ISSN: 0974-2115
www.jchps.com Journal of Chemical and Pharmaceutical Sciences

# High performance and area efficient signed wallace tree multiplier using compressors

**Abhilash R*, Sanjay Dubey, Shaik Shafi, Chinnaiah MC**
Department of ECE, BVRIT, Narsapur, Medak (Dist), Telangana, India,
***Corresponding author: E-Mail: rathnaabhilash@gmail.com**

**ABSTRACT**

Signed Wallace tree multiplier is famous for 2s complement data representation. In a digital system, Compressor techniques are applied to Signed Wallace tree Multipliers. In this paper, we propose a new approach of 4:2 compressor and 5:2 compressor architectures. The validation of proposed architectures was done on Signed Baugh-Wooley Multiplier with Wallace tree. The result shows 4.67 percentage reduction in number of slices and 2.16 percentage faster than existing compressor architecture when used in above Multiplier. The coding is done in verilog HDL the Architectures are simulated and synthesized for Xilinx Virtex 6 low power FPGA device.

**KEY WORDS:** Signed, unsigned, Baugh-Wooley, Wallace tree, Compressor, Multiplier.

## 1. INTRODUCTION

Compressors are used for summing the partial products and play a crucial role in high speed digital circuits such as multipliers. With the recent improvement in the fast multiplication operation, the system using compressors, improves the performance in the digital system with the less critical path delay. In this paper 4-2 and 5-2 compressors with delay-area optimized are proposed. The performance of these compressors is tested by using them in 8 bit signed Wallace tree multiplier and compared with the existing 3-2, 4-2, 5-2 compressors available in literature.

**Background of compressor:**

**Compressor3-2:** The Compressor 3-2 architecture, and the full adder circuit will have the same functionality. The compressor architecture shown in figure.1 is governed by the

$$sum = x_1\ xor\ x_2\ xor\ x_3 \qquad (1)$$

$$carry = (x_1 xor\ x_2) \cdot x_3 + \overline{(x_1 xor\ x_2)} \cdot x_1 \qquad (2)$$

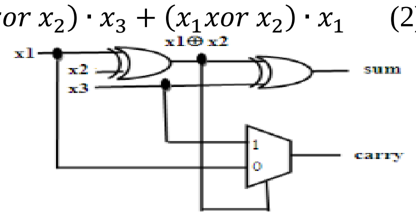


**Fig.1.Compressor 3-2**

**Compressor 4-2:** Previously the work done on 4:2 compressor architecture is shown in Fig.2. The Compressor 4-2 architecture, and the 4-2 conventional full adder circuit will have the same functionality. It has four inputs x1, x2, x3, x4 and one carry input cin in which the xor and xnor gate are used. The cout, carry, and sum are the outputs.

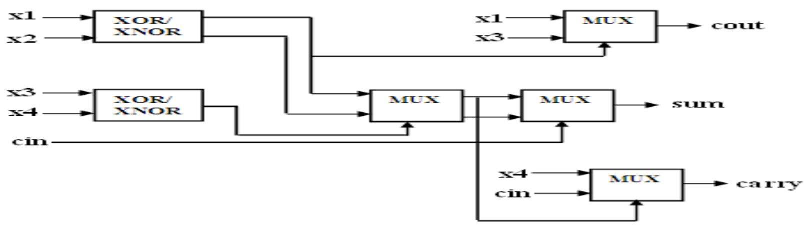


**Fig.2.Compressor 4-2**

$$um = (x_1\ xor\ x_2) \cdot \overline{(x_3\ xor\ x_4)} + \overline{(x_1 xor\ x_2)} \quad (3)$$

$$\cdot (x_3\ xor\ x_4) \cdot \overline{cin} + \overline{(x_1 xor\ x_2) \cdot \overline{(x_3 xor\ x_4)}} \quad (4)$$

$$+\overline{(x_1\ xor\ x_2)} \cdot (x_3\ xor\ x_4) \cdot cin \ (5)$$

$$carry = (x_1\ xor\ x_2\ xor\ x_3\ xor\ x_4) \cdot cin \ (6)$$

$$+\overline{(x_1\ xor\ x_2\ xor\ x_3\ xor\ x_4\ )} \cdot x_4 \qquad (7)$$

**Compressor5-2:** The compressor 5-2 architecture as five inputs x1, x2, x3, x4, x5 and two carry-in, in which the xor and xnor gates are used. The cout1, cout2, carry and sum are the outputs.

