International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

Comparison of Various Adder Designs in terms of Delay and Area

Khushboo Bais[1], Zoonubiya Ali[2]

[1]M. Tech. Scholar, Department of Electronics & Telecommunication Engineering,
Disha Institute of Management & Technology, Raipur-492 001, Chhattisgarh, India

[2]Associate Professor, Department of Electronics & Telecommunication Engineering,
Disha Institute of Management & Technology, Raipur-492 001, Chhattisgarh, India

Abstract: ***VLSI designers are constantly working towards the optimization of speed, power, and area of circuits, but practically it is difficult to optimize all at the same time. This paper presents a comparative study of the designs of parallel adders- ripple carry adder, carry look-ahead adder and Kogge-Stone adder, which have been designed using Xilinx ISE 14.7 Design Suite and synthesized for Spartan 3 FPGA. All the adders have been designed for 4-bit, 8-bit, and 16-bit operands and a comparison of delay performance and area utilization has been made as per the data obtained from the synthesis results. The effect of parallelism on speed and area of adder designs has been analysed, and it has been observed that both the parameters cannot be optimized at the same time. If parallelism is increased in order to increase the speed of operation, then it will result in large area occupancy; and if area is to be optimized then we have to adjust with the slow speed of system.***

Keywords: Ripple Carry Adder (RCA), Carry Look-Ahead Adder (CLA), Parallel Prefix Adders (PPA), Xilinx ISE, Spartan 3.

1. Introduction

With the advancements in VLSI technology, the circuit designs are getting miniature in size, consuming lesser power for performing their intended operation and becoming faster in operation. We all know that area, power and speed are the major constraints in VLSI design, and the designers are taking enormous efforts to improve their designs relative to these constraints, but all of these cannot be improved simultaneously.

In this modern era of technological advancements, everything is becoming fast-paced and heading towards completely digital processes. Hence, there is an immense need of developing faster processors which would operate on digital signals, but as we head towards improving any one of the design parameter, the other parameters are also affected, and so with the improvement in speed of operation of any circuit, its area occupancy also increases.

In circuits like digital signal processor (DSP), microprocessor, or arithmetic and logic unit (ALU) of any processor, the unit performing arithmetic operations is very important when considered with respect to the design constraints mentioned above. Most arithmetic circuits consist of adder, subtractor, multiplier, divider, etc.; the adder unit being the most basic unit among all the other units.

In this paper, we are comparing various adders in terms of their delay and area, as the adder is used in the construction of other arithmetic circuits and the performance of the adder is decisive of the performance of other circuits employing the use of adders.

2. Multi-bit Adders

The half-adder and the full-adder are the simplest addition elements which are limited to single-bit addition. For performing multi-bit addition, we need to cascade multiple full-adder (FA) units.

2.1 Ripple Carry Adder (RCA)

The simplest multi-bit adder is the ripple-carry adder (RCA), as shown in figure 1 [1]-[3]. RCA, although capable of performing multi-bit addition, increases the processing delay, as the carry signal has to ripple all the way from first adder to the last to produce the sum and carry output of the given operands.

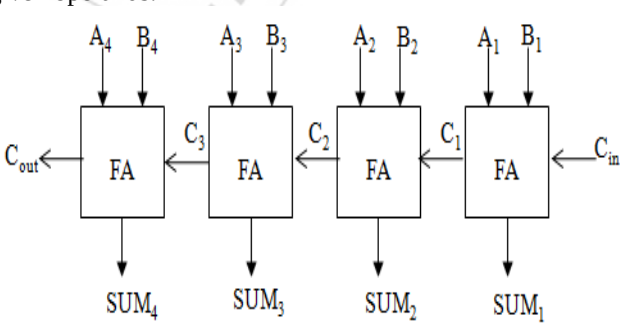

Figure 1: Block Diagram of a 4-bit Ripple Carry Adder

2.2 Carry Look-ahead Adder (CLA)

To reduce the propagation delay of the carry signals, the concept of carry look-ahead adder (CLA) was introduced, which calculates all the carries in parallel, using the concept of generate and propagate signals [1], [2]. The generate (G_n) and propagate (P_n) signals in [1], [2], [4]-[6] are given by,

$$G_n = A_n \bullet B_n \quad (1)$$

$$P_n = A_n \oplus B_n \quad (2)$$

The CLA is advantageous when compared to RCA as it calculates all the carry bits parallely with the help of the generate and propagate signals. The carry signals in the subsequent stages of an n-bit CLA depend only on the input carry, augend and addend bits. The carry signal for the CLA in [1] is given by

$$C_n = G_{n-1} + P_{n-1} \bullet C_{n-1} \quad (3)$$

and the value of sum in [1] is given by,

$$SUM_n = P_n \oplus C_{n-1} \quad (4)$$

As the number of bits in the addend and augend increases, the complexity and delay of the CLA also increases. We can overcome this limitation by using 4-bit modules of CLA [5] or we have another class of adder circuits, known as the parallel prefix adders, or the carry-tree adders, or logarithmic adders.

