32×32 Wallace Tree Multiplier Using Compressors and Sklansky Adder

Ms. Manjushri Tambe[1], Prof. R.H. Khade[2]
[1, 2] Pillai's Institute Of Information Technology

Abstract- *In the recent years in digital development, new techniques have been implemented to compensate between the three characteristics of any circuit i.e., area, power dissipation and delay. Multiplier is one of those circuits where you need to compromise between any two characteristics to achieve a better and speedy circuit. With the recent use of 3:2, 4:2 and 5:2 compressors, designers have been able to achieve better speed than the conventional Wallace tree multiplier. This paper shows designs of two higher level compressors: 6:2 compressor and 7:2 compressors. The paper will also describe two methods to design the internal carries of the higher level compressors. A 32x32 bit multiplier using 3:2, 4:2, 5:2, 6:2 and 7:2 compressor and a Sklansky adder has been designed. Comparison between the outputs of multiplier with and without use of Sklansky adder is shown. Hence, an input partitioning method, where instead of using the previous carry bits in the present column, directly the input bits from the present column are used to make the structure less dependable. The proposed architecture gives an increase of 65.40% speed than the conventional Wallace tree multiplier.*

Keywords- Wallace Tree Multiplier, Sklansky adder, compressor, Booth Algorithm

I. INTRODUCTION

Two basic ideas used behind the designing of compressors are: 1) use of associative law for addition and 2) use of mux instead of a full adder circuit. These both methods give less delay, dependency and so the increase in speed is achieved. The basic arithmetic operation include multiplication which is important in several microprocessors and digital signal processing applications where it is used to implement Digital Signal Processing algorithms such as convolution and filtering. The critical path consists mainly of the multiplier. The development of high level circuit techniques demand the need for high speed multipliers where the aim is to reduce the power dissipation of multipliers without compromising the speed and performance. Acceleration of the process can be achieved in three ways:

By reducing the number of summands;

By accelerating the summands formation;

By accelerating the summands addition. (Wallace, 1964) [1].

The multiplier can be divided into three parts and in each stage there are various methods to achieve faster response and less area.

1) Partial products generation stage, (use of BOOTH algorithm or AND array)
2) Partial products addition stage, and (array multipliers, Conventional Wallace tree)
3) The final addition stage. (Carry adders, e.g. carry propagate adder, parallel prefix adders, etc.)

The choice is made on the basis of a trade-off between area, speed and power consumption. [2]

In the partial product generation stage, the multiplicand and the multiplier are multiplied bit by bit to generate the partial products. The partial product addition stage is the most important and complicated stage and determines the overall speed of the multiplier. The 3:2, 4:2 and 5:2 compressors have been widely employed in the high speed multipliers to lower the latency of the partial product accumulation stage. The 3:2, 4:2 and 5:2 compressors are ideal for the construction of regularly structured Wallace tree with low level circuit complexity [1, 2]. Though these three compressors have high speed than an array multiplier, there is still a need to increase the speed even more. This leads to use of compressors with higher level i.e., 6:2 compressor and 7:2 compressor for high level circuit complexity. Efforts have been made to increase the compression level to the 8:3 and 9:3 compressors and some more.

II. LITERATURE SURVEY

Professor Christopher Stewart Wallace devised the Wallace Tree Multiplier algorithm in 1964. A simple array multiplier is considerably slower than the Wallace tree multiplier. The Wallace tree multiplier implements efficiently a digital circuit multiplying two integers.

The main features of Wallace Tree Multiplier are that it is an Optimized column adder tree. It combines all partial products into two vectors (carry and sum). The components used are half adder, full adder, Ripple carry adder and Carry propagating adder.

Disadvantages of conventional Wallace tree multiplier are:

1) In many FPGAs, there is no advantage provided by a Wallace tree over ripple adder trees.
2) They may actually be slower due to the irregular routing, and are more difficult to route certainly.
3) The size of Adder structure with the increase in multiplication bits.

Due to these reasons, a new method with changes in the conventional Wallace tree were made. 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). [3] There are many papers that show the use of low level compressors i.e. 3:2, 4:2 and 5:2. These compressors can be used for low complex circuits. But when we go to high level complex circuits, the speed starts to affect. So, now we have higher level compressors(6:2,7:2,8:3 and 9:3) that compress more number of bits than the low level compressors [5,6,7].