$$sum = x_1\ xor\ x_2\ xor\ x_3\ xor\ x_4\ xor\ x_5\ xor\ \ cin_1\ xor\ \ cin_2 \quad (8)$$

$$cout1 = (x_1 + x_2) \cdot x_3 + (x_1 \cdot x_2) \quad (9)$$

$$Cout2\ = (x_4\ xor\ x_5\ ) \cdot cin_1 + (\overline{x_4\ xor\ x_5}) \cdot x_4\ (10)$$

$$carry = (x_1\ xor\ x_2\ xor\ x_3)\ xor \quad (11)$$
$$(x_4\ xor\ x_5\ xor\ cin_1) \cdot cin_2 \quad (12)$$
$$+\overline{(x_1\ xor\ x_2\ xor\ x_3)}\ xor\ (x_4\ xor\ x_5\ xor\ cin_1) \quad (13)$$
$$\cdot\, (x_1\ xor\ x_2\ xor\ x_3) \quad (14)$$

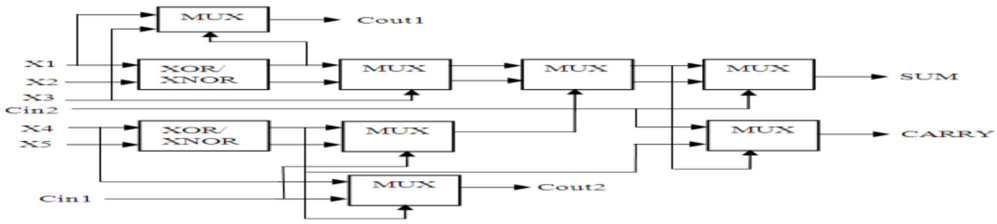


**Fig.3.Compressor 5-2**

**Background on multipliers:**

**Signed Baugh-Wooley Multiplier:** Modifying the unsigned multiplication by using twos compliment operands technique is proposed by Baugh and Wooley. The Baugh-Wooley algorithm is a simple way of performing signed multiplication which is shown in fig.4 and fig.5 for the multiplication operation of 8 bit. The partial-product array has been reorganized according to the scheme of Hatamian, and the creation of reorganized partial-product array comprises three steps:

i) The most significant partial product of the first N-1 rows and the last row of partial products except the most significant have to be negated,

ii) A constant one is added to the Nth column,

iii) The most significant bit (MSB) of the final result is negated.

**Wallace tree Multiplier:** The Wallace tree multiplier aims to decrease the overall delay which means the critical path and also the number of stages are reduced. For the different stages in the Wallace tree multiplier the compressor techniques are applied, such as Compressor 3-2, 4-2, 5-2 for achieving the reduction of slices as shown in the fig.6.

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | t7 | t6 | t5 | t4 | t3 | t2 | t1 | t0 |
| | | | | | | | | u7 | u6 | u5 | u4 | u3 | u2 | u1 | u0 |
| | | | | | | | 1 | $\overline{s70}$ | s60 | s50 | s40 | s30 | s20 | s10 | s00 |
| | | | | | | | $\overline{s71}$ | s61 | s51 | s41 | s31 | s21 | s11 | s01 | |
| | | | | | | $\overline{s72}$ | s62 | s52 | s42 | s32 | s22 | s12 | s02 | | |
| | | | | | $\overline{s73}$ | s63 | s53 | s43 | s33 | s23 | s13 | s03 | | | |
| | | | | $\overline{s74}$ | s64 | s54 | s44 | s34 | s24 | s14 | s04 | | | | |
| | | | $\overline{s75}$ | s65 | s55 | s45 | s35 | s25 | s15 | s05 | | | | | |
| | | $\overline{s76}$ | s66 | s56 | s46 | s36 | s26 | s16 | s06 | | | | | | |
| 1 | s77 | $\overline{s67}$ | $\overline{s57}$ | $\overline{s47}$ | $\overline{s37}$ | $\overline{s27}$ | $\overline{s17}$ | $\overline{s07}$ | | | | | | | |
| p15 | p14 | p13 | p12 | p11 | p10 | p9 | p8 | p7 | p6 | p5 | p4 | p3 | p2 | p1 | p0 |

