International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Simulation of Different bit Carry-Skip Adder in Verilog

Sangeeta Rani[1], Sachin Kumar[2]

[1]M. Tech Student, Department of Electronics & Communication,
Meri College of Engineering & Technology, Sampla, Haryana, India

[2]Faculty, Department of Electronics & Communication,
Meri College of Engineering & Technology, Sampla, Haryana, India

Abstract: ***Adders are the most basic and essential component used in Digital signal processing and is widely used in the digital integrated circuits. As there are various adder structures which provide the increased operational speed in the arithmetic circuits but in terms of area or delay there is a loose connection (area or delay of the adder circuit design is more as compared to the other structures discovered). With the advances in technology, researchers have tried and are trying to design adders which offer either high speed, low power consumption, less area or the combination of them. The addition of the two bits is very based on the various speed-up schemes for binary addition, a comprehensive overview and different bits carry skip adder structures is given in this paper. We will synthesize the Carry skip adder of bits – 4 Bit, 8 Bit, 16 Bit and 32 Bit in ISE XIILINX 10.1 by using HDL - Verilog and will simulate them in Modelsim 6.4a. Also Delay, Slices Used and Look up tables used by the Different bit Carry skip adder structure is given.***

Keywords: Carry skip adder, Ripple Carry Adder, Carry Look Ahead adder, Carry Save adder

1. Introduction

Adders are the building block in DSP applications. Adders perform operations like Addition, Subtraction, Multiplication and Division. Chen et. Al found that addition is the most frequently operation in the Digital signal processing [1]. The key component in DSP is the binary adder and the basic unit on which every binary adder structure is based, is the Full adder cell. Full adder cell contains the three inputs A, B and C_{in} and two outputs - sum **S** and Carry $\mathbf{C_{out}}$. By cascading the full adder cells, we have the basic Adder structure called as "Ripple carry adder" in which "n" full adder cells are cascaded to get the n - bit Ripple carry adder and carry generated at n^{th} bit is given as the input to the n+1 bit in addition to the A and B inputs. As Ripple carry adder is the most basic one among the other adders but also slowest of them all, because of the propagation of the carry from LSB to MSB.

Adder Structures other than ripple carry adder are - Carry Look Ahead Adder, Carry Select Adder, Carry Save Adder, Carry Skip Adder etc. Ripple carry adder is slowest among all the adder structure because of the rippling of the carry. Carry look Ahead adder solves the problem of carry propagation from right to left by using carry propagate and the carry generate signals for reducing delay but the area, number of gates is increased and thus complexity of the system is increased. In Carry skip adder, the propagation time of carry is reduced by skipping over the group of adder stages and presents hardware and performance compromise.

In this paper, Simulation of the Different bit Carry Skip Adders is presented. The Carry Skip Adder presented in this paper is modeled by using Verilog for 4 bit, 8 bit and 16-bit and 32 bit data. Also ISE XILINX v10.1 is used as synthesis tool. Modelsim 6.2a is used to get timing simulation.

2. Ripple Carry Adder

Ripple carry addition was used in the first electronic computers. Ripple carry adder uses the O (n) area and O (n) Delay where n is the width of operand [2]. In Ripple carry adders carry propagates from LSB to MSB. And as carry ripples from all the way down to the MSB, delay is introduced and is increased as the number of bits for the addition is increased. Hence Ripple carry adder can be used for the small bit addition.

Addition in ripple carry adder is just like the addition we do manually and carry propagates from the least significant bit to the most significant bit through the carry jump from n^{th} stage to n+1 stage.

There also exists Probabilistic Ripple carry adder which calculates that a bias voltage scaling (BIVOS) technique can yield much better energy saving than simply supplying all FAs with the same voltage.

3. Carry Look-Ahead Adder

Carry look ahead addition was introduced by Weinberger & Smith in 1956 [3]. Concept of the Carry look ahead addition is to use the Carry generate and Carry propagate signal for the propagation of the carry as fast as possible from LSB to MSB.

$G_i = A_i \cdot B_i$ called the generate function
$P_i = A_i \oplus B_i$ called the propagate function

And the outputs are

$$\mathbf{S} = A \oplus B \oplus C = P_i \oplus C_i \quad \ldots\ldots\ldots (1)$$

$$\mathbf{C_{n+1}} = G_n + G_{n-1}.P_n + G_{n-2}.P_n.P_{n-1} + \ldots\ldots$$
$$\ldots + G_0.P_nP_{n-1}\ldots P_1 + P_nP_{n-1}\ldots P_1P_0C_{in} \quad \ldots\ldots\ldots (2)$$

Propagation delay exerted by the carry look ahead mechanism is less then than the ripple carry addition. In Carry look-ahead mechanism, the propagation delay is reduced to four-gate level irrespective of the number of bits in the adder. [4]

4. Carry Select Adder (CSA)

In the ripple carry adder, addition at i^{th} bit stage cannot be completed until the $i\text{-}1^{th}$ stage carry does not arrive at the i^{th} stage, this problem can be overcome by employing Carry select adder. In Carry select adder, for any bit position it receives a C_{in} signal from the previous bit position which can be true or false. In CSA two separate additions occur one assuming the C_{in} signal is true and other assuming it is false. Carry select adder is not area efficient as it uses multiple pairs of ripple carry adders. The splitting of the carry chain can be done multiple times, breaking the computation into several pairs of short adders with output muxes choosing which adder's output to select [5]. In the Carry select adder, fixed or Variable block structure can be used [6].

