International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 9, September- 2014

Implementation of 8 Bit and 12 Bit Mulitpliers using 4-2 and 5-2 Compressors

Anushka Pothukuchi[1]; Disha Agarwal[2] ; Ankita Keshari[3]
School of Electrical Engineering
Vellore Institute of Technology
Vellore, India

***Abstract*--This paper aims to compare different structures of 8x8 and 12x12 binary multipliers. One structure uses full adders and half adders to add the partial products. The other (modified) structure explores the use of 4-2 and 5-2 compressors. The multipliers are implemented using Xilinx ISE Design Suite 12.3 and the timing, cell area and power are compared using Cadence RTL Compiler. It is shown that the modified method has better performance parameters.**

Keywords—compressor; multiplier; Wallace tree; cell area; power; timing

I. INTRODUCTION

Multiplication forms the basis of Arithmetic Logic Units which are used in several microprocessors. It also finds use in implementation of Digital Signal Processing algorithms. [1] Reducing timing of the multiplier structures is essential because it contributes to the critical path delay of the above systems. At the same time, cell area and power consumption must also be considered.

Multiplication of binary bits is done by accumulating partial products. This is conventionally done using full adders and half adders. Different algorithms like the Booth algorithm, Ripple Carry and Carry Save are used. The number of stages while adding the partial products determines the performance of the multiplier in terms of delay, power and area. [5]

In this paper, the Wallace Tree structure is used to accumulate partial products. The structure employs carry save form. Here, the carries generated in the adders are passed on to the next stage, until the last stage. At the last stage, ripple carry form is used to generate the final product. The Wallace Tree implementation was preferred because it has less critical path, hence reducing delay.

II. 4-2 COMPRESSOR ARCHITECTURE

A 4-2 compressor is used for the addition of four binary bits. It takes in four primary inputs, plus one input carry from the neighbouring cell. The outputs are Sum, Carry and an auxiliary carry that is taken as input to the next 4-2 compressor. [1]

Here, four inputs are compressed to two outputs. Compression ratio is thus 4/2=2. This is higher than that of the full adder.

When a 4-2 compressor is used, it involves less wiring compared to when two full adders are used separately.

One implementation of 4-2 is as given below:

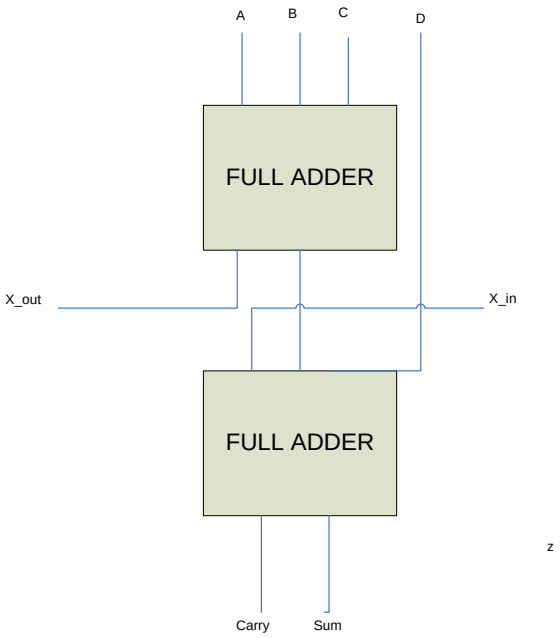

Fig. 1. Implementation of 4-2 Compressor using 3-2 Compressors

Boolean Expressions

$$\text{Sum} = A \oplus B \oplus C \oplus D \oplus X_{in} \quad (1)$$

$$\text{Carry} = (A \oplus B \oplus C \oplus X_{in}).D + (A \oplus B \oplus C).X_{in} \quad (2)$$

$$X_{out} = (A \oplus B).C + A.B \quad (3)$$

III. 5-2 Compressor Architecture

A 5-2 compressor is used for the addition of five binary bits. It takes five primary inputs, and two additional carries from the neighbouring cell. It has two main outputs, Sum and Carry, and two auxiliary carries which are inputs to the next compressor. [1]

Here five inputs are compressed to two outputs. Hence the compression ratio in this case is 5/2=2.5.

Using a 5-2 compressor reduces wiring more than that done by a 4-2 compressor.

