Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 9, August 2021: 1037 - 1048

Modified Unsigned Multiplier Algorithm with Wallace Tree Technique

1. Dr. Md.Ejaz Ahamed **2. Ziaul Haque**

Dept. of ECE,Medak College of Engg and Tech., Research Scholar, JJTU
Telangana, India,Email:aemny9@gmail.comRajasthan, India, Email: ziaul.haque197@gmail.com

Abstract—A Wallace tree technique using Modified Unsigned Multiplier is proposed in this paper and is designed at gate level using Verilog HDL. Synthesized for Xilinx Virtex 6 low power, Spartan 6, Spartan 3E FPGA device. The result shows an improvement in terms of speed and slices or area. Performance and Area of integrated circuits are a major concern for VLSI circuit designers. This paper aims at more reduction of the latency and area or the number of slices. Day to day advent of new technology in the fields of digital signal processing and communication, VLSI, there is a great demand for the high-speed processing and Efficient area reduction in the design. *Index Terms*—Wallace tree Technique, Unsigned Multiplier, VLSI.

I. INTRODUCTION

Multipliers are the key hardware blocks in digital signal processor and microprocessors and also multiplier's performance highly depends on the processor with technology advancements, high demand for performance multipliers, at the same time keeping in a consciousness moderate power dissipation and low area or number of slices. Over the past few years, many new structures of multipliers have been constructed and explored. The proposed structure is shown in below flow chart.

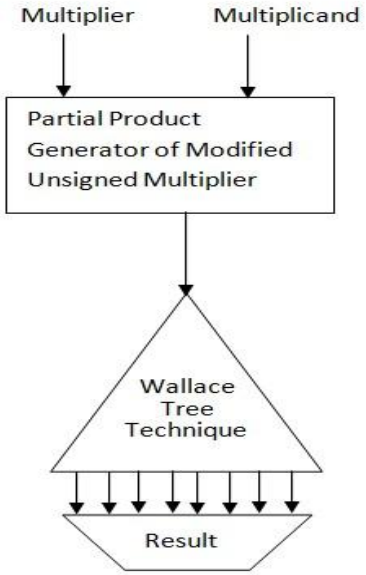

Fig. 1: Proposed Structure

II. UNSIGNED RIPPLE ARRAY MULTIPLIER

Unsigned Ripple array is a binary multiplier generally the multiplier and multiplicand operation is done by using the AND gate operation where the AND gates are replaced with NOR gate so that the performance is increased and the number of slices are reduce so that we get the optimized FPGA results.

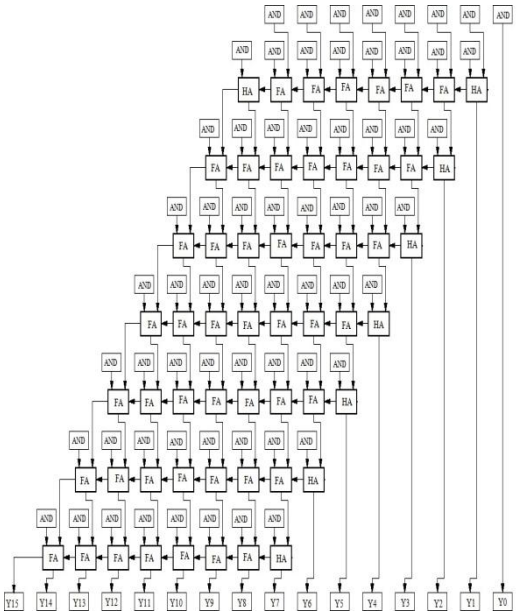

Fig. 2: Conventional Unsigned Ripple Multiplier of 8bit

III. WALLACE TREE TECHNIQUE

Wallace tree technique is used after the partial products are generated by the unsigned multiplier these partial products are arranged in tree structure for that structure we are adding the 3-2 Full Adder and 4-2 Full Adder and 5-2 Full Adder. which is show in figure 4.

IV. FULL ADDER FOR ADDING THE PARTIAL PRODUCTS

Full Adder consists of 2 inputs a,b and 1 carry-in and it consists of 2 outputs sum, carry. Conventional Full Adder with carry AND gate is shown in figure 5 and Conventional Full Adder with NOR gate (eg...a.b=(a'+b')') is used is shown in figure 6.