2.3 Parallel Prefix Adders

The parallel prefix adders (PPAs) are known for their efficient and performance-oriented designs. These adders are less complex as well as faster in operation than the previously described adder configurations. The flow of operation of a parallel prefix adder can be understood from figure 2 [6].

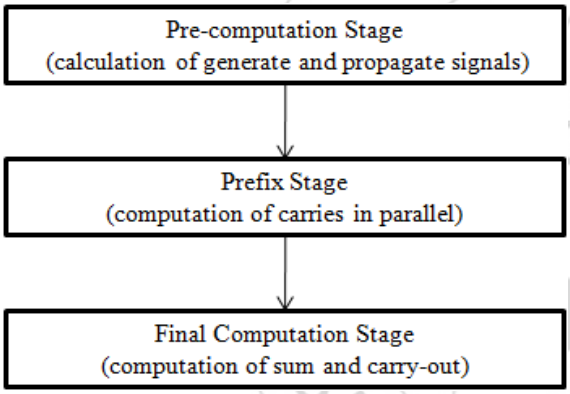

Figure 2: Parallel Prefix Addition Process

In the pre-computation stage, the generate and propagate signals are calculated using equations (1) and (2), respectively.

In the prefix stage, group generate and group propagate signals are calculated with the help of a fundamental carry operator, or a prefix operator. Using the fundamental carry operator, the group generate and propagate signals are calculated in [2], [7] as follows:

$$(G_L, P_L) \circ (G_R, P_R) = (G_L + P_L \bullet G_R, P_L \bullet P_R) \quad (5)$$

The fundamental carry operator is split into two categories: black cells (BC) and grey cells (GC), which are shown in figure 3 [2], [4]-[6].

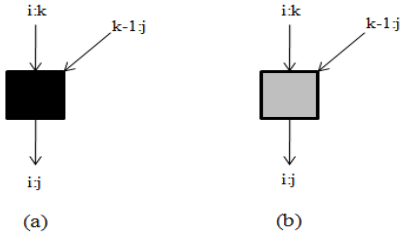

Figure 3: Symbolic Representation of the Black Cell (BC) and the Grey Cell (GC)

The BC produces both, group generate and group propagate signals, at its output, and the GC produces only group generate signal at its output. The group generate, $G_{i:j}$ and group propagate, $P_{i:j}$ signals in [2], [4]-[6], [8] are given as

$$G_{i:j} = G_{i:k} + P_{i:k} \bullet G_{k-1:j} \quad (6)$$

$$P_{i:j} = P_{i:k} \bullet P_{k-1:j} \quad (7)$$

In the final computation stage, the sum output is produced using full-adder units.

The parallel prefix adders are better when compared to CLA because it takes less number of steps for a carry to be calculated using PPA than a CLA. For example, in case of a 4-bit CLA, the final carry in [2], [7] is given by,

$$C_4 = (G_4, P_4) \circ [(G_3, P_3) \circ [(G_2, P_2) \circ (G_1, P_1)]] \quad (8)$$

and, considering PPA, the final carry in [2], [7] is given by,

$$C_4 = [(G_4, P_4) \circ (G_3, P_3)] \circ [(G_2, P_2) \circ (G_1, P_1)] \quad (9)$$

Hence, it is clear from equations (8) and (9) that for calculation of carry-out from the 4th bit, CLA requires 3 steps whereas PPA requires only 2 steps. Due to lesser number of required calculations, the delay of these adders is lesser, of the order of $\log_2 N$, for an N-bit adder [3], [7]. Hence, these adders are sometimes called as logarithmic adders.

There are various known parallel prefix adders like Kogge-Stone adder, Brent-Kung adder, Knowles adder, Sklansky adder, and many more. In this paper, Kogge-Stone adder is discussed, which would be representing the parallel prefix class of adders.

