Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 3, March 2014, pg.510 – 518

RESEARCH ARTICLE

A High Speed and Area Efficient Wallace Tree Multiplier with Booth Recoded Technique

B. Venkata Sateesh[1], Shiju C Chacko[2]

[1]PG Scholar, Department of electronics and Communication Engineering
Hindustan University, Chennai, Tamil Nadu, India
[2]Assistant Professor, Department of electronics and Communication Engineering
Hindustan University, Chennai, Tamil Nadu, India

[1] satish.bethamcherla@gmail.com; [2] shijucc@hindustanuniv.ac.in

Abstract— Wallace tree is an improved version of tree based multiplier architecture.wallace tree multiplier implemented by using booth recoder in this paper. This paper aims reduction of additional latency and area of improved version of Wallace tree multiplier. In proposed method implementing by the use of booth algorithm to generate partial products and compressor adder techniques can be used to sum partial products. The modified architecture shows result of proposed architecture is around 67 Percent faster than the existing Wallace-tree multiplier, 22 percent faster than the radix-8 Booth multiplier, 18 Percent faster than the radix-16 Booth multiplier. Proposed architecture shows better performance in terms of area and speed.

Keywords— Wallace tree; Booth encoder; Compressor; Arithmetic; radix-8

I. INTRODUCTION

Multiplier plays an important role in today's digital signal processing and various other applications. With advances in technology, many researchers have tried and are trying to design which offers either of targets-high speed, low power consumption. Regularity of the layout and hence less area or even combination of them in one of the multiplier thus making them suitable for various high speed and low power compactable implementation. The key of the designing multiplier is the generation of partial products and the summation of the partial product Therefore, two methods can be used to improve the

speed of the multiplier.1.to reduce the number of partial products.2.to reduce the delay of the summation of partial products. People proposed lot of structure and algorithms to improve the performance of the multiplier.

However the fact remains that the speed and power are two conflicting performance constrains [4]. Multiplication is a basic arithmetic operation important in applications which rely on efficient implementation of generic arithmetic logic units(ALU) and Floating point units to execute dedicated operations like convolution and filtering. In the implementation of multiplier, the main s phases are generating of partial products, reduction of partial products using CSA (carry save architecture) and a carry propagation adder for the computation of the final result. It is obvious that the second phase that is the reduction of the partial products contributes most of the overall delay, area, power [2].

To generate partial products different algorithms such as booth algorithm, modified booth algorithm, booth recoder algorithm are used. these algorithms have proved to be useful in reducing the number of partial products generated. In the next step, speeding up the adding is highly critical for reducing the partial product. This step usually contributes the most to the delay, power and area of multiplier. For this purpose, use of the higher order compressor instead of conventional compressors have been explored [3].By this we are mainly concentration on reduction of partial products and there by achieving high speed over other multiplier architectures. The addition operation can be speed by using compressor. The compressor can operate without carry signal along its digital stages and hence it is much faster than conventional adder. In this paper we are going to generate partial products generation and reduction is accomplished using various architectures [2].

In the proposed architecture, the partial products generation is done by the use of booth recoding algorithm the partial products summation and reduction by the use of higher order compressor structures such as 4:2 and 5:2 compressor structures and .normal conventional compressor structure 3:2.

II. WALLACE TREE MULTIPLIER

C.S Wallace designed a possible structure, which perform the additional operations in parallel, resulting in lesser delay. It does a different way of parallel addition of partial product bits using a tree of carry save adder, which is known as Wallace. Wallace tree multiplier increase the speed by parallel addition operation in Wallace tree. All the partial products in each column are added together by a set of counter in parallel without any carry propagation. Another set of counter converts into two row matrix. Operation of Wallace multiplier follows three steps.