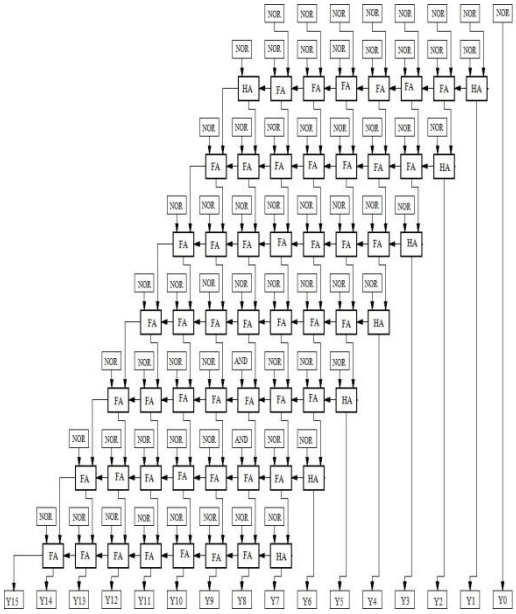

Fig. 3: Modified Unsigned Ripple Multiplier of 8bit proposed by srihari-at with complement input A.B=(A'+B')'

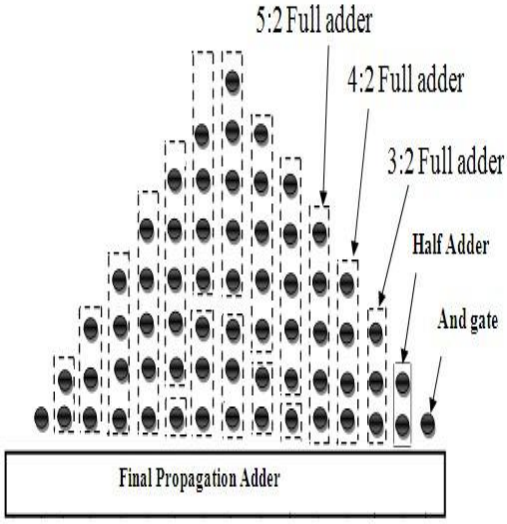

Fig. 4: Wallace Tree Technique applied for Modified Unsigned Ripple Array multiplier of 8bit

V. SIMULATION ANALYSIS

The simulation Analysis is illistrated in figure 10 in the waveform we observe the multiplication of two numbers of eight bit a,b and the result m of fifteen bits.

A. Result

In the result section the Comparison of Delay in table I and Comparison of Area in table II is shown.

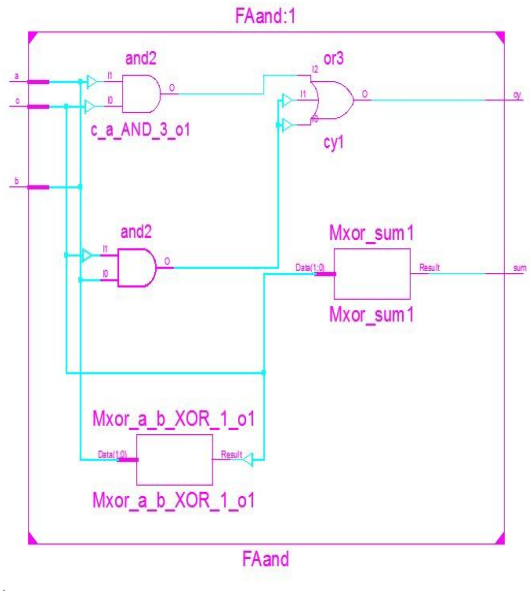

Fig. 5: Full Adder with AND gate carry(ADFA)

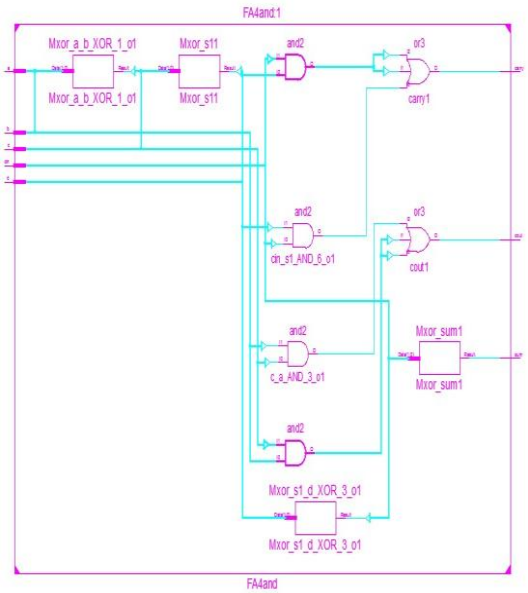

Fig. 6: Full Adder4 with AND gate carry(ADFA4)

