HIGH SPEED WALLACE TREE MULTIPLIER USING COMPRESSOR

Sethulakshmi R
M.Tech(VLSI &Embedded Systems)
Mangalam College of Engineering,
Kottayam
Sethu1890@hotmail.com

Greeshma R
Assistant Professor
Mangalam College of Engineering,
Kottayam

Abstract- **Designing multipliers that are of high-speed, low power, and regular in layout are of substantial research interest. Speed of the multiplier can be increased by reducing the generated partial products. Many attempts have been made to reduce the number of partial products generated in a multiplication process one of them is Wallace tree multiplier. Wallace Tree CSA structures have been used to sum the partial products in reduced time. In this paper Wallace tree construction is investigated and evaluated. Speed of traditional Wallace tree multiplier can be improve by using compressor techniques. In this paper Wallace tree is constructed by traditional method and with the help of compressor techniques such as 4:2 compressor, 5:2 compressor, 6:2 compressor, 7:2 compressor. Therefore, minimizing the number of half adders used in a multiplier reduction will reduce the complexity.**

Keywords-Multiplier, Multiplication, Wallace Tree Multiplier, Compressor techniques

I INTRODUCTION

The concept of digital data manipulation has made a dramatic impact on our society. One has long grown accustomed to the idea of digital computers. Evolving steadily from mainframe and minicomputers, personal and laptop computers have proliferated into our daily lives. In the majority of applications utilizing efficient implementation of generic arithmetic logic units and floating point units to execute dedicated algorithms, multipliers have been a critical and obligatory component in dictating the overall circuit performance when constrained by power consumption and computation speed.

Multiplication is basically a two-step process, essentially consisting of the formation of partial products followed by its reduction to give final binary result. The formation of partial products is a simple process, whereas the summation of partial products contributes to most of the delay and area of the multiplier. One method used to increase the performance is to use encoding techniques, like modified Booth encoding, to reduce the number of partial products generated . However, to achieve even higher performance, advanced hardware multiplier architectures search for faster and more efficient methods for summing the partial products. A powerful technique for improving the performance of partial product reduction is to use carry save adders (CSAs) in Wallace and Dadda trees.

A significant departure from carry save adder arrangement was achieved with the introduction of compressors that involve limited carry propagation .

This paper addresses high-level optimization techniques for Wallace Tree Multiplier multipliers. High-level techniques refer to algorithm and architecture level techniques that consider multiplication's arithmetic features and input data characteristics. One of the important algorithm presents in paper for VLSI implementable multiplication is Wallace Tree Multiplier using Xilinx.

II. MULTIPLIER

A basic multiplier can consist of three parts (i) partial product generation (iii) partial product addition and (iii) final addition [9]. A multiplier essentially consist of two operands, a multiplicand

"Y" and a multiplier "X" and produces a product ". In the first stage, the multiplicand and the multiplier are multiplied bit by bit to generate the partial products product The second stage is the most important, as it is the most complicated and determines the speed of the overall multiplier to add these partial product to generate the Product "P". This paper will be focused on the optimization of this stage, which consists of the addition of all the partial products. If speed is not an issue, the partial products can be added serially, reducing the design complexity. However, in high speed design, 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. Although fast, since its critical path delay is proportional to the logarithm of the number of bits in the multiplier, the Wallace tree introduces other problems such as wasted layout area and increased complexity. In the last stage, the two-row outputs of the tree are added using any high-speed adder such as carry save adder to generate the output result.

III. WALLACE TREE MULTIPLIER

Digital multipliers can be classified into serial, parallel and serial-parallel multipliers. Parallel multipliers are again of two types, Array multipliers and Tree multipliers. Wallace tree multiplier as the name itself indicates belongs to the tree multiplier category. Professor Christopher Stewart Wallace (26 October 19 33 – 7 August 2004) an Australian computer scientist (and Physicist who also contributed to a variety of other areas) 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.

In the Wallace Tree method, three bit signals are passed to a one bit full adder ("3W") which is called a three input Wallace Tree circuit, and the output signal (sum signal) is supplied to the next stage full adder of the same bit, and the carry output signal thereof is passed to the next stage full adder of the same no of bit, and the carry output signal thereof is supplied to the next stage of the full adder located at a one bit higher position.