5. Carry Save Adders (CSaA)

Carry Save addition was discovered to add the decimal numbers. But it is also applicable to the Binary number system. Carry save addition is used when we want to add more than 2 numbers. Using carry save addition, the idea is to take 3 numbers that we want to add together, x + y + z, and convert it into 2 numbers c + s such that x + y + z = c + s. In carry save addition, we refrain from directly passing on the carry information until the very last step. The carry save approach breaks this process down into two steps. The first is to compute the sum ignoring any carries: The final addition is then computed as:

Shifting the carry sequence C left by one place.
Placing a 0 to the front (MSB) of the partial sum sequence S.
Finally, a ripple carry adder is used to add these two together and computing the resulting sum. [7]

6. Overview of Carry Skip Adder

In the carry skip adder to speed-up operation, carry propagation is skipped to position i without waiting for rippling. A carry-skip adder reduces the carry-propagation time by skipping over groups of consecutive adder stages. The carry-skip adder is usually comparable in speed to the carry look-ahead technique, but it requires less chip area and consumes less power. To implement carry-skip adder, stages are divided into r–bit blocks of simple carry scheme. Carry-skip logic is added to each block to detect when carry-in the block can be passed directly to the next block. In each block, a ripple carry adder is utilized to produce the sum and carry out bit for each block. Every block generates a block propagate and block generate signal. Also for the given column, CSKA uses the carry out equation in terms of the carry in signal.

$$C_{k+1} = G_k + P_k . C_k \quad \ldots\ldots (3)$$

Also Generate and Propagate signals used by the carry skip adder are:

$$G_k = A_k . B_k \quad \ldots\ldots (4)$$
$$P_k = A_k \oplus B_k \quad \ldots\ldots (5)$$

From this equation, it can be seen that setting the carry-in signal of a block to zero causes the carry out to serve as a block generate signal. Therefore, an *r* bit *AND* gate is also used to form the block propagate signal. The block generates and block propagates signals produce the input carry to the next block [8].

Fig. 1 shows the 8 Bit Carry Skip Adder using 2 Bit Block of RCA (Ripple Carry Adder) and Fig. 2 shows the n Bit Carry Skip Adder formed by 4 Bit block of RCA.

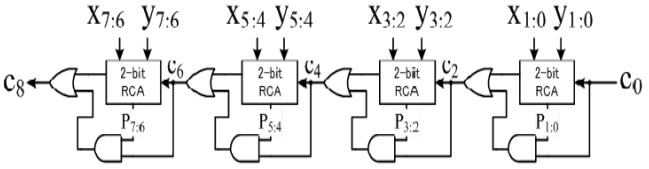

Figure 1: Formation of 8 Bit Carry Skip Adder using 2 Bit Block of RCA

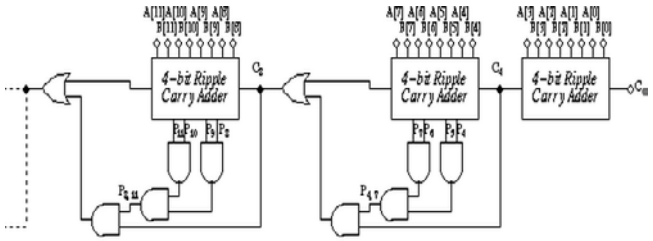

Figure 2: Formation of n Bit Carry Skip Adder using 4 Bit Block of RCA

In other words, each block tries to detect if a carry is going to bypass the entire smaller CPA block. For example, to obtain the carry into bit position 8, the following equation is utilized.

