Design and Implementation of Low Delay Look Ahead Adder and Ripple Adder

Samrat Kumar Singh[1] Deepak Kumar[2]
[1]M. Tech. Scholar [2]Assistant Professor
[1,2]Department of Electronics & Communication Engineering
[1,2]VIST Bhopal India

Abstract— In recent years, a lot of attentions have been attracted by the reversible logic due to the characteristic of zero energy dissipation. In this paper, the author proposed a 16 bit carry look-ahead adder is constructed by four 4 digits groups based on the theory of reversible logic, which has the advantages of theoretical zero power dissipation and high efficiency. This paper focuses on the implementation 16 and bits of highly optimized area efficient Ripple carry adder (RCA) and Carry look ahead (CLA) adders. Ultimately, we can establish that the Carry look ahead adders are so greatest among all the formerly active designs. All these processes will be Simulated & Synthesized on the ISE Xilinx 14.7 software.

Key words: Adder, Ripple Carry Adder, Carry Look-Ahead Adder, Carry Select Adder VHDL Code

I. Introduction

With the increasing importance of energy dissipation in integrated circuits, the reversible logic implementations gain in prominence as a way to reduce power since irreversible computing is one of the most significant factors to generate energy dissipation. By the Landauer's principle, each bit of information lost will generate kl1n2 joules of heat energy, where T stands for the absolute temperature at which computation is performed and k is Boltzmann's constant[1][8] so how to avoid information lost is an efficient way to decline the energy dissipation in digital integrated circuits. High-speed adder is the Necessary component in a data path e.g. Microprocessors and a Digital signal Processor. For adding two binary numbers, there are several adder Structures based on different Design ideas. There are many binary adder architecture ideas to be implemented in such applications. The easiest type of parallel adder to build is a ripple carry adder, which uses a Chain of one bit full adder to generate its output. The Ripple Carry Adder (RCA) gives the most compact design but Takes longer computation time. The time critical Applications use Carry Look-ahead scheme (CLA) to derive Fast results but lead to increase In area. In mobile electronics, falling area and influence utilization are Solution factors in growing portability and sequence life Even in servers and desktop Computers, power an important design constraint [1]

II. Ripple Adder

Ripple carry adder A ripple carry adder is a digital circuit that produces the arithmetic sum of two binary numbers.[4] may be capable of constructed with full adders connected in cascaded through the carry output beginning every full adder associated to the carry input of the next full adder in the chain the interconnection of four full adder (FA) circuits to provide a 4-bit ripple carry adder observe from so as to the input is beginning the accurate side because the first cell traditionally represents the least significant bit (LSB). Bits and in the figure represent the least significant bits of the numbers to be added. The sum output is represented by the bits Show in fig. 1Below [10]

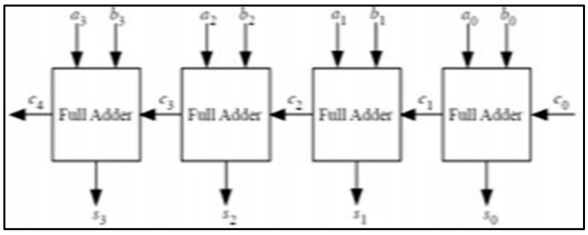

Fig. 1: Full adder

III. Carry look ahead adders (CLA)

The Enter to speed up adding up is influential the carry in to the soaring order bits sooner. There is assortment of scheme to expect the carry so that the nastiest case scenario is a function of the log2 of the number of bits in the adder. These preventative signals are quicker since they go throughout fewer gates in succession, but it takes many more gates to anticipate the proper carry a key to understanding fast-carry schemes is to remember that, unlike software, hardware executes in parallel whenever Inputs change Look-Ahead Adder a Carry Look-Ahead adder (CLA) is a type of adder used to improve speed by reducing the amount of time required to determine carry bits. It can be contrasted through the simpler, but frequently slower, ripple carry adder intended for which the carry small piece is designed alongside the sum bit, and each bit must wait until the previous carry has been calculated to begin calculating its own result and carry bits. Show in fig.2.