- Multiply each bit of one of the arguments, by each bit of the other, Depending on Positions of the multiplied bits, the wires carry different weights.
- Reduce the number of partial products to two by layers of full and half adders.
- Group the wires in two numbers, and add them with a conventional adder
- To improve the speed and reduce the latency of normal Wallace tree multiplier by approaching two methods, Booth recoder algorithm for generation and reduction of partial products and compressor technique uses to reduce the

summation of partial products by including 4:2compressor and 5:2with conventional structure full adder, half adder .it changes conventional structure as shown below.

Proposed architecture

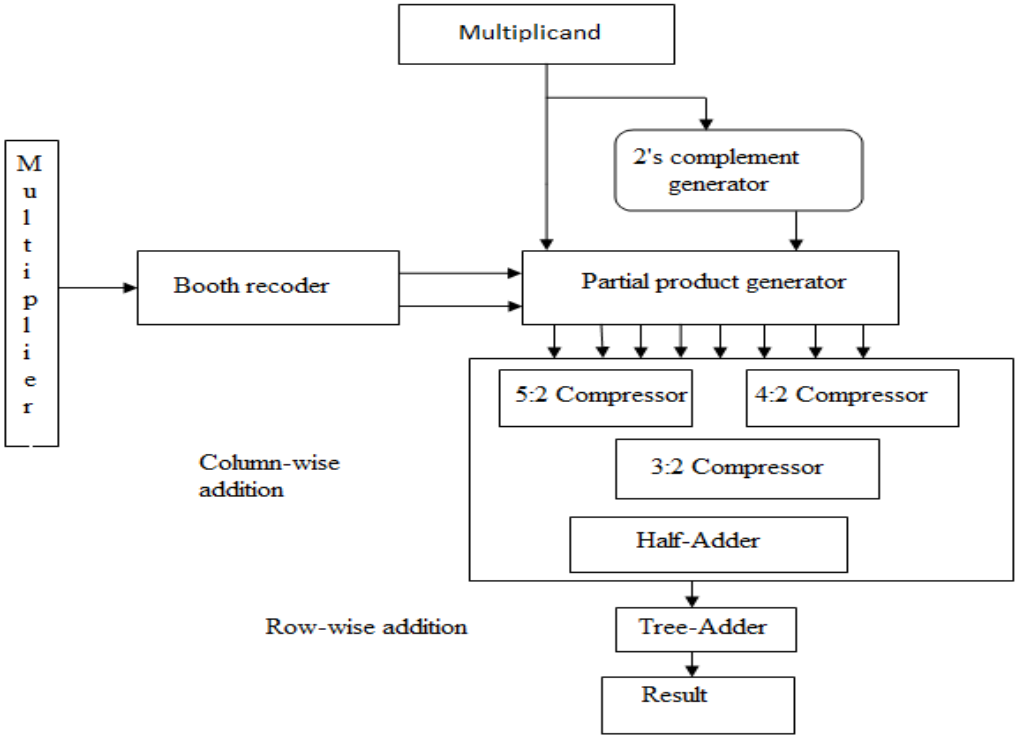


Figure 1.Proposed architecture

III. GENERATION OF PARTIAL PRODUCTS

Booth's recoder algorithm groups the original multiplier into groups of three consecutive digits where the outermost digit in each group is shared with the outermost digit of the adjacent group. Each of these group of three binary digits then corresponds to one of the numbers from the set {2, 1 0,-1,-1}. Each recoder produces a 3-bit output where the first bit represents the number 1 and the second bit represent number 2. The third and final bit indicates whether the number in the first or second bit is negative. The multiplicand is recoded according to the following equation:

$$Z= -2X_{i+1}+ X_i + X_{i-1}$$

Table:1 Booth recoder

X_{i+1}	X	X_{i-1}	Zi/2
0	0	0	0 * multiplicand
0	0	1	1* multiplicand
0	1	0	1* multiplicand
0	1	1	2 * multiplicand
1	0	0	-2 * multiplicand
1	0	1	-1 * multiplicand
1	1	0	-1 * multiplicand
1	1	1	0 * multiplicand

To generate and reduce the number of partial products of multiplier, modified booth algorithm has been used, in the modified booth algorithm, multiplier has been divided in groups of 4 bits and each groups of 4 bits are operational according to modified booth algorithm for generation of partial products 0,+1A,-1A,+2A,-2 A,+3A,-3A,+4A,-4A.[2] These partial products are summed using compressor in structure of Wallace tree.[4].

