

Speed and area analysis on hierarchy multiplier

Moon Cheng Wong, Razaidi Hussin

School of Microelectronic Engineering, Universiti Malaysia Perlis, Pau Putra Campus, 02600 Arau, Perlis, Malaysia.

Abstract. This paper proposes designs of hierarchy multiplier by utilising different designs on 4:2 and 7:3 compressor and multiple compressors. The hierarchy multipliers is optimised in the term of speed or area of hierarchy multiplier by redesigning 4:2 compressor units and introducing a combination of 4:2 compressor and 7:3 compressor units in a Vedic multiplier block. All designs are simulated using Altera Quartus II software. The aim of this paper is to improve the performance in speed by moderately increasing the area without considering the power consumption. The proposed design is 4.5% to 8.3% faster and consumes -0.5% to 5.8% less area.

1 Introduction

Multiplication is a basic arithmetic operation in processor, computation and digital signal processing, image processing application and so on. High processing speed is needed for fast multiplication, and complexity of VLSI design increases every 18 months to satisfy Moore’s law in VLSI advanced technology [2-4].

Partial production generation, partial product reduction and partial production addition are three basic stages of multiplication. First, the multiplicand and multiplier are multiplied to generate partial products. Then, 4:2 or 7:3 compressor units are used to reduce the delay of partial production [5-7]. Finally, the final addition is done by Binary-to-Excess-one converter (BEC). To optimise speed, Carry Save Adder tress is implemented by utilising 4:2 and 7:3 compressor. The detail will be discussed in the next section.

Hierarchy multiplier make use of fine-grained parallelism to reduce propagation delay of computer intensive operation. Four Vedic base multiplier in hierarchical are allowed to perform multiplication in parallel to speed up the hierarchical multiplier. The compressor blocks are implemented in the four Vedic blocks [1].

2 Hierarchy multiplier

A 16-bit hierarchy multiplier make use of four 8-bit base multiplier units using Vedic multiplication algorithm. Vedic multiplication is an ancient Indian Mathematics which is based on Urdhva Tiryakbhyam algorithm. The Vedic multiplier is time, area and power efficient due to the gate delay and space increase slowly when increasing the number of Vedic multiplier bit compared to other multipliers [8-10]. All the base multipliers consist of AND-gates, compressors, half adders and full adders.

Furthermore, they also contain CSA adder, Carry Select Adder (CSIA) and BEC with multiplexer (MUX) for final addition of partial product [1]. However this paper is only focusing on how is the implementation of different design of 4:2 compressors and 7:3 compressors, as well as the use of both compressors in an architecture in base multipliers to improve the performance of hierarchy multiplier.

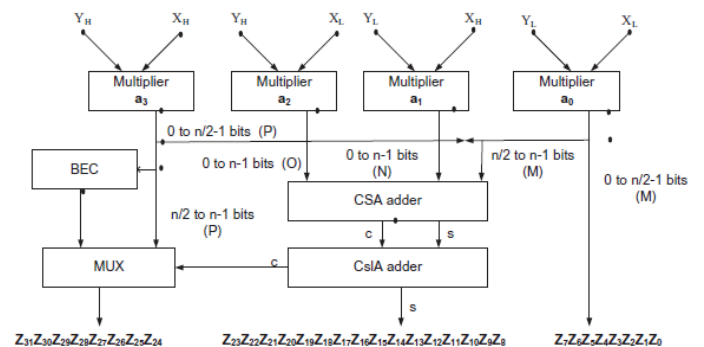


Fig. 1. Block diagram of hierarchy multiplier [1].

3 Base multiplier Architecture

A base multiplier is built from AND gates, compressors and adders. AND gates are used to generate partial product followed by compressors and adders, which are to reduce partial product. The base multiplier is illustrated in Figure 2.

Corresponding author: miowmc93@gmail.com

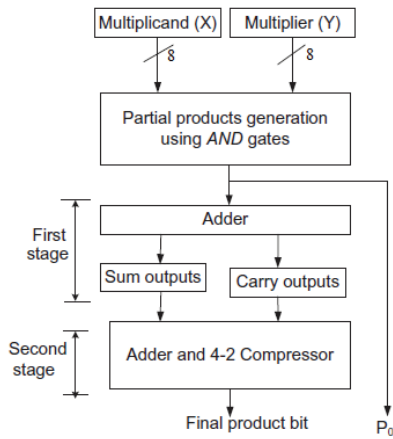


Fig. 2. Block diagram of a base multiplier [1].