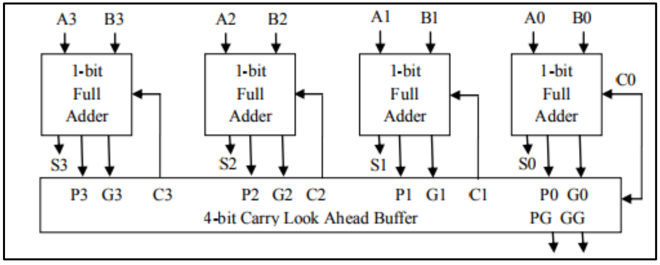

Fig. 2: Carry Look-Ahead Adder

The Carry Look-Ahead Adder calculates one or additional carry bits before the sum, which condense the wait time to estimate the result of the larger value bits. Consider a full adder circuit as shown in Figure 2

$$Pi = Ai\ Bi \quad (1)$$

$$Gi = Ai\ Bi \quad (2)$$

Where, Gi is carry generate and Pi is carry propagate. The output sum and carry can be expressed as

$$Si = Pi\ Ci\ (3)\ Ci{+}1 = Gi + Pi\ Ci \quad (3)$$

Gi produce on carry at what time both Ai and Bi are one, despite of the input carry. Pi is connected with the propagation of the carry from Ci to Ci+1. Now the Boolean function for the carry output of each stage can be written as follows.

$$C2 = G1 + P1\,C1 \quad (4)$$
$$C3 = G2 + P2\,C2 \quad (5)$$
$$C4 = G3 + P3\,C3 \quad (6)$$

From the above Boolean function it can be seen that C4 does not have to wait for C3 and C2 to propagate; in fact C4 is propagated at the same time as C2 and C3. Using a look ahead carry generator we can easily construct a 4-bit parallel adder with a look ahead carry scheme which has been implemented using CMOS and CEPAL logic styles.

$$C1 = G0 + P0 \cdot C0 \quad C2 = G1 + P1 \cdot C1 \quad C3 = G2 + P2 \cdot C2$$
$$C4 = G3 + P3 \cdot C3$$

Now substitute C1 into C2, C2 into C3, etc. we get:

$$C1 = G0 + P0 \cdot C0$$
$$C2 = G1 + P1 \cdot (G0 + P0 \cdot C0)$$
$$C3 = G2 + P2 \cdot (G1 + P1 \cdot (G0 + P0 \cdot C0))$$
$$C4 = G3 + P3 \cdot (G2 + P2 \cdot (G1 + P1 \cdot (G0 + P0 \cdot C0)))$$

IV. Carry Select Adder

A Carry Select Adder is a scrupulous technique to implement an adder, which is a logic constituent that computes the (n+1) bit sum of two n-bit numbers. The carry-select adder is Straightforward other than rather fast. The carry-select adder normally consists of two ripple carry adders and a multiplexer. addition jointly two n-bit numbers from side to surface a carry-select adder is completed through two adders (consequently two ripple carry adders) in command to execute the calculation twice, one time with the statement of the carry being zero and the previous high and mighty one. [9] later than the two results are considered, the accurate sum, as well as the correct carry, is then selected by means of the multiplexer once the correct carry is recognized. The structure of a 16 bit CSLA is exposed below:

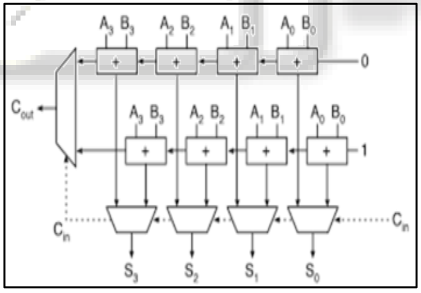

Fig. 3: The structure of a 16 bit CSLA

V. Proposed Designs

A. Ripple Carry Adder

To conniving the Ripple carry adder, the necessary element is full adder. So firstly we can design a made to order reversible full adder intended for getting the optimized results with using simple 3*3 Peres gate [10] the RTL exposed in the fig 5. It is proved that the modified reversible full adder design can exist realized by means of two garbage outputs and simply on ancillary input. While we are conniving the full adder, the 3rd input of the first Peres gate should be considered as zero. The output of the ripple carry adder is given away below.