**Fig.4.Signed Baugh-Wooley Multiplier**

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | | | | | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | | | | | | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | | | | | | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| | | | | | | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | |
| | | | | | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | | |
| | | | | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | | | |
| | | | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | | | | |
| | | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | | | | | |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

**Fig.5.Example of Signed Baugh-Wooley Multiplication**

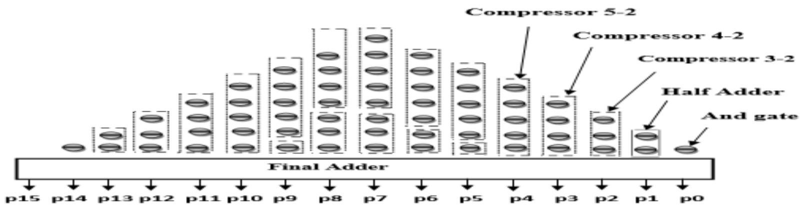


**Fig.6.Wallace tree Multiplier**

**Booth Multiplier:** Andrew Donald Booth developed the famous Booth algorithm in the year 1950. It is a fastest algorithm in which multiplication of the unsigned and signed can be done. In booth encoder the two bits of multiplier term are overlapped with one another as shown in figure.7. The booth encoder generates partial products and by using Wallace tree technique the number of levels will be reduced. So the critical path also reduces and this type of architecture achieves better performance.

M=00000010(2) R=11110100(-12) Let make R as 11110100z where z=0, so now R=111101000

**R= 111101000**

**Fig.7.Multiplier term with division of block from least significant bit**

ISSN: 0974-2115
www.jchps.com Journal of Chemical and Pharmaceutical Sciences

**Table.1.Booth algorithm of Radix2**

| Ck | S |
|---|---|
| 00 | 0 |
| 01 | +1(MD) |
| 10 | -1(-MD) |
| 11 | 0 |

**Table.2.Booth algorithm of Radix4**

| Ck | Sk |
|---|---|
| 001 | 1(MD) |
| 010 | 1(MD) |
| 011 | 2(lbit left shift) |
| 100 | -2(lbit left shift) |
| 101 | -1(-MD) |
| 110 | -1(-MD) |
| 111 | 0 |