Table 2. Radix-8 Booth encoder

Multiplier bits	Operation for group
0000	0
0001	+1A
0010	+2A
0011	+3A
0100	+4A
0101	+5A
0110	+6A
0111	+7A
1000	-7A
1001	-6A
1010	-5A
1011	-4A
1100	-3A
1101	-2A
1110	-1A
1111	0

In radix-8 booth algorithm, multiplier operand B is partitioned into 11groups having each group of 4 bits. In first group first bit is taken zero and other bits are least significant three bit of operant multiplier. In second group, first bit is most significant bit of first group and other bits are next three bits of multiplier operand. In the third group, first bit is most significant

bit of second and other bits are next three bits of multiplier operand. This process is carried on. For each group, partial product is generated using multiplicand operand A.

IV. COMPRESSOR FOR PARTIAL PRODUCTS REDUCTION

Compressor is the building block for the partial product reduction. A compressor is a combinational device that compresses N inputs to 2 output lines i.e. sum and carry. The addition operation can be made fasts by using compressors. The compressor carry without carry signal then it speed up the operation. To speed up and latency can be reduced by the number adder in the partial products reduction stage. The critical path delay is minimized through at each stage is utilised properly by replacing with XOR blocks with multiplier blocks. Various adder structures are shown below; these conventional structures are replaced by compressor

3:2 COMPRESSOR ARCHITECTURE

The 3:2 compressor operations are same as the normal full adder. It process 3 input lines x1, x2, x3 and generate 2 outputs sum and carry. To reduce the critical path delay by considered as the X3=Cin ,the final sum and carry show below .the structures as shown 2(a) and2 (b).the critical overall delay is 2*Δ − XOR(Where Δ refers the delay).

$$\text{Sum}= x_1 \oplus x_2 \oplus x_3$$

$$\text{Carry}=(x_1 \oplus x_2) \oplus x_3+(x_1 \oplus x_2) \oplus x_1$$

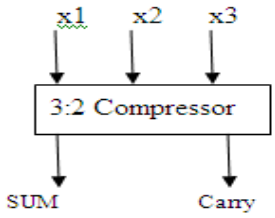


Figure .2(a) 3:2compressor

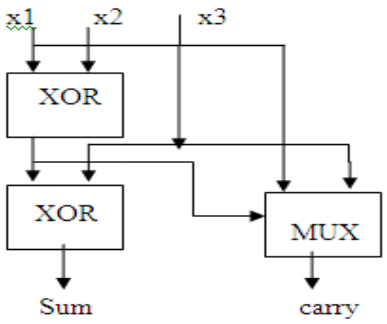


Figure.2(b) architecture

4:2 Compressor Architecture

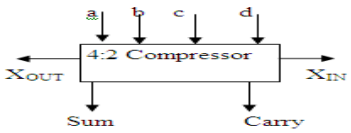
The 4:2 compressor takes four inputs x1, x2, x3, andx4 and two outputs sum and carry along with the carry_in andcarry_ out.the input C_{in} is the output from the previous lower significant compressor. The C_{OUT} is the output of the next compressor in the next significant stage. The critical path delay of the proposed implementation is Δ−XOR+2* Δ MUX. The structures are shown below, Structures of 4:2.

$$Sum=(x_1 \oplus x_2)\bullet \overline{(x_3 \oplus x_4)}+\overline{(x_1 \oplus x_2)}\bullet(x_3 \oplus x_4)\bullet\overline{C_{in}}+$$

$$\overline{(x_1 \oplus x_2)\bullet\overline{(x_3 \oplus x_4)}+\overline{(x_1 \oplus x_2)}\bullet(x_3 \oplus x_4)\bullet C_{in}}$$

$$C_{OUT=}(x_1 \oplus x_2)\bullet x_3+\overline{(x_1 \oplus x_2)}\bullet x_1$$

$$Carry=(x1 \oplus x2 \oplus x3 \oplus x4)\bullet C_{in}+\overline{(x1 \oplus x2 \oplus x3 \oplus x4)\bullet x4}$$