A 8-bit multiplier is constructed by using Wallace tree architecture .The architecture has been shown in Figure 3. Partial products are added in 6 steps.

In the Wallace Tree method, the circuit layout is not easy although the speed of the operation is high since the circuit is quite irregular. The delay generated in Wallace tree multiplier can be further reduced by using modified tree structures called compressors.

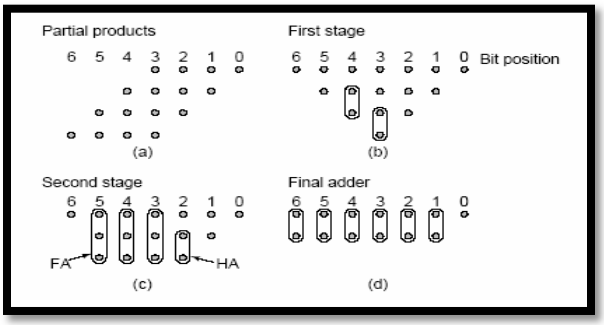

Fig.1. Logic used in 4 bit Wallace Tree Multiplier

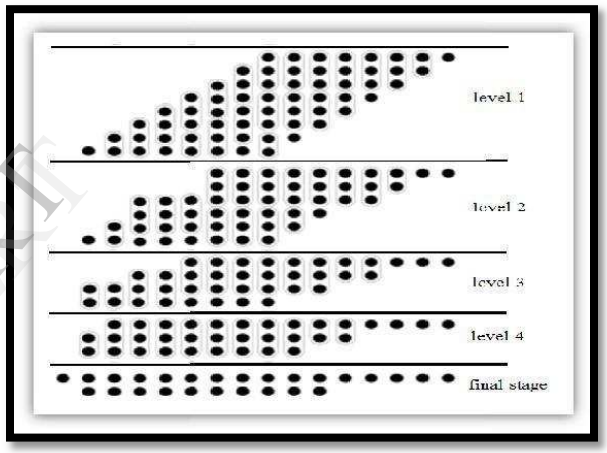

Fig.2.Implementation of 8×8 Wallace tree multiplier

IV. COMPRESSOR

Compressors are building blocks used for accumulating partial products during the multiplication process. The basic idea in an n : 2 compressor is that n operands can be reduced to two, by doing the addition while keeping the carries and sums separate. This means that all of the columns can be added in parallel without relying on the result of the previous column, creating a two-output adder with a time delay that is independent of the size of its inputs. The full adder is the most primitive compressor and is often referred to as the 3 : 2 compressor since it compresses three operands into two Sum and Carry.This 3: 2 compressor has a delay of two XOR gates, and is normally used in carry-save form to sum up the partial products in a multiplier tree. Though the addition used in this manner is much

faster than that of a ripple carry adder, the interconnections are very irregular thereby making the structure more complex.

1) 4:2 Compressor

The 4:2 compressor structure actually compresses five partial products bits into three. 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 j-1(known as carry-in). The outputs of 4:2 compressor consists of one bit in the position j and two bits in the position j+1.This structure is called compressor since it compresses four partial products into two(while using one bit laterally connected between adjacent 4:2 compressors). Figure 3 shows the block diagram of 4-2 compressor.

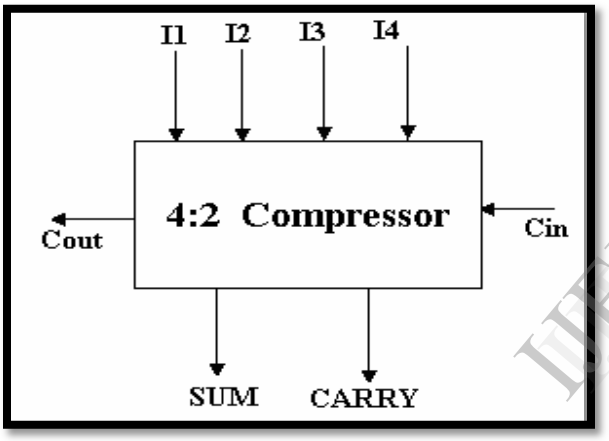

Fig.3. Block Diagram of 4:2 Compressor

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 Figure 4. An alternative implementation is shown in Figure 4. 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 product.