$$M \text{ x } R = 2(00)\ \ 2^0\ \ =)\ 2(0)\ \ 1 = 0 \tag{15}$$

$$= 2(00)\ \ 2^1\ \ = 2(0)\ \ 2 = 0 \tag{16}$$

$$= 2(10)\ \ 2^2\ \ = 2(1)\ \ 4 = \ 8 \tag{17}$$

$$= 2(01)\ \ 2^3\ \ = 2(1)\ \ 8 = 16 \tag{18}$$

$$= 2(10)\ \ 2^4\ \ = 2(1)\ \ 16 = \ 32 \tag{19}$$

$$= 2(11)\ \ 2^5\ \ = 2(0)\ \ 32 = 0 \tag{20}$$

$$= 2(11)\ \ 2^6\ \ = 2(0)\ \ 64 = 0 \tag{21}$$

$$= 2(11)\ \ 2^7\ \ =)\ 2(0)\ \ 128 = 0 \tag{22}$$

$$\text{Result} = \ 24 \tag{23}$$

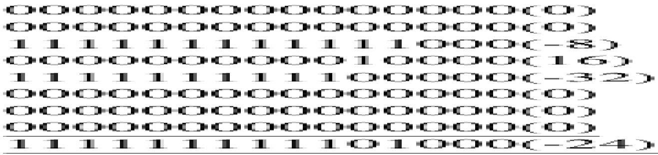


**Fig.8.Final product representing in binary form**

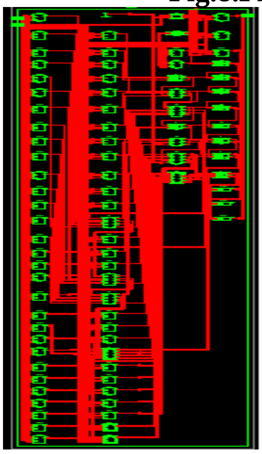

**Fig.9.RTL Schematic of Signed Booth Wallace tree multiplier Radix2**

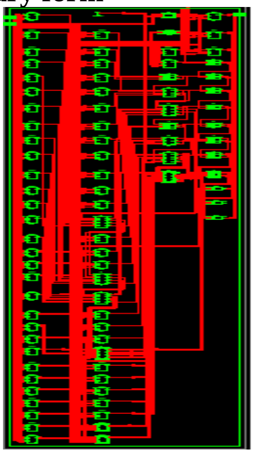

**Fig.10.RTL Schematic of Signed Booth Wallace tree Radix4**

## 2. PROPOSED COMPRESSORS

**Proposed Compressor 4-2 Architecture:** The proposed compressor 4-2 architecture is shown in fig.7. The xor gate is used for arithmetic operation in which summing of partial products is done, where sum inputs are x1, x2, x3, x4 and Cin. By using the basic logic gates carry-out module is designed. Multiplexer accepts all signals but selects the signals one by one, so the selection of the signals will be faster which reduces latency.

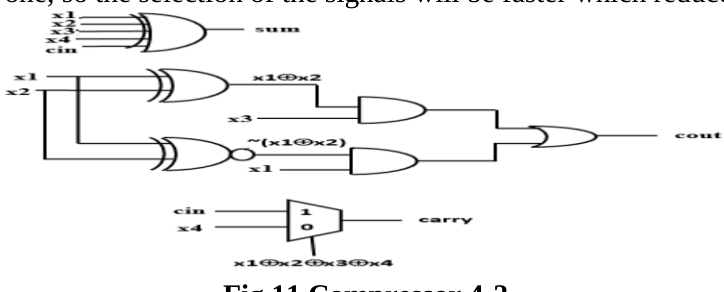


**Fig.11.Compressor 4-2**

$$sum = x_1\ xor\ x_2\ xor\ x_3\ xor\ x_4\ xor\ cin \quad (24)$$
$$Cout = (x_1\ xor\ x_2\ ) \cdot x_3 + (\overline{x_1\ xor\ x_2}) \cdot x_1 \quad (25)$$
$$carry = (x_1\ xor\ x_2\ xor\ x_3\ xor\ x_4) \cdot cin \quad (26)$$
$$+\overline{(x_1\ xor\ x_2\ xor\ x_3\ xor\ x_4\ )} \cdot x_4 \quad (27)$$

**Proposed Compressor 5-2 Architecture:** In this proposed 5-2 Compressor architecture, the multiplexer, inverter and xor gate combinations are used for the summing operation in which inputs are x1, x2, x3, x4, x5 and carry-inputs are cin1 and cin2. For the carry-out1 and carry out2 the multiplexer gates are used because the selection of the signals will be faster and there will be reduction of delay. The output carry is designed with the basic gates.

$$sum = x_1\ xor\ x_2\ xor\ x_3\ xor\ x_4\ xor\ x_5 \quad (28)$$
$$Cout1 = (x_1\ xor\ x_2\ ) \cdot x_3 + (\overline{x_1\ xor\ x_2}) \cdot x_1 \quad (29)$$
$$cout2 = (x_1\ xor\ x_2\ xor\ x_3\ xor\ x_4) \cdot cin_1 \quad (30)$$
$$+\overline{(x_1\ xor\ x_2\ xor\ x_3\ xor\ x_4\ )} \quad (31)$$
$$\cdot (x_1\ xor\ x_2\ xor\ x_3) \quad (32)$$
$$carry = [(x_1\ xor\ x_2\ xor\ x_3\ xor\ x_4\ xor\ cin_1) \cdot\ x_5] \quad (33)$$
$$+ [(x_1\ xor\ x_2\ xor\ x_3\ xor\ x_4\ xor\ cin_1) + x_5] \cdot cin_2 \quad (34)$$