The final stage of the multiplier adds the two rows of the sum and a carry bit. There are different types of fast carry adders like ripple carry adder, carry propagate adder, Carry Save Adder, Carry Look-Ahead Adder, Carry Increment adder, Carry Skip Adder, Carry Bypass Adder, Carry Select Adder[8], parallel prefix adders. Parallel prefix structures are fast computers in comparison to other types of linear adders. Sklansky adder, Baugh- Wooley, Kogge -Stone, Han Carlson adder, Brent- Kung adder, Knowles [2, 1, 1, 1] adder, Ladner-Fischer adder, etc. [9] are some of the parallel prefix structures. Sklansky adder is preferred as it is faster compared to all the adders though its area requirement is more [9].

III. DESIGN AND IMPLEMENTATION

Since there are many papers that focus on the low level compressors (3:2, 4:2 and 5:2), we will see only the higher level (6:2 and 7:2) compressors and two different ways of designing the internal carries.

A. 6:2 Compressor

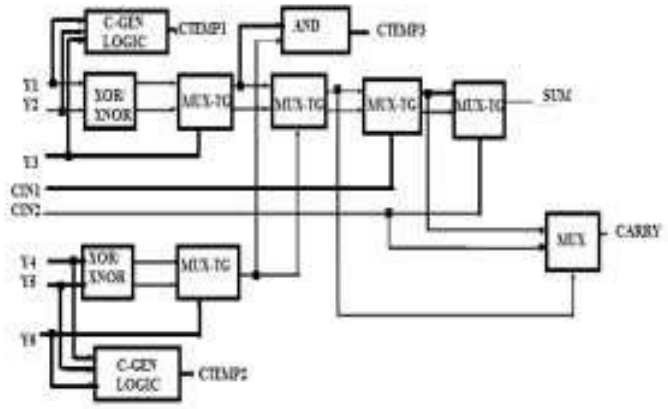
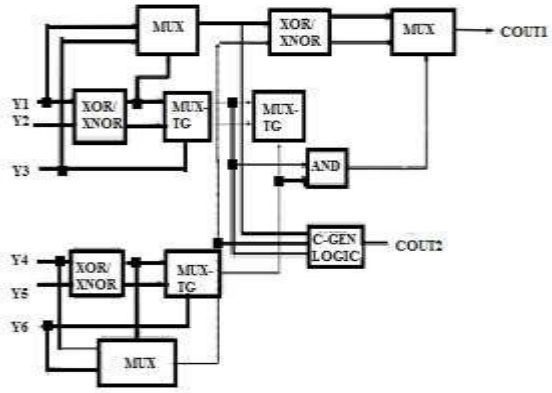


Fig.1 Circuit diagram of 6:2 compressor

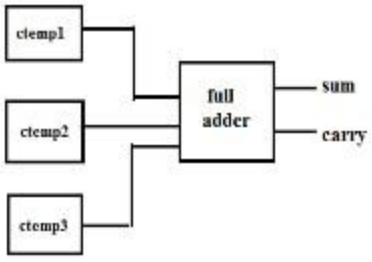
The equations for sum and carry are:

1. $SUM = Y1 \oplus Y2 \oplus Y3 \oplus Y4 \oplus Y5 \oplus Y6 \oplus CIN1 \oplus CIN2.$
2. $CARRY = (Y1 \oplus Y2 \oplus Y3) \oplus (Y4 \oplus Y5 \oplus Y6 \oplus CIN1) \bullet CIN2 + (Y1 \oplus Y2 \oplus Y3) \oplus (Y4 \oplus Y5 \oplus Y6 \oplus CIN1) \bullet (Y1 \oplus Y2 \oplus Y3 \oplus Y4 \oplus Y5 \oplus Y6)$

There are two methods two design the internal carries CIN1 and CIN2:



(a) Method 1



(b) Method 2

Fig. 2 Block Diagram for Internal Carry Generation Logic

Though in the first method, it may seem that the fan out is less, it still suffers from low speed problem as the carries depend upon the outputs of the multiplexers used in it. In the second method additional circuit is required consisting of three full adders, the speed of this method is higher than the first method. Hence, in this project, I have used the second method for designing. Thus the equations for COUT1 and COUT2 in terms of temporary carries are:

