputs X1 , X2 , X3 and generates 2 outputs, the sum bit S, and the carry bit C. The compressor is governed by the basic equation

$$X1 + X2 + X3 = Sum + 2*Carry.$$

We can see the fact that both the XOR and XNOR values are computed is efficiently used to reduce the delay by replacing the second XOR with a MUX. This is due to the availability of the select bit at the MUX block before the inputs arrive. Thus the time taken for the switching of the transistors in the critical path is reduced.

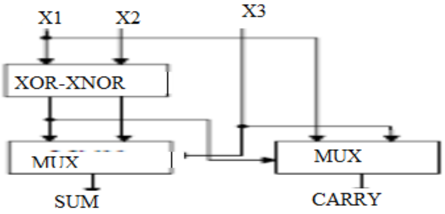

Fig. 2. 3:2 Compressor

B. *4-2 Compressor*

The 4:2 compressor structures actually compress five partial products bits into three [1, 2, 3]. The architecture is connected in such a way that four of the inputs are coming from the same bit position of the weight j while one bit is fed from the neighboring position known as carry-in. The output of 4:2 compressor consists of one bit in the position j and two bits in the position .This structure is called compressor since it compresses four partial products into two. A 4-2 compressor can also be built using 3-2 compressors. It consists of two 3-2 compressors (full adders) in series and involves a critical path of 4 XOR delays as shown in Fig. 3. An alternative implementation is shown in Figure. This implementation is better and involves a critical path delay of three XOR's, hence reducing the critical path delay by 1 XOR. The output Cout, being independent of the input Cin accelerates the carry save summation of the partial products.

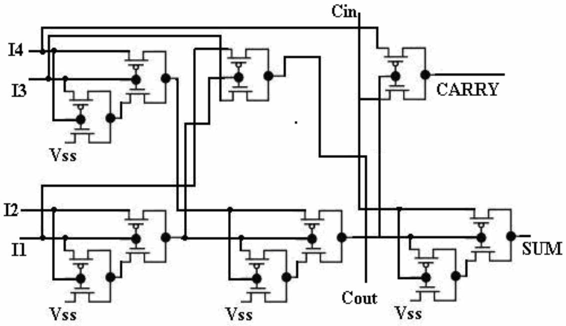

Fig. 3. 4:2 Compressor

C. *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). 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. The architecture is connected in such a way that five of the inputs come from the same bit position of the weight while other two input are fed from the neighboring position known as carry-in. The outputs of 5:2 compressor consists of one bit in the position sum and two bits in the position cout1, cout2, carry. A simple implementation of the (5:2) compressor is to cascade three (3:2) full adders in a hierarchical structure, as shown in Fig. 4. This architecture has a critical path delay of 6 XOR gates. Fig. 4 shows architecture of a (5:2) compressor. The implementation shows that this design has a critical path delay of 4XOR + 1MUX unlike the conventional implementation with a delay of 5XOR.

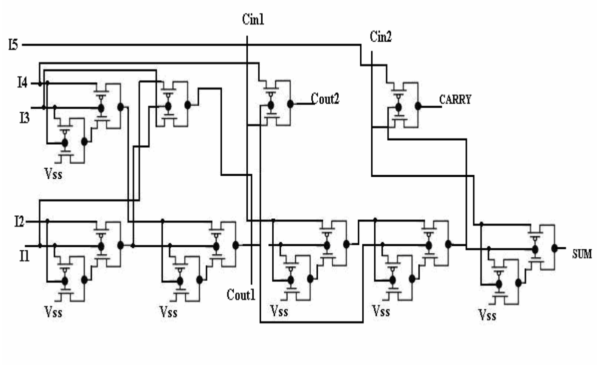

Fig. 4. 5:2 Compressor

IV. Han- Carlson Adder

The Han-Carlson trees are a family of networks between Kogge-Stone and Brent-Kung. The logic performs Kogge-Stone on the odd numbered bits and then uses one more stage to ripple into the even positions. In Han-Carlson adder using transmission gate, 312 transistors are used, delay is 60.18e – 09s and Power is 1.6178w. The final stage in the Wallace tree multiplier for addition of partial products can be further reduced by the use of tree adders. The use of tree adders primarily reduces the power consumption. Furthermore, it also accounts for increased speed of operation. The basic concept of tree adders extend from the idea of carry look-ahead computation and the class of parallel carry look-ahead schemes. These structures target high-performance applications. Here, the Han-Carlson type of tree adder is preferred due to its lower power consumption than that incurred by other tree adder structures. Furthermore, the latency of Han-Carlson adder is reduced, which is less than that realized by Brent Kung, and Kogge Stone tree adder circuits. It is more efficient and Suitable for VLSI implementation. It gives a good overview of prefix addition formulation, and presents their own hybrid synthesis of the Ladner-Fisner and Kogge Stone adder graphs. Again this trades an increase in logical depth for a reduction in fanout. It is a effectively a higher radix variant of the kogge-stone. It has Log N+1 stages. It has less fanout. It trades logical length for wire length.