TABLE I: Comparison of Delay (Virtex 6 Low power XC6VLX75TL, FF484 (-1))

Type of multiplier	Width	Delay(ns)
Conventional Unsigned Wallace tree with ADFA,	8bit	10.032

ADFA4, ADFA5		
Modified Wallace tree with ADFA, ADFA4, ADFA5	8bit	9.163
Conventional Unsigned Wallace tree with NRFA, NRFA4, NRFA5	8bit	9.008
Modified Wallace tree with NRFA, NRFA4, NRFA5	8bit	8.929

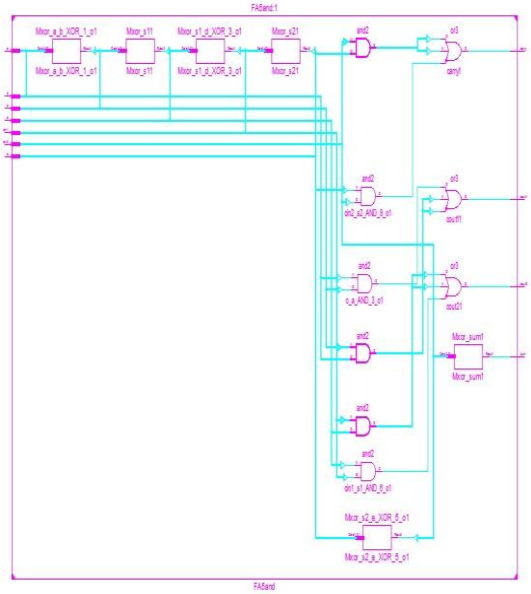

Fig. 7: Full Adder5 with AND gate carry(ADFA5)

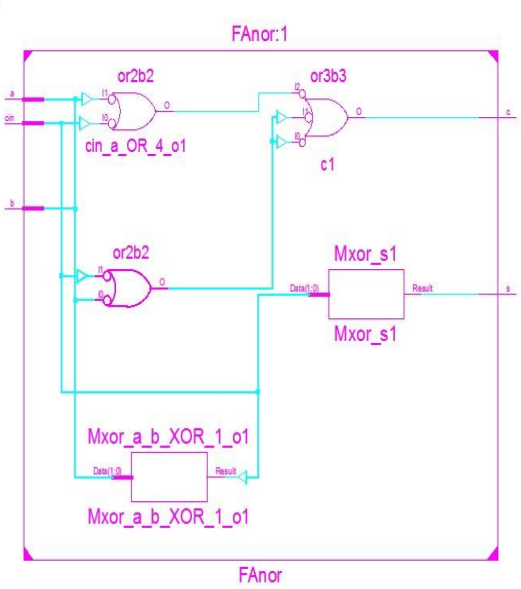

Fig. 8: Full Adder with NOR gate carry(NRFA)

TABLE II: Comparison of Delay (Spartan 6 XC6SLX150, FGG484, (-3)

Type of multiplier	Width	Delay(ns)
Conventional Unsigned Wallace tree with ADFA, ADFA4, ADFA5	8bit	19.726
Modified Wallace tree with ADFA, ADFA4, ADFA5	8bit	18.428
Conventional Unsigned Wallace tree with NRFA, NRFA4, NRFA5	8bit	18.280
Modified Wallace tree	8bit	18.237

with NRFA, NRFA4, NRFA5		

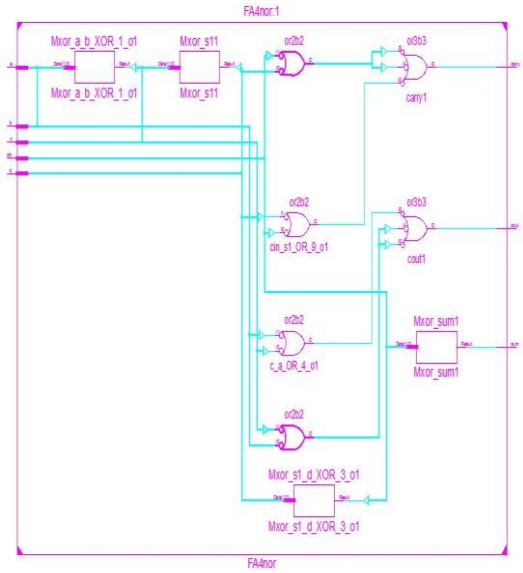

Fig. 9: Full Adder4 with NOR gate carry(NRFA4)