3.1 A 4:2 Compressor Architecture

A standard 4:2 compressor is implemented by cascading two full adders unit as shown in Figure 3. A 4:2 compressor has four inputs (X1, X2, X3 and X4), carry input (Cin) and two outputs (Sum and Carry which along with Cout) [11-12]. The design in [1], design 1 and design 2 can be divided into two full adder blocks as shown in Figure 3.

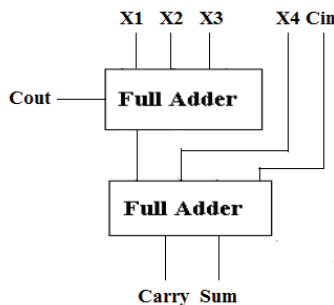


Fig. 3. Block diagram of 4:2 compressor.

3.1.1 Design in journal [1]

The 4:2 compressor in [1] utilises cascading of two full adder unit which both has two XOR gates and a MUX respectively [1]. The 4:2 compressor can be divided into two groups of full adders as shown in Figure 4.

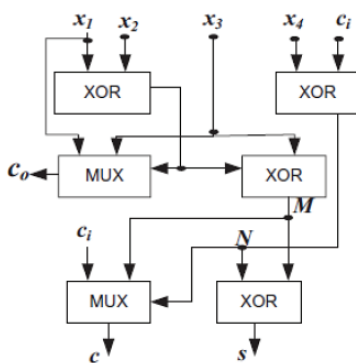


Fig. 4. Block diagram of 4:2 compressor in [1].

$$Sum = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus c_{in} \quad (1)$$

$$c_{out} = \overline{(x_1 \oplus x_2)}x_1 + (x_1 \oplus x_2)x_3 \quad (2)$$

$$Carry = \overline{(x_4 \oplus c_{in})}c_{in} + (x_4 \oplus c_{in})(x_1 \oplus x_2 \oplus x_3) \quad (3)$$

3.1.2 Design 1

Cascading two Mux-Add units is similar as cascading two full adders in order to implement a 4:2 compressor. The mux-add in [13] suffers in number of transistors. The modified Mux-Add reduces transistor count. A XOR or XNOR consists of eight CMOS transistors respectively, both AND or OR use six CMOS transistors, MUX has four CMOS gates and NOT has two transistors. The total number of transistors in a full adder block is summarised in Table 1. The larger the design, the less the number of transistors will save cost.

Table 1: Comparison a MUX-Add unit between [13] and a Full adder unit in design 1.

Design	Number of transistor (CMOS)
[13]	36
Design 1	30

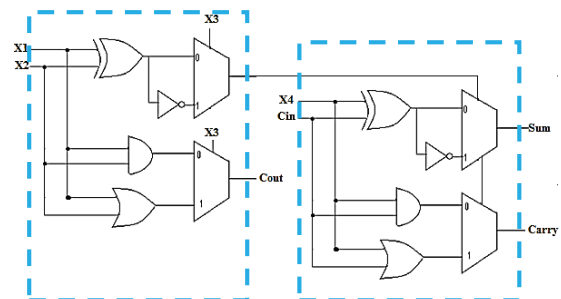


Fig. 5. Modified 4:2 compressor using two Mux-Add units.

Equations of design 1 is as below:

$$Sum = \left[\overline{(x_1 \oplus x_2)}\overline{x_3} + \overline{(x_1 \oplus x_2)}x_3 \right] \overline{(x_4 \oplus c_{in})} + \left[\overline{(x_1 \oplus x_2)}\overline{x_3} + \overline{(x_1 \oplus x_2)}x_3 \right] (x_4 \oplus c_{in}) \quad (4)$$

$$c_{out} = (x_1x_2)\overline{x_3} + (x_1 + x_2)x_3 \quad (5)$$

$$Carry = \overline{\left[\overline{(x_1 \oplus x_2)}\overline{x_3} + \overline{(x_1 \oplus x_2)}x_3 \right]}(x_4c_{in}) + \left[\overline{(x_1 \oplus x_2)}\overline{x_3} + \overline{(x_1 \oplus x_2)}x_3 \right] (x_4 + c_{in}) \quad (6)$$

3.1.3 Design 2

Design 2 is designed with same number of CMOS transistors in journal 1 as shown in table2. This new 4:2 compressor utilises the sum part of Mux-Add (Figure 5)

and carry part of the design in [1] (Figure 4) as shown in Figure 6.