$$Sum = A \oplus B \oplus C$$
$$Carry = (A \oplus B).\ Cin \oplus AB$$

RCA requires n-bit full adder intend circuits; ripple carry adder propagates their character carry input through each and every full adder circuit blocks. The output carry of the ith full adder track is connected to (i+1) th full adder design circuit. Therefore the coming subsequently full adder circuit has to remain until the preceding logic block to give the carry for meticulous stage. Ultimately, it will supply the sum and carry subsequently the n-stages for n-bit RCA addition. The output of the original Peres gate is applied to the inputs of the second Peres gates correspondingly hence the sum and carry are generating at the concluding stages of the ripple carry design. This paper proposed the structural design for 16-bits of ripple carry adder only, but we can synthesize, simulate In this way it is probable to design any bit (n^{th}) of Ripple carry addition also. The simulation result of 16-bit ripple carry adder is shown in fig .4

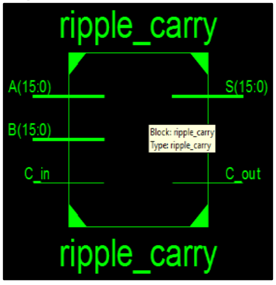

Fig. 4: RTL view of 16-bit ripple carry adder

Ripple-carry Optimized Adder meaningful that the worst case delay of ripple carry adder is the delay among carry in and carry out of each full adder, I investigate another optimized design of full adder to see if it will get a better presentation. The top level of the ripple carry adder is the same, though the full adder schematic is different, as Fig.8. The intuition of this full adder optimization is to minimize the delay of carry in and carry out of the full adder which is the answer direct of the ripple carry adder.

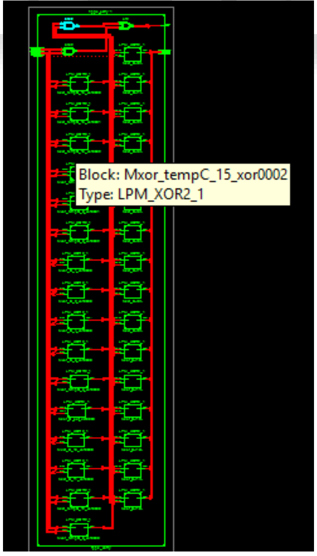

Fig. 5: RTL view of 16-bit ripple carry adder

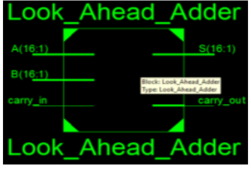

Fig. 6: RTL view of 16-bit carry look ahead adder

Carry-look ahead Adder to diminish the working out time, a new way to apply the 4-bit adder is to use carry-look ahead component to parallel produce the carry for each bit addition. The top-level schematic of CLA is as Fig.6.

It basically generates indication bits for carry propagation and generation (P and G) of each full adder and calculates all the carries simultaneously. The schematics of propagation full adder and carry look ahead unit are as Fig.6 and Fig.7. Carry look ahead logic uses the concepts of generating and propagating carries.