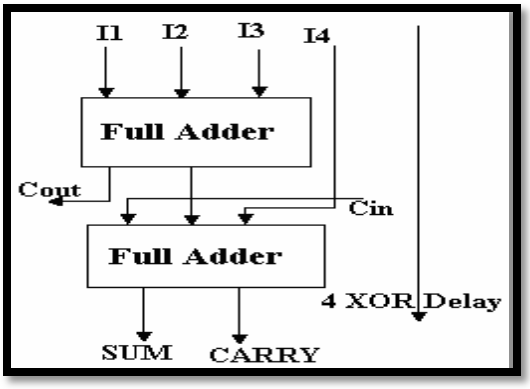

Fig. 4. 4:2 Compressor Design using Full Adders

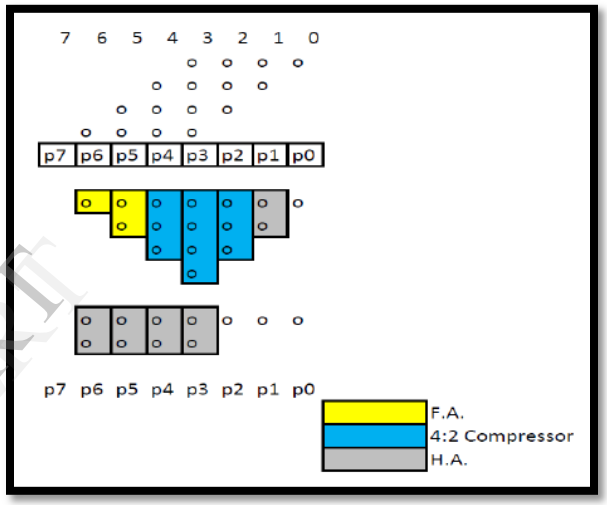

Fig.5. Logic used in 4 bit Wallace Tree Multiplier using 4:2 Compressor

2)5:2 Compressor

The block diagram of a (5:2) compressor shown in Figure 6 has seven inputs and four outputs. Five of the inputs are the primary inputs I1, I2, I3,I4 and I5 and two other inputs, C_{inl} and C_{in2} . The architecture is connected in such a way that five of the inputs come from the same bit position of the weight j while other two inputs (C_{inl} and C_{in2}) are fed from the neighboring position j-1(known as carry-in). The outputs of 5:2 compressor consists of one bit in the position j (sum) and two bits in the position j+1 (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 Figure 4. This architecture has a critical path delay of 6 XOR gates. Figure 6 shows another 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.