Table 2: Comparison a full adder unit between [1] and a Full adder unit in design 2.

Design	Number of transistor (CMOS)
Journal 1	16
Design 2	16

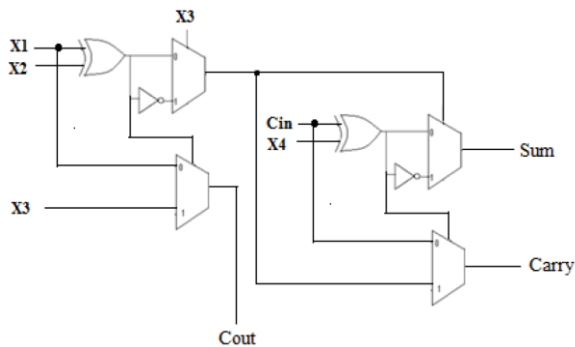


Fig. 6. A new 4:2 compressor design.

$$\text{Sum} = \text{equation (4)}$$

$$c_{out} = \text{equation (2)}$$

$$\text{Carry} = (\overline{x_4 \oplus c_{in}})c_{in} + (x_4 \oplus c_{in})[(x_1 \oplus x_2)x_3 + (x_1 \oplus x_2)\overline{x_3}] \quad (7)$$

4 A New Base Multiplier Architecture

A new base multiplier architecture is almost similar with Figure 2 but it uses a combination of compressors which are 4:2 compressor and 7:3 compressor for optimization. The architecture is shown in Figure 7.

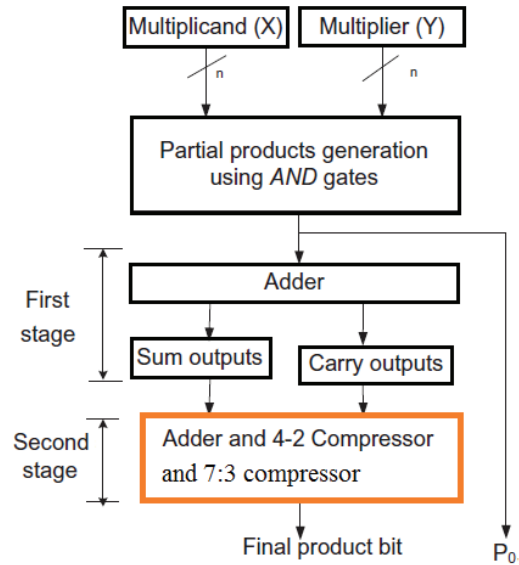


Fig. 7. Block diagram of a new base multiplier.

4.1 A 7:3 Compressor Architecture

A 7:3 compressor consists of four Full Adder (FA) blocks. The carry and the summation of inputs (X1-X6) are cascaded to different next stage full adder respectively. The FA which carries summation output from the first two FA will add up with X7 to produce sum output. The summation of carries from the first two full adders will produce output Carry and Cout. However the carry ($Carry_{i+1}$) is propagate to next two stage. The 7:3 compressor allows parallel addition when the sum and carry propagate to different next stage of FA. The 7:3 compressor is used in partial product reduction, so seven partial products which are generated from AND gate can be added up at the same time. A standard 7:3 compressor is shown in Figure 8. This section highlights the existence of 7:3 compressor for summation of seven partial products in parallel at the centre of compressor tree, which lead to faster multiplication process. This is due to hidden critical path of carry signal that is reduced by optimising the connection of compressor topology [14-16].

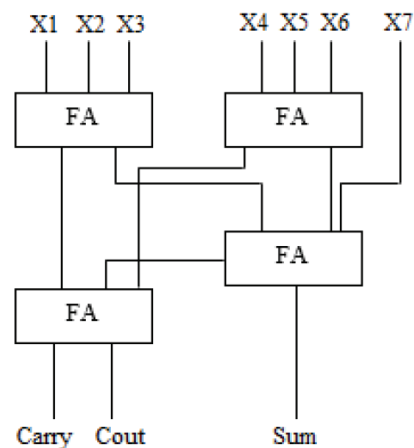


Fig. 8. A standard 7:3 compressor design [14-15].

4.1.1 Design 3

Design 3 is implemented using the design in journal [1] (Figure 4) with only a full adder unit design and duplicated by four times. The block diagram of design 3 is shown in Figure 9. Each full adder unit contains two XOR gates and a MUX.