2.3.1 Kogge-Stone Adder

In the year 1973, Peter M. Kogge and Harold S. Stone, gave a parallel algorithm for solution for recurrence equations [9], which started being used in multi-bit addition for faster operation. The KSA tree structure, which has been shown in figure 4 [2], [4], [8] for 16-bit operands, calculates all the carries in parallel, thus enhancing the speed of the adder at the cost of area.

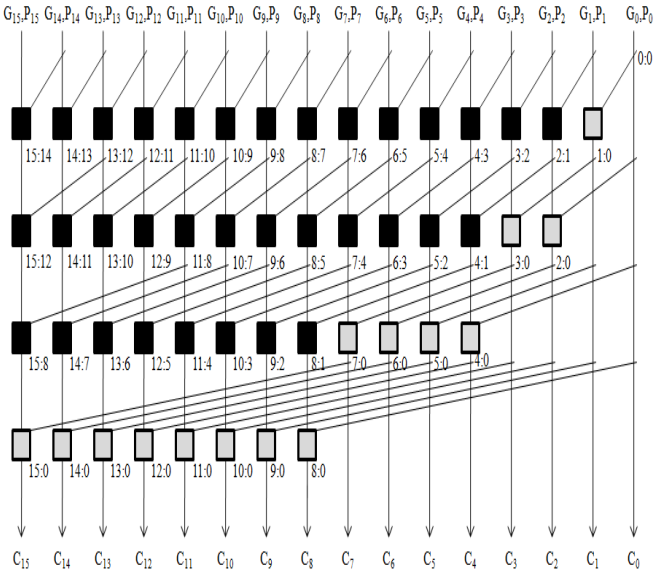

Figure 4: Prefix Stage for 16-bit Kogge-Stone Adder

Originally, the concept developed by Kogge and Stone used only black cells, but many researchers worked on the design to achieve faster speed with low area consumption and came up with a solution that replacing the last black cell in each column, with a grey cell will not produce any effect on the sum or carry, if we assume $G_{0:0} = C_{in}$ and $P_{0:0} = 0$ and proceed for further computations. Hence, from the carry tree or prefix stage of the KSA we can relate the carry signals with the group generate signals in [2], [4] by the following relation,

$$C_i = G_{i:0} \quad (10)$$

and the sum output in [2], [4], [6], [8] will follow the relation,

$$SUM_i = P_i \oplus G_{i-1:0} \quad (11)$$

3. Simulation Results

All the adder designs that have been discussed above, have been designed using Xilinx ISE Design Suite 14.7, synthesized for Spartan 3 FPGA (XC3S400-5PQ208) using XST, and simulated with ISim simulator, for 4-bit, 8-bit and 16-bit operands. The simulation results for 16-bit RCA, CLA and KSA are as shown in the figures 5, 6, and 7.