One implementation of the 5-2 compressor is:

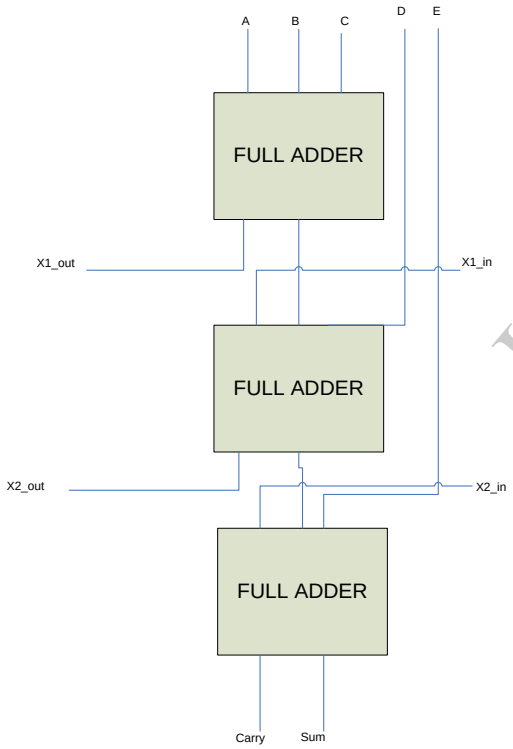

Fig. 2. Implementation of 5-2 Compressors using 3-2 Compressors

Boolean Expressions

$$\text{Sum} = A \oplus B \oplus C \oplus D \oplus E \oplus X1_{in} \oplus X2_{in} \quad (4)$$

$$\text{Carry} = (A \oplus B \oplus C \oplus D \oplus X1_{in} \oplus X2_{in}).E + (A \oplus B \oplus C \oplus D \oplus X1_{in}).X2_{in} \quad (5)$$

$$X1_{out} = (A \oplus B).C + A.B \quad (6)$$

$$X2_{out} = (A \oplus B \oplus C \oplus X1_{in}).D + (A \oplus B \oplus C).X1_{in} \quad (7)$$

IV. 8x8 Multiplier

8 bit multiplication was implemented using the Wallace Tree structure. Wallace tree employs the carry save approach, where carries that are generated in a particular stage go to the next stage instead of going to the next cell in the same stage (ripple carry adder).

A. Conventional Method

First, full adders and half adders were used for partial product accumulation.

The structure used (where dots represent adders) was:

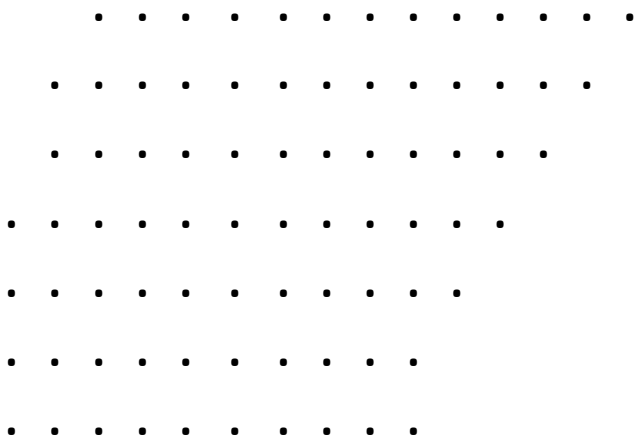

Fig. 3. Implementation of 8x8 multiplier by conventional method

B. Modified Method

This method shows 8 bit multiplication using the 4-2 and 5-2 compressors to improve performance of the multiplier.

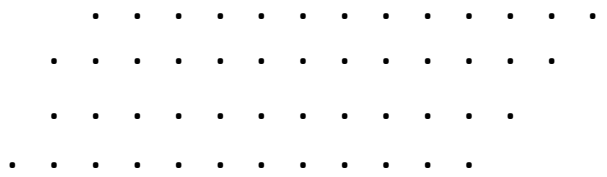

Fig. 4. Implementation of 8x8 multiplier using compressors

V. 12X12 MULTIPLIER

12 bit multiplication was also implemented similarly, using Wallace tree.

A. Conventional Structure

The structure given below uses full adders and half adders for partial product generation. The dots represent these adders.