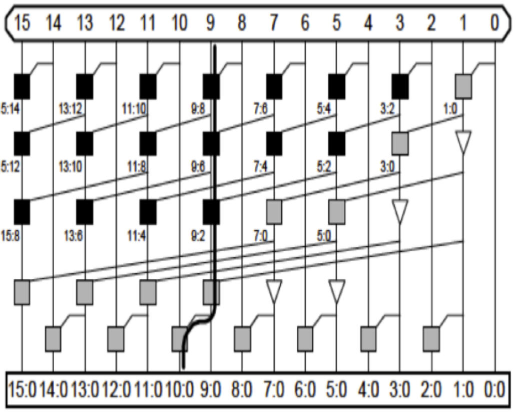

Fig. 5. Han-Carlson Adder Structure

V. Conventional and Proposed Wallace Tree Multipliers

In the conventional 8 bit Wallace tree multiplier design, more number of addition operations is required. Using the carry save adder, three partial product terms can be added at a time to form the carry and sum. The sum signal is used by the full adder of next level. The carry signal is used by the Adder involved in the generation of the next output bit, with a resulting overall delay proportional to $\log_{3/2}n$, for n number of rows. A multiplier consists of various stages of full adders, each higher stage adds up to the total delay of the system. In the first and second stages of the Wallace structure, the partial products do not depend upon any other values other than the inputs obtained from the AND array. However, for the immediate higher stages, the final value (PP3) depends on the carry out value of previous stage. This operation is repeated for the consecutive stages. Hence, the major cause of delay is the propagation of the carry out from the previous stage to the next stage. In conventional Wallace tree structure, the total number of stages in the critical path sums up to 13. Each full adder accounts for a latency of 2. Therefore, the total latency of the given structure when calculated is 26. The latency count gets added by one, when considering the AND array, thus resulting in a total latency of 27.

Our proposed architecture aims to increase speed and reduce power consumption. Overall latency is reduced using parallel prefix adder. The design makes use of compressors in place of full adders, and the final carry propagate stage is replaced by a Han-Carlson 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 3rd stage consists of a 5:2 compressor.

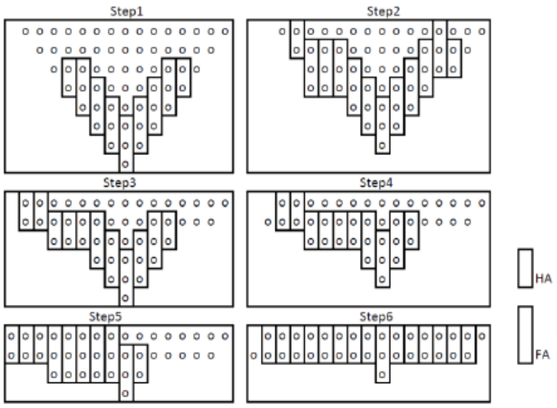

Fig. 6. logic used in Wallace tree multiplier

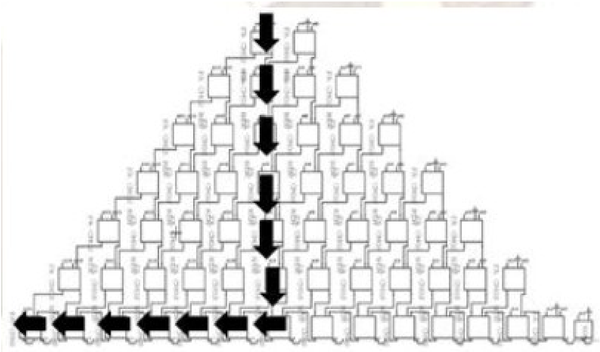

Fig. 7. Schematic of Wallace tree multiplier depicting critical path

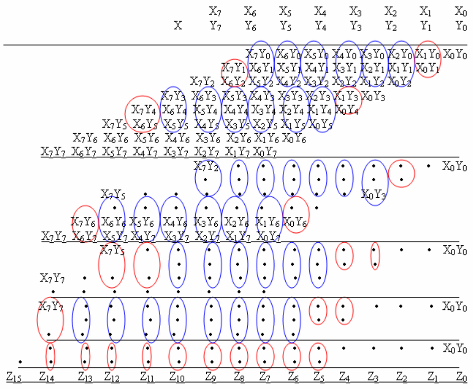
X
Y
Z_{15} Z_{14} Z_{13} Z_{12} Z_{11} Z_{10} Z_9 Z_8 Z_7 Z_6 Z_5 Z_4 Z_3 Z_2 Z_1 Z_0

Fig. 8. 8*8 wallace tree Multipication

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. 8, 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 Han-Carlson tree adder in the structure further results in a reduced latency of 6 with a latency with fanout of 2 and log N+1 stages. Hence, this novel structure brings down the overall latency count. Thus, a significant latency reduction is obtained. The symbolic arrangement of the proposed structure is depicted in fig. 9.