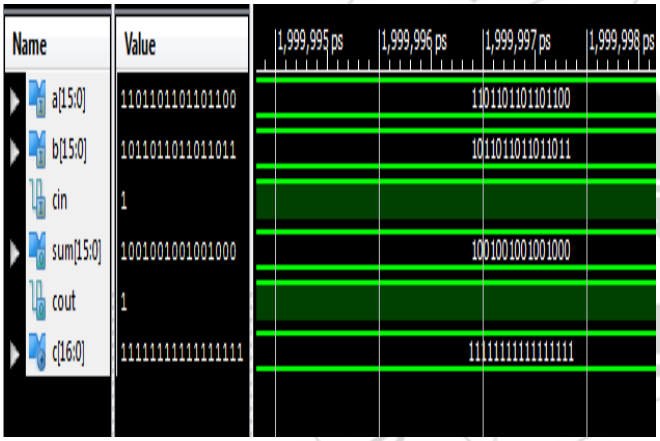

Figure 5: Waveform for 16-bit Ripple Carry Adder

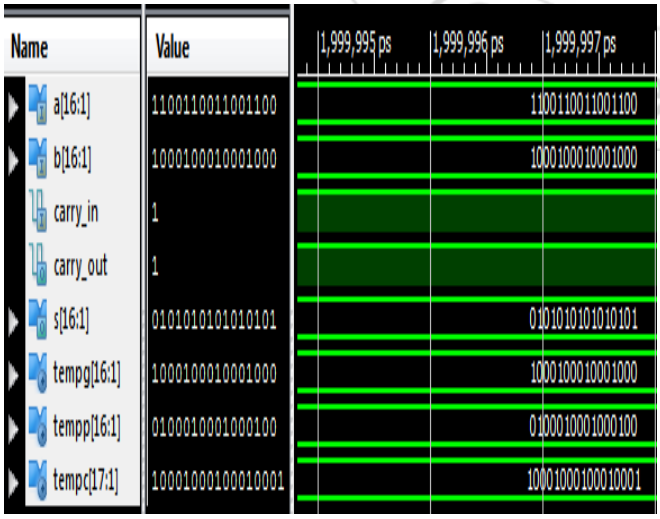

Figure 6: Waveform for 16-bit Carry Look-ahead Adder

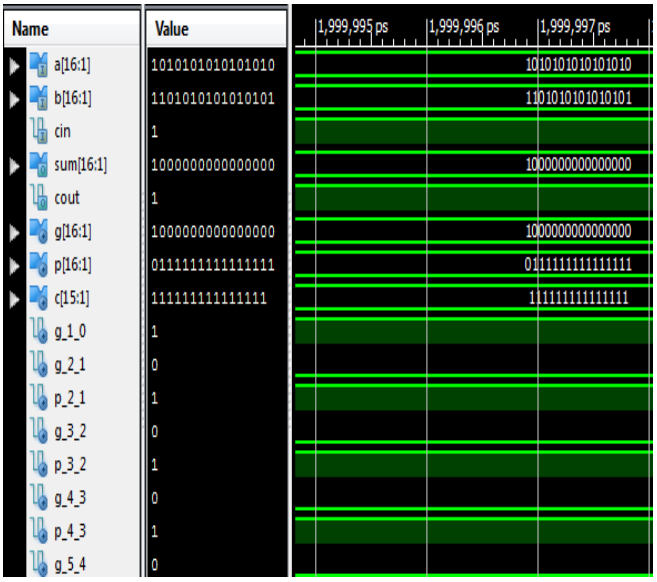

Figure 7: Waveform for 16-bit Kogge-Stone Adder

4. Comparison of Delay and Area of Adders

The delay values of all the adders have been listed in Table 1 and a graph (figure 8) has been plotted, which shows the comparison of the maximum combinational path delay of the three adders for different number of bits.

Table 1: Comparison of Delay of Adders

Type of Adder	Delay		
	Maximum Combinational Path Delay (in ns)	Logic Delay (in ns)	Route Delay (in ns)
4-bit RCA	12.008	7.540	4.468
4-bit CLA	12.008	7.540	4.468
4-bit KSA	11.786	7.540	4.246
8-bit RCA	17.585	9.456	8.129
8-bit CLA	17.585	9.456	8.129
8-bit KSA	17.030	9.456	7.574
16-bit RCA	28.741	13.288	15.453
16-bit CLA	27.076	13.288	13.788
16-bit KSA	21.264	10.893	10.371

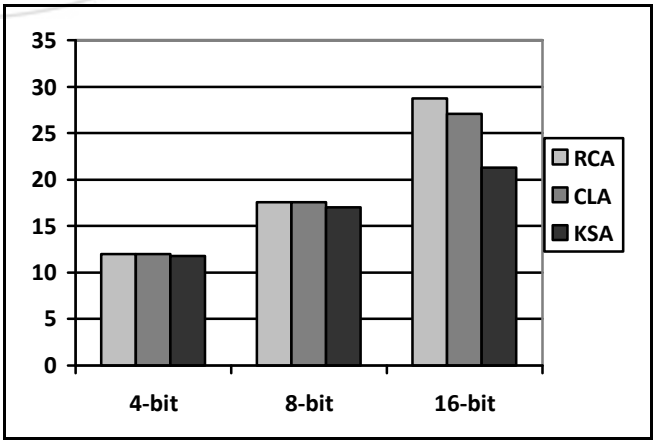

Figure 8: Graphical Representation of the Maximum Combinational Path Delay (in ns) of the Adders

From the delay comparison of adders, it is clear that KSA, a representative of parallel prefix adders, is the fastest. The delay performance of CLA is at par RCA for 4-bit and 8-bit operands but as the number of bits increases, the parallelism of CLA proves to be advantageous and its delay reduces when compared to RCA. For 4-bit and 8-bit adders, the logic delay is same for all the adders but KSA is offering less delay due to improved routing, whereas for 16-bit operands, KSA is advantageous in logic as well as routing.

Table 2 describes the area utilization of each design in terms of slices and LUTs, and also the IOBs used by the adder has been given. The area comparison in terms of number of slices has been plotted in figure 9. KSA was the best when considered for delay but for area, as the number of bits of operands increases KSA occupies more area due to increase in parallel prefix stages.