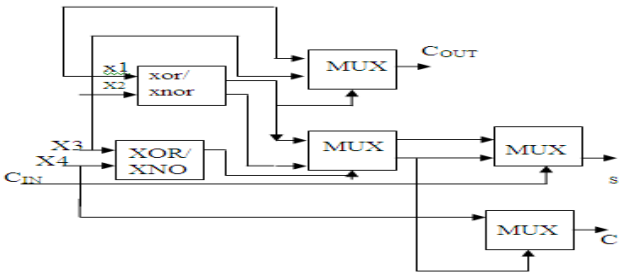


Figure: 3(a) 4:2 compressor 3(b) architecture

5:2 compressor architecture

The 5-2 Compressor block has 5 inputs X1,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.4a. 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. The conventional implementation of the compressor block is shown in Fig where 3 cascaded full adder cells are used. When these full adders are replaced with their constituent blocks of XOR gates then it can be observed that the overall-XOR for the sum delay or carry output is. Many equal architectures to have been6*Δ proposed where the delay has been reduced to 4* -XORΔ.

The basic equation of that governs the function of the 5:2 compressors as shown In figure below:

$$x_1+x_2+x_3+x_4+x_5+c_{in1}+c_{in2}=\text{sum}+2*(\text{carry}+C_{OUT1}+C_{OUT2})$$

the sum and carry of the MUX implemented structures

$$Sum = X_1 \oplus X_2 \oplus X_3 \oplus X_4 \oplus X5 \oplus C_{in1} \oplus c_{in2}$$

$$C_{out2}=(X_4 \oplus X_5)\bullet C_{in1}+(\overline{X_4 \oplus X_5})\bullet X_4$$

$$Carry=((X_1 \oplus X_2 \oplus X_3 \oplus) \oplus (X_4 \oplus X_5 \oplus C_{in1}))\bullet C_{in2}+\overline{((x_1 \oplus x_2 \oplus x_{3)} \oplus (x_4 \oplus x_5 \oplus c_{in1}))\bullet(x_1 \oplus x_2 \oplus x_3)}$$