1) COUT1=CTEMP1 XOR CTEMP2 XOR CTEMP3.
2) COUT2= (CTEMP1.CTEMP2) + (CTEMP2.CTEMP3)+(CTEMP1.CTEMP3)
3) Where, $CTEMP1 = (Y1.Y2) + (Y2.Y3) + (Y1.Y3)$,
4) $CTEMP2 = (Y4.Y5) + (Y5.Y6) + (Y6.Y4) and$
5) $CTEMP3 = (Y1 \oplus Y2 \oplus Y3) \bullet (Y4 \oplus Y5 \oplus Y6)$

B. 7:2 Compressor

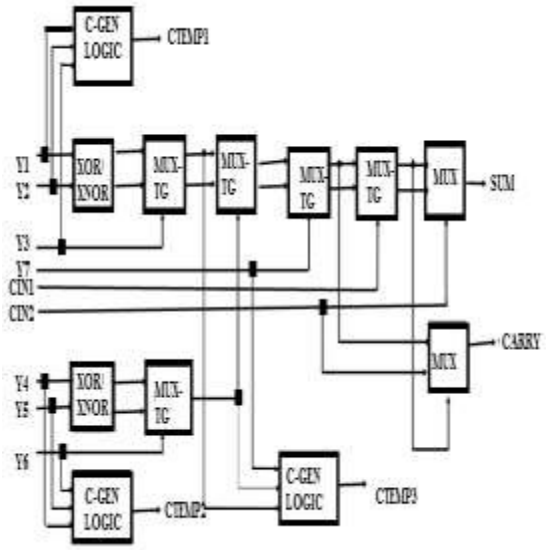


Fig.3 Block Diagram of 7:2 Compressor

$$Sum = Y1 \oplus Y2 \oplus Y3 \oplus Y4 \oplus Y5 \oplus Y6 \oplus Y7 \oplus CIN1 \oplus CIN2.$$

$$CARRY = (Y1 \oplus Y2 \oplus Y3) \oplus (Y4 \oplus Y5 \oplus Y6 \oplus Y7 \oplus CIN1) \bullet CIN2 + \overline{((Y1 \oplus Y2 \oplus Y3) \oplus (Y4 \oplus Y5 \oplus Y6 \oplus Y7 \oplus CIN1))} \bullet (Y1 \oplus Y2 \oplus Y3 \oplus Y4 \oplus Y5 \oplus Y6 \oplus Y7)$$

Again like the 6:2 compressor, there are two methods to design the carry outputs. But here too, we will use the second method using full adder operation. The carry equations in terms of temporary carries are:

COUT1=CTEMP1 XOR CTEMP2 XOR CTEMP3.

$$COUT2 = (CTEMP1.CTEMP2) + (CTEMP2.CTEMP3) + (CTEMP1.CTEMP3)$$

Where,

$$CTEMP1 = (Y1.Y2) + (Y2.Y3) + (Y1.Y3),$$

$$CTEMP2 = (Y4.Y5) + (Y5.Y6) + (Y6.Y4) \text{ and}$$

$$CTEMP3 = ((Y1 \oplus Y2 \oplus Y3) \bullet (Y4 \oplus Y5 \oplus Y6)) + ((Y1 \oplus Y2 \oplus Y3) \bullet Y7) + (Y7.(Y4 \oplus Y5 \oplus Y6)).$$

C. Input Partitioning Method

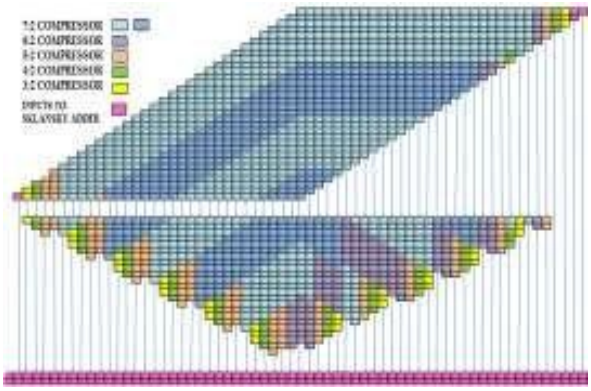


Fig. 4 Input Partitioning (Partial Products) Layout

In the above figure, we can see that the input bits are grouped in such a way that the dependency on the previous carry bits is decreased wherever possible. For example, in a7:2 compressor, where we have to add 9 bits: 7 input bits of the ith column, 2 previous carry bits from i-1th and i-2ndcolumn; we have directly added all the 9 bits from the same column for the upper row bits while the lower row bits take the internal carries from the i-1 and i-2 columns to complete the total compressor bits. This minimizes the circuit dependency for the upper row bits.