Table 2: Comparison of Area of Adders

Type of Adder	Area		
	No. of Slices used (out of 3584)	No. of 4-input LUTs used (out of 7168)	No. of Bonded IOBs used (out of 141)
4-bit RCA	4	8	14
4-bit CLA	4	8	14
4-bit KSA	4	8	14
8-bit RCA	9	16	26
8-bit CLA	9	16	26
8-bit KSA	9	16	26
16-bit RCA	18	32	50
16-bit CLA	18	32	50
16-bit KSA	37	64	50

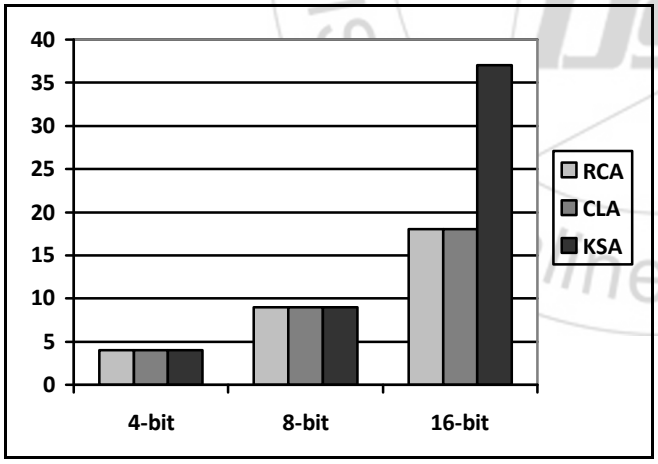

Figure 9: Graphical Representation of the Number of Slices Utilized by the Adders

It can be seen that for 4-bit and 8-bit operands, the area of all the three adders are equal, but for 16-bit value KSA shows tremendous increase in area. Hence, from area point of view, RCA and CLA are better than KSA.

5. Conclusion

The above discussion can be summarized as; KSA has the best delay performance whereas RCA and CLA offer a better area profile.

From the comparison of delay and area of various adders for different number of bits, we can derive a conclusion that speed and area cannot be optimized at the same time. If one parameter is improved, the other definitely shows degradation.

References

[1] A. Anand Kumar, "Fundamentals of Digital Circuits", 2nd Edition, PHI Learning Pvt. Ltd., 2009.
[2] Sudheer Kumar Yezerla, and B Rajendra Naik, "Design and Estimation of delay, power and area for Parallel prefix adders", IEEE Proceedings of 2014 RAECS, UIET, Panjab University, Chandigarh, pp. 1-6, March 2014.
[3] Jan M. Rabaey, Anantha Chandrakasan, and Borivoje Nikolic, "Digital Integrated Circuits: A Design Perspective", 2nd Edition, PHI, 2005.
[4] Neil H. E. Weste, and David Harris, "CMOS VLSI Design: A Circuits and Systems Perspective", 3rd Edition, Pearson Education, 2005.
[5] Kartheek Boddireddy, Boya Pradeep Kumar, and Chandra Sekhar Paidimarry, "Design and Implementation of Area and Delay Optimized Carry Tree Adders using FPGA", International Conference on Computers and Communications Technologies, IEEE, pp. 1-6, Dec. 2014.
[6] Geeta Rani, and Sachin Kumar, "Delay Analysis of Parallel-Prefix Adders", International Journal of Science and Research (IJSR), Vol. 3, Issue 6, pp. 2339-2342, June 2014.
[7] David H. K. Hoe, Chris Martinez, and Sri Jyothsna Vundavalli, "Design and Characterization of Parallel Prefix Adders using FPGAs", 43rd Southeastern Symposium on System Theory, IEEE, pp. 168-172, 2011.
[8] David Harris, "A Taxonomy of Parallel Prefix Adders", 37th Asilomer Conference on Signals, Systems and Computers, Vol. 2, pp. 2213-2217, 2003.
[9] Peter M. Kogge, and Harold S. Stone, "A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations", IEEE Transactions on Computers, Vol. C-22, No. 8, pp. 786-793, August 1973.

Author Profile

Khushboo Bais has done her B.E. in Electronics & Telecom. Engg., in 2012, from DIMAT, Raipur and is currently pursuing M.Tech. in VLSI & Embedded System Design from DIMAT, Raipur.

Zoonubiya Ali has done her B.E. in Electronics & Telecom. Engg., M.Tech. in Electronics Engg., and is currently pursuing Ph.D. from Nagpur University. She holds the designation of Associate Professor at DIMAT, Raipur, and has 18 years of teaching experience.