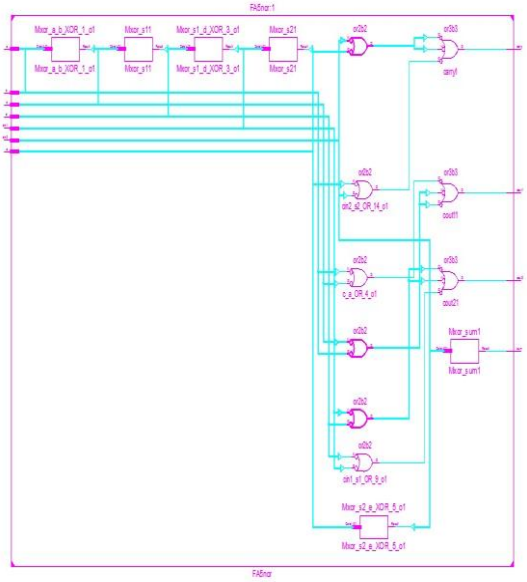

Fig. 10: Full Adder5 with NOR gate carry(NRFA5)

TABLE III: Comparison of Delay (Spartan 3E XC3S500E, FG320, (-5)

Type of multiplier	Width	Delay(ns)
Conventional Unsigned Wallace tree with ADFA, ADFA4, ADFA5	8bit	20.984
Modified Wallace tree with ADFA, ADFA4, ADFA5	8bit	21.973
Conventional Unsigned Wallace tree with NRFA, NRFA4, NRFA5	8bit	21.023
Modified Wallace tree with NRFA, NRFA4, NRFA5	8bit	21.023

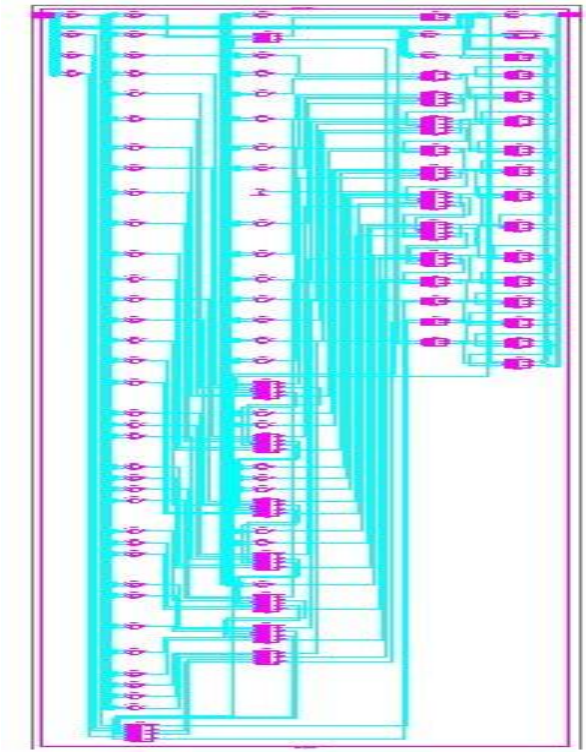

Fig. 11: RTL Modified Unsigned Ripple multiplier with Wallace Technique

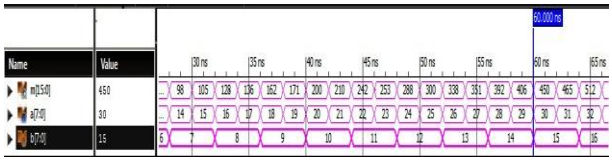

Fig. 12: simulation result

TABLE IV: Comparison of Area (Virtex 6 Low power XC6VLX75TL, FF484 (-1))

Type of multiplier	Width	Slice LUTs Used
Conventional Unsigned Wallace tree with ADFA, ADFA4, ADFA5	8bit	109/46560
Modified Wallace tree with ADFA, ADFA4,	8bit	101/46560

ADFA5		
Conventional Unsigned Wallace tree with NRFA, NRFA4, NRFA5	8bit	110/46560
Modified Wallace tree with NRFA, NRFA4, NRFA5	8bit	106/46560

TABLE V: Comparison of Area (Spartan 6 XC6SLX150, FGG484, (-3)