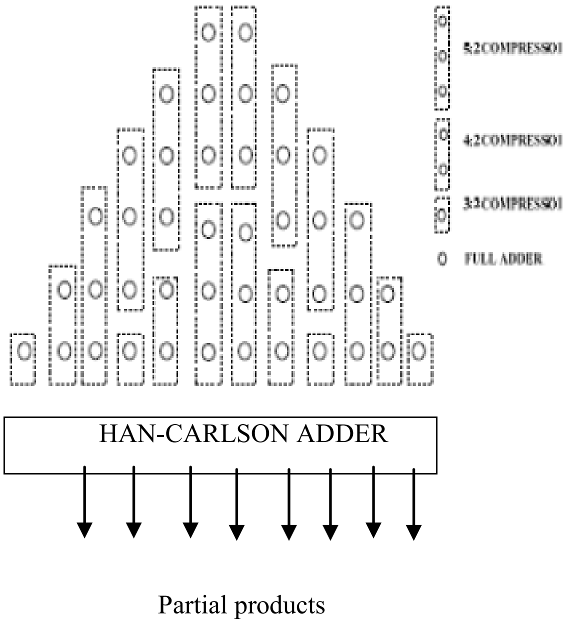

Fig. 9. Proposed Wallace tree multiplier

VI. Simulation Results And Discussion

In this section, the proposed and the conventional architectures have been compared. The latency defines the number of total phases required to compute the output and is found to be less than the latency of the conventional Wallace tree multiplier. Table shows the delay comparison in nano seconds and the power consumption of the conventional and proposed multipliers operated at 200MHz for various supply voltage levels. That high speed is achieved by introducing parallel multiplier architecture to achieve high speed. Instead of using carry save adders in this multiplier, full adders and half adders of 4:2 compressors and 3:2 compressors can be used in their reduction phase so that the complexity is reduced.

VII. Conclusion

In this paper, An Optimized Wallace Tree Multiplier using Parallel Prefix Han-Carlson Adder is proposed. The latency of existing Wallace tree multiplier has been reduced .The comparison result also shows that a significant reduction of power is achieved. At an operating frequency of 200 MHz. Each layer of the tree reduces the number of vectors by a factor of 3:2. Minimum propagation delay. Reduce the number of sequential adding stages. The computation time of the Wallace tree has achieved the lower bound of O (log3/2 N). For n-bit Wallace tree multiplier, the number of steps needed is (log3/2(n/2) + 1). Wallace tree have significant complexity and timing advantages over traditional matrix multipliers. The results prove that the proposed architecture is more efficient than the conventional one in terms of power consumption and latency.

Table I. Comparison of conventional and proposed Wallace tree multiplier

Circuit structure	Parameters					
	w	Delay (ns)	Power (mw)	Leak power (uw)	Dynamic power (uw)	Power diss (mw)
Existing	8	22.48	177.98	5.178	74.642	5.784
Proposed	8	20.98	155.64	4.97	71.132	5.500

REFERENCES

[1] V.G. Oklobdzija, D. Villeger, S. S. Liu, “A Method for Speed Partial Product Reduction and Generation of Fast Parallel Multipliers Using and Alghoritmic Approach”, IEEE Transaction on computers, Vol. 45, No 3, March 1996.

[2] P. Stelling, C. Martel, V. G. Oklobdzija, R. Ravi, “Optimal Circuits for Parallel Multipliers”, IEEE Transaction on Computers, Vol. 47, No.3, pp. 273-285, March, 1998.

[3] V. Oklobdzija, "High-Speed VLSI Arithmetic Units: Adders and Multipliers", in "Design of High-Performance Microprocessor Circuits", Book Chapter, IEEE Press, 2000.

[4] K. Prasad, and K.K. Parhi, “Low-power 4-2 and 5-2 compressors,” in Proc. of the 35th Asilomar Conference on Signals, Systems and Computers, vol.I, pp. 129-133,2001.

[5] H. T. Bui, A. K. Al-Sheraidah, “Design and analysis of 10-transistor full adders using novel XOR-XNOR gates,” in Int. Conf. Signal Processing 2000 (World Computer congress) Beijing, China, Aug. 2000.