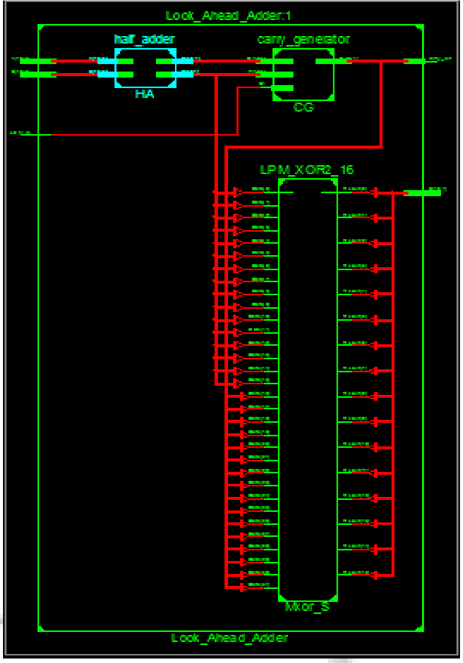

Fig. 7: RTL view of 16-bit carry look ahead adder

Carry-look ahead adder (CLA) or fast adder is a type of adder used in digital logic. A carry-look ahead adder improves speed by reducing the amount of time required to determine carry bits. In ripple carry adders, the carry must propagate through the gate levels before the sum is obtainable at the output terminals. An n-bit ripple carry adder will have 2n gate levels. The propagation time can be a preventive factor on the speed of the circuit. One of the mainly popular methods to reduce delay is to employ a carry look ahead mechanism.

VI. Simulation and Synthesis Results

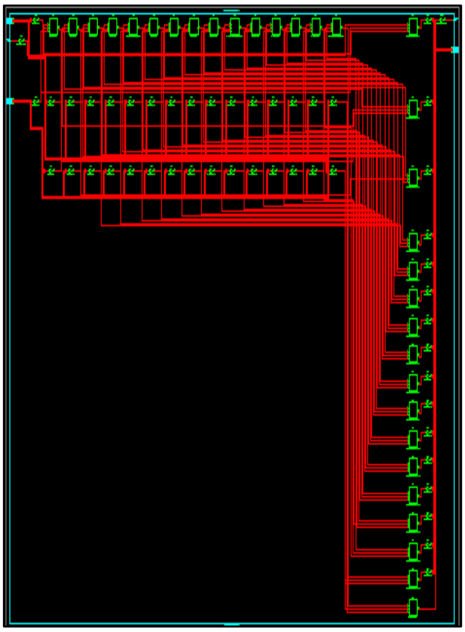

Fig. 8: Technological schematic of adder design

We perform the simulation and synthesis and précis the results of all the adders. The Functional verification (simulation) and synthesis (high level description is converted into RTL) of all the adders is performed and Results are summarized. Build a 16-bit carry look results are summarized. Build a 16-bit carry look ahead adder Design and implement the circuit (describe how carry look ahead works and how you derive the final circuit layout) Build and simulate the circuit using Xilinx. Show exactly 3 different simulation cases and total REAL time to XST completion: 21.00 secs total CPU time to XST completion: 21.27 secs Number of Slices: 18 out of 5472, Number of 4 input LUTs: 32 out of 10944, Number of IOs: 50 Number of bonded IOBs: 50 out of 240.

- Design Statistics: IOs: 50
- Cell Usage:
- BELS: 32
- LUT3: 32
- IO Buffers: 50
- IBUF: 33
- OBUF: 17

Every part of the single ended outputs in this intend are with slew rate limited productivity drivers. The delay on speed critical single ended outputs can be dramatically reduced by designating them as fast outputs. Analyzing hierarchy for entity Look Ahead Adder.