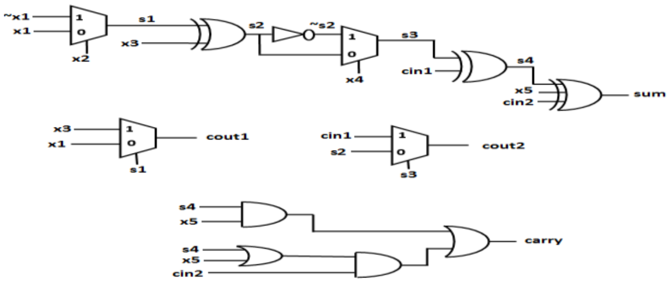


**Fig.12.Compressor 5-2**

**Signed wallace tree multiplier using proposed compressors:** The Signed Baugh-Wooley multiplier with Wallace tree is shown in fig.9. The performance of the proposed 4-2 and 5-2 compressors is tested on the above multiplier architecture.

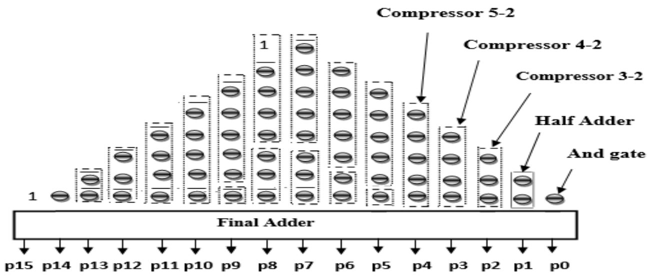


**Fig.13.Signed Wallace tree multiplier using Proposed Compressors**

## 3. Results

**Simulation Results:** The simulation results are shown from fig.10 to fig.14. The speed and area consumption of both architectures are synthesized using Xilinx tools. The result shows 4.67 percent-age reduction in number of slices and 2.16 percentage faster than existing compressor architecture when used on above Multiplier.

RTL schematic

Register Transfer Language (RTL) schematic of Signed Baugh-Wooley Wallace tree multiplier is shown in fig.19. The existing 3-2 and proposed 4-2, 5-2 Compressor architectures are used to do summing of the partial products. The Table-I shows that the performance increases and occupies less slices when compared to existing.