Type of multiplier	Width	Slice LUTs Used
Conventional Unsigned Wallace tree with ADFA, ADFA4, ADFA5	8bit	109/92152
Modified Wallace tree with ADFA, ADFA4, ADFA5	8bit	101/92152
Conventional Unsigned Wallace tree with NRFA, NRFA4, NRFA5	8bit	110/92152
Modified Wallace tree with NRFA, NRFA4, NRFA5	8bit	106/92152

TABLE VI: Comparison of Area (Spartan 3E XC3S500E, FG320, (-5)

Type of multiplier	Width	Slice LUTs Used
Conventional Unsigned Wallace tree with ADFA, ADFA4, ADFA5	8bit	83/4656
Modified Wallace tree with ADFA, ADFA4, ADFA5	8bit	82/4656
Conventional Unsigned Wallace tree with NRFA, NRFA4, NRFA5	8bit	84/4656
Modified Wallace tree with NRFA, NRFA4, NRFA5	8bit	84/4656

Fig. 13: Modified Ripple Multiplier with wallace using ADFA, ADFA4, ADFA5

HWALLnor8x8 Project Status (01/19/2016 - 04:58:33)			
Project File:	Multiplier.xise	Parser Errors:	No Errors
Module Name:	MWALLnor8x8	Implementation State:	Translated
Target Device:	xc6vlx75tl-1ff484	• Errors:	No Errors
Product Version:	ISE 14.7	• Warnings:	No Warnings
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	106	46560	0%
Number of fully used LUT-FF pairs	0	106	0%
Number of bonded IOBs	32	240	13%

Detailed Reports					
Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Tue Jan 19 04:57:56 2016	0	0	0
Translation Report	Current	Tue Jan 19 04:58:27 2016	0	0	0

HWALLnor8x8 Project Status (01/19/2016 - 06:13:35)			
Project File:	Multiplier.xise	Parser Errors:	No Errors
Module Name:	MWALLnor8x8	Implementation State:	Translated
Target Device:	xc6slx150-3fgg484	• Errors:	No Errors
Product Version:	ISE 14.7	• Warnings:	No Warnings
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	106	92152	0%
Number of fully used LUT-FF pairs	0	106	0%
Number of bonded IOBs	32	338	9%

HWALLnor8x8 Project Status (01/19/2016 - 06:18:21)			
Project File:	Multiplier.xise	Parser Errors:	No Errors
Module Name:	MWALLnor8x8	Implementation State:	Translated
Target Device:	xc3s500e-5fg320	• Errors:	No Errors
Product Version:	ISE 14.7	• Warnings:	No Warnings
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	84	4656	1%
Number of 4 input LUTs	146	9312	1%
Number of bonded IOBs	32	232	13%

Fig. 14: Modified Ripple Multiplier with wallace using NRFA, NRFA4, NRFA5

VI. CONCLUSION

The results shows that the proposed structure is faster compared to the existing Wallace tree and also the area or slices occupied by the Virtex 6 Low power XC6VLX75TL, FF484 (-1), Spartan 6 XC6SLX150, FGG484, (-3), Spartan 3E XC3S500E, FG320, (-5) is more efficient.

REFERENCES

[1] Bajaj, R.; Chhabra, S.; Veeramachaneni, S.; Srinivas, M.B., "A novel, low-power array multiplier architecture," in Communications and Information Technology, 2009. ISCIT 2009. 9th International Symposium on pp.119-123, 28-30 Sept. 2009.

[2] P. Burton and D.R. Noaks, High-Speed Iterative Multiplier, Electronics Letters, vol. 4, pp 262, 1968.

[3] J.M.Rabaey, A.Chandrakasan, and B.Nicolic, Digital Integrated Circuits, (2nd Edition) Prentice Hall, 2002.

[4] Wayne Wolf, (2002). Modern VLSI Design: System-On-Chip Design. 3rd Edition, Prentice Hall.

[5] J. Hoffman, G. Lacaze, and P. Csillag, Iterative Logical Network for Parallel Multiplication, Electronics Letters, vol. 4, pp 178, 1968.

[6] H. Guilt, Fully Iterative Fast Array for Binary Multiplication, Electronics Letters, vol. 5, pp 263, 1969.

[7] R. De Mori, Suggestion for an IC Fast Parallel Multiplier, Electronics Letters, vol. 5, pp 50-51, Feb. 1969.

[8] J. Hoffman, G. Lacaze, and P. Csillag, Iterative Logical Network for Parallel Multiplication, Electronics Letters, vol. 4, p. 178, 1968.