IV. RESULTS

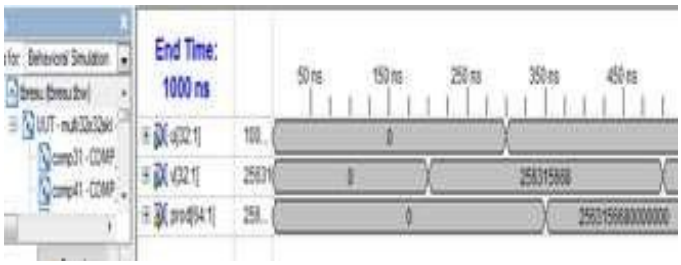


Fig.5 Test Bench Waveform of the Proposed Architecture

Table1. Comparison Between Different Architectures

Name of the multiplier	32x32 Proposed Wallace tree multiplier		32x32 Conventional Wallace tree multiplier	
	without Sklansky adder	Sklansky adder	without Sklansky adder	Sklansky adder
Area(Total equivalent gate count for design)	15,087	15405	15876	16,248
Delay (in ns)	174.885	103.500	210.577	158.237

V. CONCLUSION

In the proposed multiplier architecture, we have made two changes in the conventional Wallace tree multiplier; first being the use of compressors instead of half and full adders and secondly the use of Sklansky adder instead of a carry propagating adder. Also in comparison with the other novel compressor based architectures, we have decreased the output delay by use of the same column input bits instead of the previous column carry bits. Moreover, as it is a 32x32 multiplier we have used higher order compressors which will give much higher speed instead of just using 3:2, 4:2 and 5:2 compressors. This all leads to minimizing of the propagation delay and area of the overall architecture.

REFERENCES

[1] K Padma Priya1, P Sai Arun Kumar, B Mounika, P, "Review On Optimization Techniques Of Wallace Tree Multiplier", International Journal Of Electrical And Electronics Engineering And Technology, 2319 -2518 Vol. 3, No. 2, April 2014.

[2] Dakupati, Ravi Sankar and Ashraf Ali, "Design of Wallace Tree Multiplier by Sklansky Adder.", International Journal of Engineering Research and Applications (IJERA) , Vol. 3, Issue 1, 248-9622, 1036-1040 Page , January - February 2013.

[3] Bhaaskaran,V.S.K. Brindha, B. Sakthikumaran, S., Vinoth, C., "A Novel Low Power And High Speed Wallace Tree Multiplier For RISC Processor", 3rd International Conference on Electronics Computer Technology (ICECT), Volume:1, 2011.

[4] Tajul Hamimi Harun (2007), "High Speed 8 x 8 Wallace Tree Multiplier", A Report submitted in 2007 (from internet).

[5] Naveen Kr. Gahlan, Prabhat Shukla, Jasbir Kaur, "Implementation Of Wallace Tree Multiplier Using Compressor", International Journal of Computer Technology & Applications (IJCTA), Volume 3 (3), 1194-1199, May-June 2012, ISSN: 2229-6093.

[6] Shima Mehrabi, Keivan Navi, Omid Hasheimpour, "Performance Comparison Of High Speed High-Order (N: 2) and (N:3) CNFET-Based Compressors", International Journal of modelling and optimization, Volume 3 (3), no.5, October 2013.

[7] Marimathu, R., Dhruv Bansal, S. S. Balamurugan, P.S.Mallick, "Design of 8:4 And 9:4 Compressors For High Speed Multiplication", American Journal Of Applied Sciences 10(8), 893-900, 2013, ISSN: 1546-9239.

[8] R.UMA, Vidya Vijayan, M. Mohanapriya, Sharon Paul, "Area, Delay and Power Comparison of Adder Topologies", International Journal of VLSI design & Communication Systems (VLSICS) Vol.3, No.1, 3113 153, February 2012.

[9] JUN CHEN, "Parallel-Prefix Structures For Binary and Modulo Adders", Bachelor of Science in Information and Control Engineering Shanghai Jiao Tong University Shanghai, China 2000.