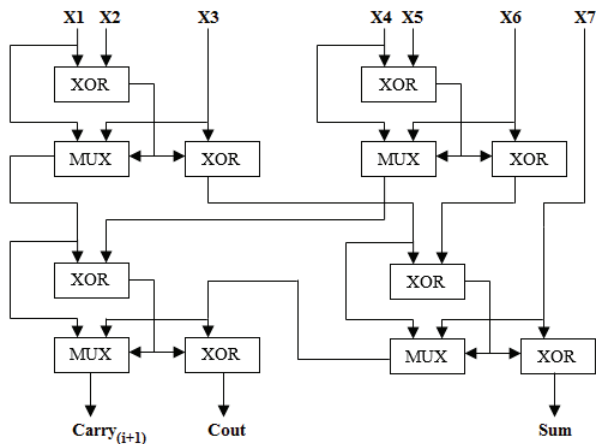


Fig. 9. A 7:3 compressor design by utilise a full adder design unit in [1].

4.1.2 Design 4

Design 4 utilises only a full adder block in design 2 and connect the other three full adder units as shown in Figure 10. Each full adder consists of a XOR gate, a NOT gate and two MUX.

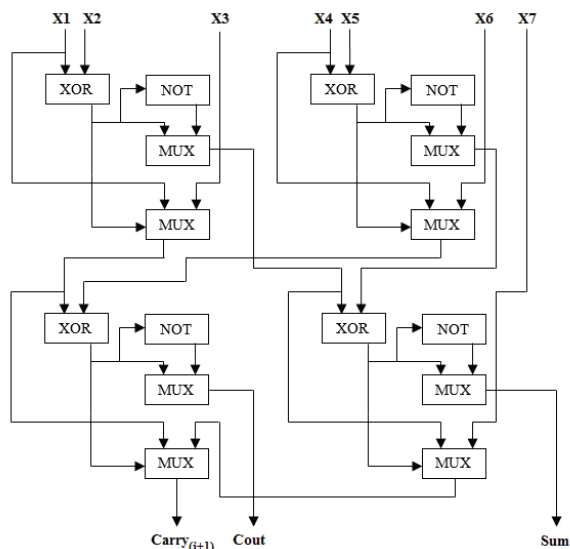


Fig. 10. A 7:3 compressor design by utilise a full adder design 2.

5 Simulation result and Analysis

All simulation is simulated in EP2C70F896C6 device from Cyclone II family in Altera Quartus II software by describing using Verilog Hardware Description Language. The functional verification of the 16-bit hierarchy multiplier is done by simulator tools in the

software. The performance is analysed after each design is successfully debugged.

5.1 Functional Verification

The function of multiplier is verified before proceeding to performance analysis. The inputs are set from the lower to the higher range to ensure that the multiplier perform correct output and propagation of carry to every components is accurate as shown in Figure 11. The boundary condition is considered in this design since the use of four base multipliers in performing multiplication. The consideration of boundary condition is to verify that the input is propagating properly between base multiplier blocks. A set of inputs is used to verify the functionality of boundary issue as shown in Figure 12.

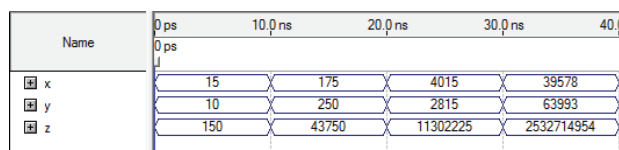


Fig. 11. Waveform from lower to higher range.

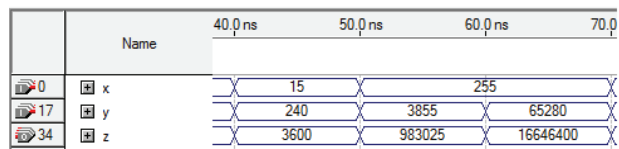


Fig. 12. Waveform for boundary condition.

5.2 Performance Analysis

The longest total propagation delay (tpd) and total logic elements or area is used to analyse the performance of hierarchy multiplier. The performance is summarized in Table 3. Since the aim of this paper is to optimise speed and area of the hierarchy multiplier, the percentage (%) improved is calculated using the formula below.

$$\% \text{ improved} = \frac{\text{Performance of journal [1]} - \text{performance of each design}}{\text{Performance of journal [1]}} \times 100\%$$

Table 3: Comparison the performance among designs.