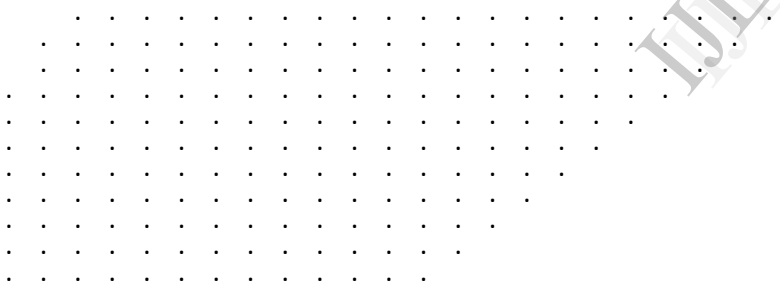

Fig. 5. Implementation of 12x12 multiplier by conventional method

B. Modified Method

This structure uses 4-2 and 5-2 compressors:

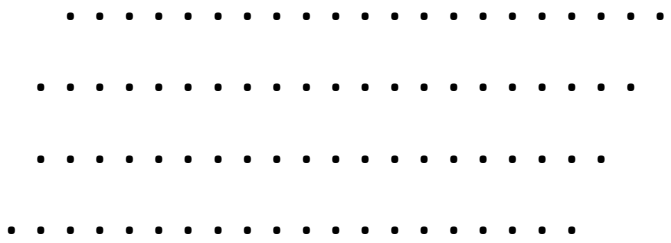

Fig. 6. Implementation of 12x12 multipliers using compressors

VI. SIMULATION AND RESULTS

Xilinx ISE Design Suite 12.3 was used to implement and simulate the 8x8 multiplier and 12x12 multiplier structures in order to test their functionality.

Cadence RTL Compiler was used to compare the different design parameters (power, cell area and delay) of the multipliers.

The results are as follows:

TABLE I RESULTS

	8x8 Multiplier			12x12 Multiplier		
Structure	***Cell Area***	***Power (nW)***	***Timing (ps)***	***Cell Area***	***Power (nW)***	***Timing (ps)***
Conventional	1306.066	113129.504	2696	3164.616	348587.857	3902
Using compressors	1238.32	112744.294	2693	2930.357	334162.882	4042

VII. CONCLUSION

The aim of this paper was to compare the timing, power and cell area between multipliers designed by two different structures. The first structure employed was the conventional method using full adders and half adders. The modified structure used 4-2 and 5-2 compressors also.

As per the results, the modified structure showed improved performance in two out of three parameters (area and power), with a trade-off on timing.

ACKNOWLEDGMENT

We would like to thank Prof. Balamurugan S and Prof. Marimituhu R of VIT University for their guidance and support.

REFERENCES

[1] Karuna Prasad and Keshab K Parhi, "Low power 4-2 and 5-2 Compressors", Signals, Systems and Computers, 2001. Conference Record of the Thirty Fifth Asilomar Conference, pp 129-133 vol 1, 4-7 Nov 2001

[2] Neil H.E. Weste and K. Eshraghian, "Principles of CMOS VLSI Design", Wesley Publishing Company 1993

[3] Chip-Hong Chang; Jiangmin Gu "Low voltage,low power (5:2) compressor cell for fast arithmetic circuits", Acoustics, Speech and Signal Processing 2003. Proceedings (ICASSP '03). IEEE International Conference, pp 661-4 vol 2, April 2003

[4]Anandi, V; Rangarajan, R; Ramesh, M "Low Power VLSI Compressors", Green Computing, Communication and Conservation of Energy (ICGCE), 2013 IEEE International Conference, pp 231-236, Dec. 2013

[5] Veeramachaneni, S; Krishna M.K; Avinash, L; Puppala, S.R; Srinivas, M.B, "Novel Architectures for High Speed and Low-Power 3-2, 4-2 and 5-2 Compressors", VLSI Design, 2007, 6th International Conference on Embedded Systems, 20th IEEE International Conference, pp 324-329, Jan. 2007

[6] Sureka, N; Porselvi, R; Kumuthapriya, K. "An efficient high speed Wallace tree multiplier", Information Communication and Embedded Systems (ICICES), 2013 IEEE International Conference, pp 1023-1026, Feb. 2013

[7]Aguirre-Hernandez, M.; Linares-Aranda, M. "CMOS Full-Adders for Energy-Efficient Arithmetic Applications", Very Large Scale Integration (VLSI) Systems, IEEE Transactions, pp 718-721 Vol. 19 Issue 4, April 2011

[8] Quintana, J.M.; Avedillo, M.J.; Jiménez, R.; Rodriguez-Villegas, E. "Low-power logic styles for full-adder circuits", Electronics, Circuits and Systems, 2001. ICECS 2001. 8th IEEE International Conference, pp 1417-1420, Vol. 3, Sep 2001

[9] Shen-FuHsiao; Ming-RounJiang; Jia-SienYeh, "Design of high-speed low-power 3-2 counter and 4-2 compressor for fast multipliers", Electronics Letters, Vol. 34, Issue 4, p. 341-343 Feb. 1998

[10] Keshab K. Parhi, VLSI Digital Signal Processing Systems, John Wiley and Sons, Inc., 1999