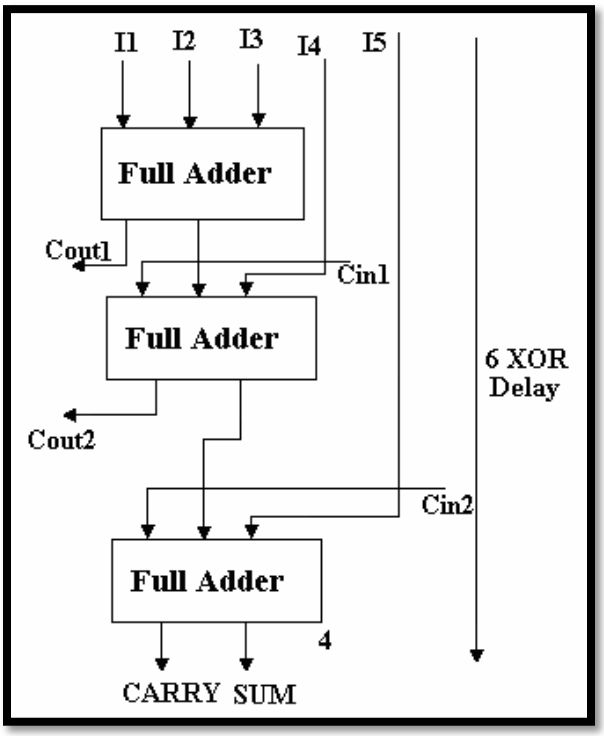

Fig.6. 5:2 Compressor using Full Adders

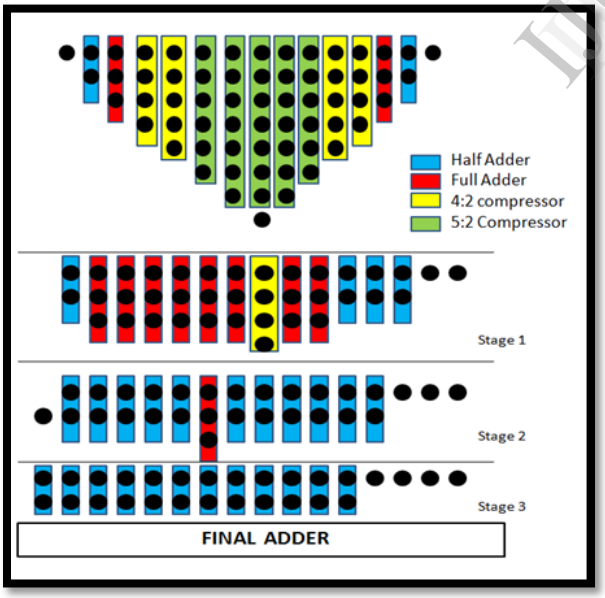

Fig.7.Wallace using 5:2 compressor

3) 6:2 Compressor

The 6:2 compressor has 6 input bits and produces 2 sum output bits (out_0 and out_1), it also has a carry-in (C_{in0}, C_{in1}) and a carry-out (C_{out0}, C_{out1}) bit (thus, the total number of input/output bits are 8 and 4); All input bits, including C_{in0}, have rank 0 and C_{in1} has rank 1 ; the two output bits have ranks 0 and 1 respectively, while C_{out0} has rank 1 and C_{out1} has rank 2 as shown in Fig 8.

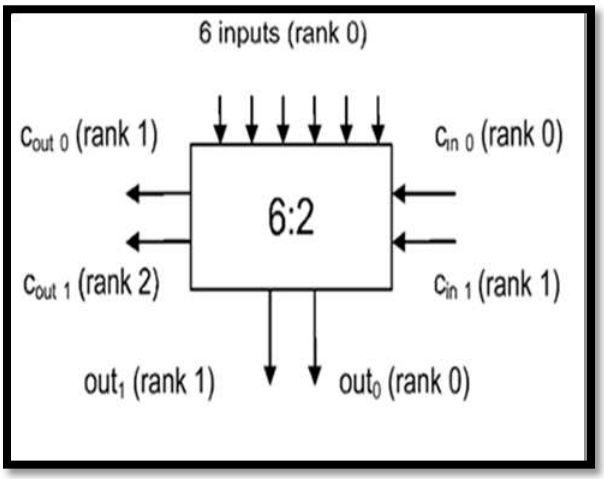

Fig.8(a) 6:2 compressor I/O diagram

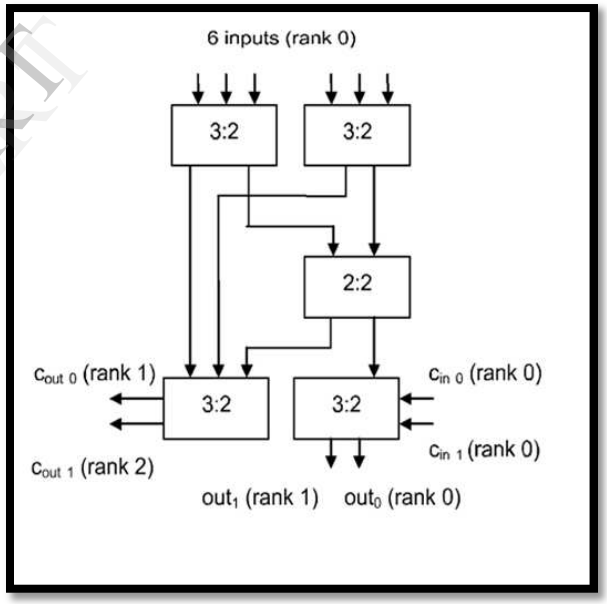

Fig.8(b) 6:2 compressor architecture

4) 7:2 Compressor

The 7:2 compressor has 7 input bits and produces 2 sum output bits (out_0 and out_1), it also has a carry-in (C_{in0}, C_{in1}) and a carry-out (C_{out0}, C_{out1}) bit (thus, the total number of input/output bits are 9 and 4); All input bits, including C_{in0}, have rank 0 and C_{in1} has rank 1 ; the two output bits have ranks 0 and 1 respectively, while C_{out0} has rank 1 and C_{out1} has rank 2 as shown in Fig 8.