Design	Longest delay tpd (ns)	% improved in tpd	Total logic elements	% improved
Journal [1]	32.758	-	636	-
Design 1	31.297	4.5	599	5.8
Design 2	31.102	5.1	619	2.7
Design 3	30.252	7.7	639	-0.5
Design 4	30.030	8.3	636	0.0

From Table 3, both design 1 and design 2 are improved in terms of speed and area compared to journal [1] design. Design 1 reduce in 4.5% delays and consume less 5.8% area as well as design 2 improve 5.1% in terms of speed and 2.7% less area compare to journal [1]. Nevertheless, design 3 sacrifices 0.5% area and design 4 has no impact on area but both designs perform more significant effect in optimising the speed which are 7.7% and 8.3% respectively.

Design 1 is suggested for designers who suffer in limitation of size without increase in latency but designer who have plenty of space is more suitable to use design 4. Design 2 is more appropriate to have overall performance of hierarchy multiplier.

Conclusion

An efficient hierarchy multiplier is developed by redesigning 4:2 compressor and utilities multiple compressors which are 4:2 compressor and 7:3 compressor in partial production reduction. The use of design 1 to modify the 4:2 compressor architecture consumes less 5.8% area than previous design in journal [1]. The proposed designs improve the speed of hierarchy multiplier from 4.5% to 8.3%. The proposed multiple compressor has disadvantage in area consumption compare to design 1 and design 2, but it is faster compare to only using 4:2 compressor. Design 2 performs the best in terms of speed and area, with reduction of 5.1% and 2.7% respectively.

We appreciate and thanks Universiti Malaysia Perlis for providing financial support to produce this paper.

References

1. M. Shoba and R. Nakkeeran, *Eng. Sci. Technol. an Int. J.*, **20**, no. 1, pp. 321–331 (2017)
2. P. J. Denning and T. G. Lewis, *Communications of the ACM*, **60**, no. 1, pp. 54–65 (2016)
3. S. Saravanan, I. Vennila, and S. Mohanram, *Circuits and Systems*, **7**, pp. 630–642 (2016)
4. J. Wu et al., *Appl. Comput. Intell. Soft Comput.*, **2013**, pp. 1–13 (2013)
5. P. D. Gopineedi, H. Thapliyal, M. B. Srinivas, and H. R. Arabnia, *Proceedings of the 2006 International Conference on Embedded Systems & Applications*, pp. 1–7 (2006)
6. K. Prasad and K. K. Parhi, *Asilomar Conf. Signals, Syst. Comput.*, pp. 129–133 (2001)
7. R. Hussin, A. Y. M. Shakaff, N. Idris, Z. Sauli, R. C. Ismail, and A. Kamarudin, *2008 Int. Conf. Electron. Des. ICED*, (2008)
8. D. Kayal, P. Mostafa, A. Dandapat, and C. K. Sarkar, *J. Signal Process. Syst.*, **76**, no. 1, pp. 1–9 (2014)
9. S. R. Vaidya, D. R. Dandekar, *Proceedings of the 2011 International Conference on Communication, Computing & Security*, pp. 383–386 (2011)
10. N. B. Reddy and K. Wasimakram, *International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)* **19**, no. 1 (2015)
11. T. U. Anusree and P. L. Bonifus, *International Journal of Computer Trends and Technology (IJCTT)* **36**, no. 4, pp. 213–218 (2016)
12. A. Pishvaie, G. Jaberipur, and A. Jahanian, *Canadian Journal of Electrical and Computer Engineering*, **36**, no. 3, pp. 6–8 (2013)
13. I. Banerjee, *Int. Journal of Engineering Research and Applications*, **3**, no. 6, pp. 75–78 (2013)
14. S. Mehrabi, K. Navi, and O. Hashemipour, *2013 IEEE 5th International Nanoelectronics Conference (INEC)*. pp. 322–325 (2013)
15. N. Ravi, T. Jayachandra Prasad, M. Umamahesh, and T. Subba Rao, *Proc. Int. Conf. "Recent Adv. Sp. Technol. Serv. Clim. Chang. - 2010", RSTS CC-2010*, **8**, no. 3, pp. 189–193 (2010)
16. V. Oklobdzija, D. Villeger, and S. Liu, *IEEE Transactions on Computers*, **45**, no. 3, pp. 294–306 (1996)