$$C_{out1} = (x_1 \oplus x_2) \bullet x_3 + x_1 \bullet x_2$$

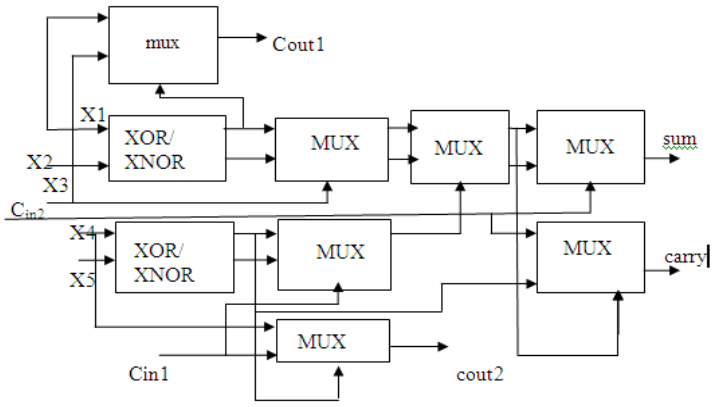


Figure.4 5:2 architecture

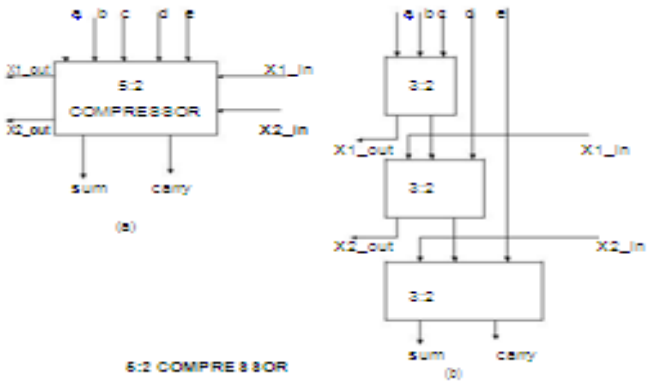


Fig.5.(a)compressor block 5(b).conventional implementation

In all the general implementations of the XOR or MUX block, in particular CMOS implementation, the output and its complement are generated. But in the existing architectures this advantage is not being utilized at all. 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 aims to reduce the overall latency of Wallace tree multiplier, this leads to increased speed and reduce the power consumption. The design makes use of booth recoder algorithm for partial product generation and compressors in place full adder, and final carry propagation stage is replaced by the carry selective adder. The below figure depicts the first stage consists of a 3:2 compressor. Second stage two full adders have been grouped and implemented using 4:2.third stage consists of 5:2 which is combination of three full adders.

V. SIMULATION RESULT AND OBESERVATIONS

Figure.7 simulation output of 64x64 bit booth recoded Wallace tree multiplier

VI. CONCLUSION

The proposed 64x64 bit Wallace tree multiplier has been implemented on FPGA and the delay comparison. With existing Wallace tree multiplier, booth multiplier radix-8, and radix-16, 32x32 bit Wallace multiplier as shown below.

TYPEOF MULTIPLIER	WIDTH	DELAY(ns)
Wallace tree multiplier	8-bit	7.168
Modified Booth multiplier(radix-8)	32-bit	12.452
Modified booth multiplier (radix-16)	32-bit	11.653
32x32booth recoder Wallace	32-bit	9.293
Proposed multiplier	32-bit	8.393

REFERENCES

[1] Vinoth, C.; Bhaaskaran, V.S.K.; Brindha, B.; Sakthikumaran, S.; Kavinilavu, V.; Bhaskar, B.; Kanagasabapathy, M.; and Sharath, B.; "A novel low power and high speed Wallace tree multiplier for RISC processor," *3rd International Conference on Electronics Computer Technology (ICECT), 2011*, vol.1, pp.330-334, 8-10 April 2011.

[2] Sreehari Veeramachaneni; Kirthi M Krishna; Lingamneni Avinash; Sreekanth Reddy Puppala and 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, 2007*, pp.324-329, Jan. 2007.

[3] Prasad, K. and Parhi, K.K.; "Low-power 4-2 and 5-2 compressors," *Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers, 2001*, vol.1, no., pp.129-133 vol.1, 4-7 Nov. 2001

[4] Jagadeshwar rao.m, sanjay Dubey; "A High speed and area efficient booth recoded Wallace tree multiplier for fast arithmetic circuits "Asia pacific conference on postgraduate research .pp220-234, 5th-7th December 2012.

[5] Weinan Ma and Shuguo Li; "A new high compression compressor for large multiplier," *Solid-State and Integrated-Circuit Technology, 2008. ICSICT 2008. 9th International Conference on* , vol., no., pp.1877-1880, 20-23 Oct. 2008.

[6] Shen-Fu Hsiao; Ming-Roun Jiang; Jia-Sien Yeh; , "Design of high-speed low-power 3-2 counter and 4-2 compressor for fast multipliers," *Electronics Letters* , vol.34, no.4, pp.341-343, 19 Feb 1998.

[7] Zhongde Wang; Jullien, G.A. and Miller, W.C.; "A new design technique for column compression multipliers," *IEEE Transactions on Computers,*, vol.44, no.8, pp.962-970, Aug 1995.

[8] Chen Ping-hua and Zhao Juan; "High-speed Parallel 32×32-b Multiplier Using a Radix-16 Booth Encoder," *Third International Symposium on Intelligent Information Technology Application Workshops, 2009. IITAW '09*, pp.406-409, 21-22 Nov. 2009.