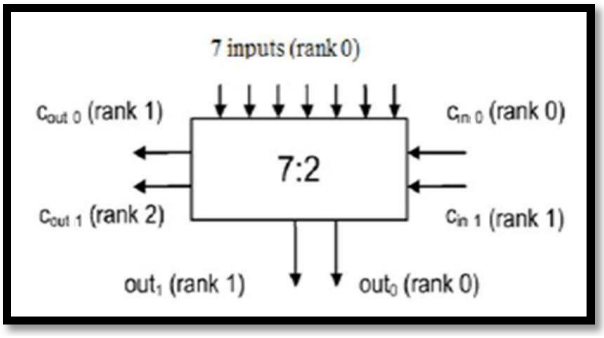

Fig.9. 7:2 compressor I/O diagram

V. VERIFICATION OF SIMULATION

For the overall analyses of the multipliers using Compressors, we carried out the following delay values and the simulation is carried out using Xilinx Tool. The delay results are exposed in Table 1

Wallace Tree multiplier (WTM)	No. of occupied slice	No. of LUT	Time delay (ns)
4 x 4 WTM	19	33	18.183
4 x 4 WTM using 4:2 compressor	15	27	19.324
4 x 4 WTM using 5:2 compressor	14	24	17.56
8 x 8 WTM	101	198	33.614
8 x 8 WTM using 5:2and 4:2 compressor	88	153	30.478
8 x 8 WTM using 6:2and 4:2 compressor	85	148	27.901
8 x 8 WTM using 7:2and 4:2 compressor	86	120	24.794

VII. CONCLUSIONS

The Wallace tree multipliers can be solved & analyzed using a new modified method of Wallace tree construction using compressors. The modified tree has a slightly smaller critical path, a slightly larger wiring overhead but gives high speed.. This modified design of multiplier which consist of 7:2 compressor ,6:2 compressor ,5:2 compressor ,4:2 compressor, 3:2 compressor, full adders and reduced no. of half adder and reduces the complexity and reduce the time delay.

VIII.REFERENCES

[1] Dursun Baran, Mustafa Aktan and Vojin G. Oklobdzija," Multiplier Structures for Low Power Applications in Deep-CMOS", IEEE International Symposium on Circuits and Systems (ISCAS), 2011

[2] S. A. Shinde, R. K. Kamat," FPGA based Improved Hardware Implementation of Booth Wallace Multiplier using Handel C", ELEKTRONIKA IR ELEKTROTECHNIKA,Page No.71-74, 2011

[3] P.V. Rao, Cyril Prassana Raj P,S. Ravi,"VLSI Design and Analysis of Multipliers for Low Power",IEEE 2009 Fifth International Conference On Intelligence Information Hiding and Multipedia Signal processing,pp. 1354-1357,2009

[4] Soojin Kim, Kyeongsoon Cho," Design of High-speed Modified Booth Multipliers Operating at GHz Ranges",World Academy of Science ,Engineering and Technology,2010.

[5] Sumit R. Vaidya, D.R. Dandekar,"Delay-power Performance Comparison of Multipliers in VLSI Circuit Design", IJCNC, vol. 2,No. 4,pp. 47-56,July 2010

[6] Issam S. Abu-Khater, Abdellatif Bellaouar, M. I. Elmasry, "Circuit Techniques for CMOS Low-Power High-Performance Multipliers", IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 31, NO. 10,Page No.1535-1546, OCTOBER 1996

[7] Prvinkumar G. Parate, Prafulla S. Patil, Dr (Mrs) S. Subbaraman "ASIC Implementation of 4 Bit Multipliers",IEEE First International Conference on Emerging Trends in Engineering and Technology, pp. 408-413,2008

[8] Kiaml Z. Pekmestzi," Multiplexer-Based Array Multipliers", IEEE TRANSACTIONS ON COMPUTERS, VOL. 48, NO. 1, JANUARY 1999.

[9] N. Ravi, A.Satish , Dr. T. Jayachandra Prasad, Dr. T. Subba Rao,"A New Design for Array multiplier with Trade Off in power and Area", IJCSI ,vol. 8,issue 3, May 2011.