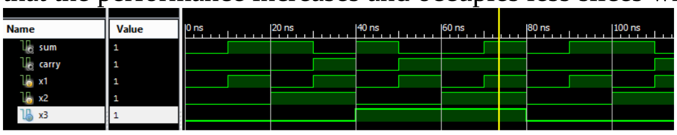

**Fig.14.Compressor 3-2**

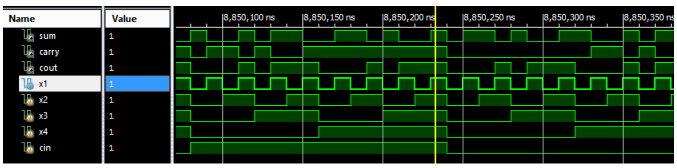

**Fig.15.Compressor 4-2**

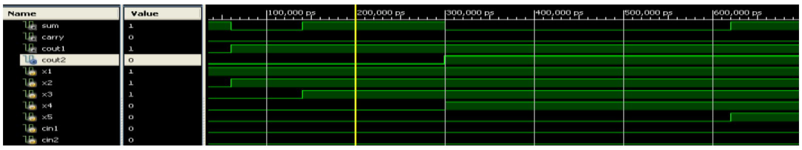

Fig.16.Compressor 5-2


Fig.17.Wallace tree multiplier

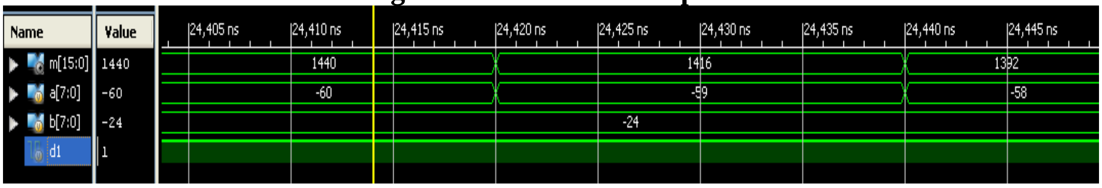


Fig.18.Signed Wallace tree multiplier

**Table.3.Signed Baugh-Wooley with Wallace tree multiplier**

| Type of multiplier(8bit) | Area (no of silices) | Delay (ns) |
|---|---|---|
| Using Proposed Compressors | 107/46560 | 8.343 |
| Using Existing Compressors | 112/46560 | 8.528 |
| Using Conventional (Fulladder) | 109/46560 | 10.143 |

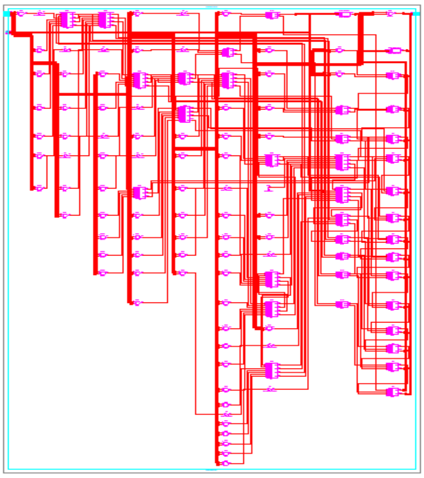

Fig.19.RTL schematic

**Table.4.Signed Booth with Wallace tree multiplier of Radix2**

| Type of multiplier(8bit) | Area (no of silices) | Delay (ns) |
|---|---|---|
| Using Proposed Compressors | 170/46560 | 10.899 |
| Using Existing Compressors | 171/46560 | 9.686 |
| Using Conventional (Fulladder) | 167/46560 | 11.050 |

**Table.5.Signed Booth with Wallace tree multiplier of Radix4**

| Type of multiplier (8bit) | Area (no of silices) | Delay (ns) |
|---|---|---|
| Using Proposed Compressors | 106/46560 | 7.189 |
| Using Existing Compressors | 107/46560 | 7.189 |
| Using Conventional (Fulladder) | 107/46560 | 7.189 |

## 4. CONCLUSION

By using the new approach in compressor architectures which are used in the Wallace Tree multiplier, we can obtain better results in area and delay. These multipliers when used for Robotics computational algorithms will play a crucial role. The coding is done in Verilog; the design is simulated and synthesized using Xilinx tools.

**REFERENCES**

Baugh CR, Wooley BA, A twos complement parallel array multiplication algorithm, IEEE Trans Comput., 22, 1973, 1045-1047.

Daniel R Kelly, Braden J Phillips, and Said Al-Sarawi, Approximate Multiplication and Division for Arithmetic Data Value Speculation in a RISC Processor, D.R. Kelly Chip Tec, Centre for High Performance Integrated Technologies and Systems, The University of Adelaide, Adelaide, Australia. Algorithm-Architecture Matching for Signal and Image Processing, Springer Science Business Media B.V, 2011.

Rao M.J, Dubey S, A high speed and area efficient Booth recoded Wallace tree multiplier for fast arithmetic circuits, Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (Prime Asia), 2012, 220-223.

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

Umatri Pradhananga, Xingguo Xiong, and Linfeng Zhang, PSPICE Implementation of Block-Wise Shut-Down Technique for 8x8 Bit Low Power Pipelined Booth Multiplier, New Trends in Networking, Com-puting, E-learning, Systems Sciences, and Engineering, Lecture Notes in Electrical Engineering 312, DOI 10.1007/978-3-319-06764-340, Springer International Publishing Switzerland, 2015.

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), 1, 2011, 330-334.

Xing guo Xiong, Muzi Lin, Low Power 8-Bit Baugh-Wooley Multiplier Based on Wallace Tree, Architecture Department of Electrical and Computer Engineering, University of Bridgeport, Lecture Notes in Electrical Engineering 151, DOI: 10.1007/978-1-4614-3558-773, Springer Science+Business Media New York, 2013.