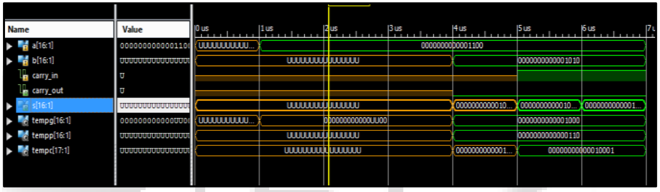

Fig. 9: Simulation result of adder design

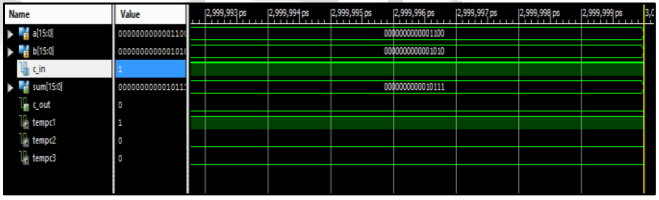

Fig. 10: Simulation result of adder design

Following the watching of simulation waveforms, synthesis is performed for calculation of impediment and area and thereby the speed and power of the CLA's are calculated and a comparison of regular, modified and improved adder is made in terms of delay, area and power and listed in the below table. The evaluation of all the three types of Carry Select Adders is completed in conditions of delay, beginning the beyond comparison results listed in the table, we can say that the delay is compact.

Parameter		Ripple adder	Look ahead adder
Proposed Logic	Delay at 16 bit	18.140ns	17.210ns
Design [1]	Delay at 16 bit	24.022 ns	21.303 ns

Table 1: Show Comparison Delay of the design

VII. Conclusion

All the three models of ripple adder, look ahead adder and carry select adder are designed and are implemented in VHDL using Xilinx 14.7 ISE tool and the results are compared in terms of delay. The performance of this three adder in terms of delay is evaluated by implementing logic by using the look ahead adder in the adder part. The proposed design is optimized in terms of delay and hardware complexity. Firstly, the gate used here, in carry look ahead adder are parity preserving gate, hence the whole adder preserves the parity. Therefore, no intermediate checking will be required if there is no fault detected Total REAL time to XST completion: 21.00 second Total CPU time to XST completion: 21.27 second, Number of 4 input LUTs 32 10, 944. Number of occupied Slices 24 5,472, Number of Slices containing only related logic 24 used out of 24, In the world of low power VLSI design everything is possible, might be in the future the full adder design is more precise than all the previous existing designs.

References

[1] Reversible Adder Design For Ripple carry and Carry look ahead (4, 8, 16, 32-bit) –International Conference on Computing, Communication and Automation (ICCCA2016) - NeelamSomani. Chitrita Chaudhary, Sharad Yadav
[2] J. L. Hennessey and D. A. Patterson, Computer Architecture. A Quantitative Approach. San Francisco: Morgan Kaufmann, 1990.
[3] M. A. Franklin and T. Pan, "Performance comparison of asynchronous adders," in Proc. Int. Symp. Advanced Research in Asynchronous Circuits and Systems, 1994, pp. 117-125.
[4] F.-C. Cheng, S.H. Unger and M. Theobald, "Self-timed carry-lookahead adders," IEEE Trans. Comp., vol. 49, pp. 659-672, 2000.
[5] M. M. Mano and C. R. Kime, Logic and Computer Design Fundamentals. New Jersey: Prentice Hall, 2001.
[6] I. Flores, The Logic of Computer Arithmetic. New Jersey: Prentice Hall, 1963.
[7] A. De Gloria and M. Olivieri, "Statistical carry lookahead adders," IEEE Trans. Comp., vol. 45, pp. 340-347, 1996.
[8] A. Kaizerman, S. Fisher and A. Fish, "Dual Mode Logic," Provisional USPatent application no. 61/514,497.
[9] J. M. Rabaey, A. P. Chandrakasan and B. Nikolic, "Digital integrated circuits: a design perspective," Upper Saddle River, N.J.: Pearson Education, 2003.
[10] B. H. Calhoun and A. P. Chandrakasan, "Ultra-dynamic Voltage scaling (UDVS) using sub-threshold operation and local Voltage dithering,", IEEE J. Solid-St. Circ., vol. 41, pp. 238-245, 2006.
[11] M.-E. Hwang and K. Roy, "ABRM: Adaptive-Ratio Modulation for Process-Tolerant Ultradynamic Voltage Scaling," IEEE Trans. VLSI Syst., vol. 18, pp. 281-290, 2010.