$$C_8 = g_{4:7} + p_{4:7} \cdot c_4 \quad \ldots\ldots\ldots (6)$$

Where

$$P_{4:7} = p_7 \cdot p_6 \cdot p_5 \cdot p_4 \quad \ldots\ldots\ldots (7)$$

7. Simulation Results

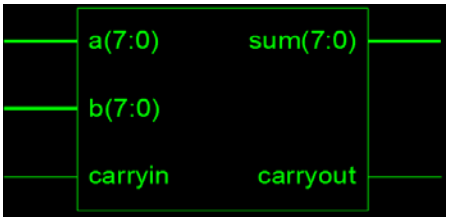

Figure 3: Basic Block Diagram of 8 Bit Carry skip adder

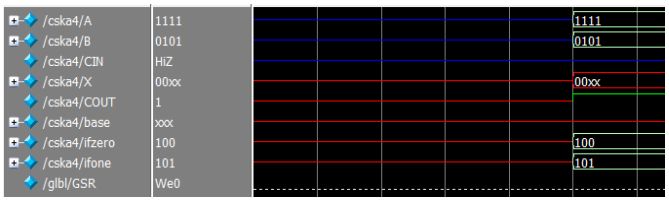

Figure 4: Addition using 4 Bit Carry skip adder

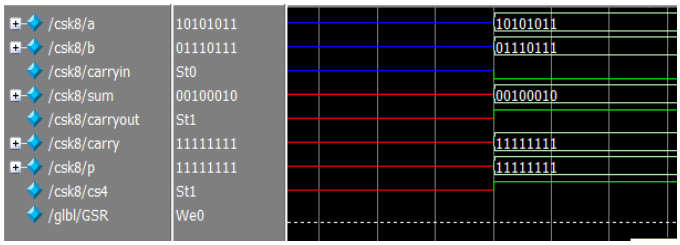

Figure 5: Addition using 8 Bit Carry Skip adder

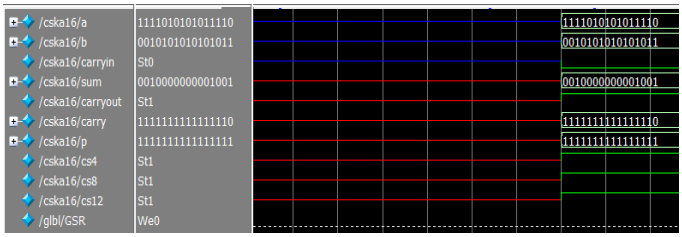

Figure 6: Addition using 16 Bit Carry Skip adder

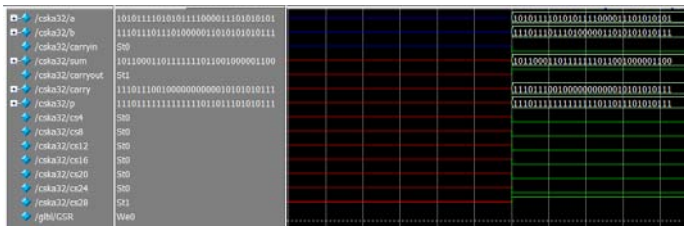

Figure 7: Addition using 32 Bit Carry Skip adder

Table 1: Device Utilization Summary of 4 - Bit Carry Skip Adder

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	5	960	0%
Number of 4 input LUTs	10	1920	0%
Number of bonded IOBs	14	66	21%

Maximum combinational path delay: 8.736ns

Table 2: Device Utilization Summary of 8 - Bit Carry Skip Adder

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	9	960	0%
Number of 4 input LUTs	15	1920	0%
Number of bonded IOBs	26	66	39%

Maximum combinational path delay: 11.338ns

Table 3: Device Utilization Summary of 16 – Bit Carrys Skip Adder

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	14	960	1%
Number of 4 input LUTs	25	1920	1%
Number of bonded IOBs	50	66	75%

Maximum combinational path delay: 14.170 ns

Table 4: Device Utilization Summary Of 32 – Bit Carrys Skip Adder

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	26	960	2%
Number of 4 input LUTs	45	1920	2%
Number of bonded IOBs	98	66	148%

Maximum combinational path delay: 29.834 ns

Table 5: Device Utilization Summary Of 4, 8, 16, 32 Bit Carrys Skip Adder

Logic Utilization	4 – Bit CSkA	8 - Bit CSkA	16 – Bit CSkA	32 – Bit CSkA
Number of Slices	5	9	14	26
Number of 4 input LUTs	10	15	25	45
Number of Bonded IOBs	14	26	50	98
Delay	8.736 ns	11.338 ns	14.170 ns	29.834 ns

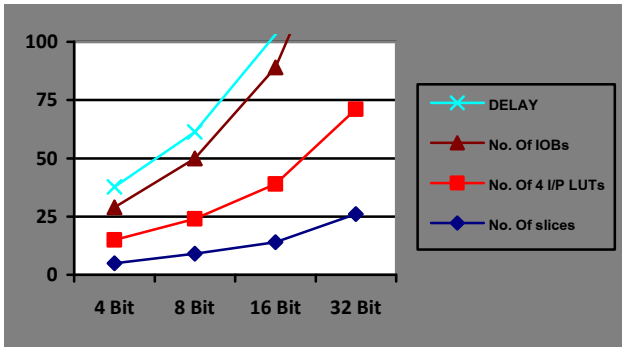

Figure 8: Graphical representation of Device Utilization summary of 4,8,16 and 32 bit Carry Skip Adder

8. Conclusion

Table IV shows the Number of slices used, Number of 4 input LUTs and Maximum Combinational Delay of 4, 8, 16 and 32 bit Carry Skip adders using XILINX 10.1 and Modelsim 6.4a with HDL - Verilog. Carry skip adder is compromise between Ripple carry adder and Carry look ahead adder (This statement is proved when we compare 16 Bit Carry look ahead adder and 16 Bit CSkA in terms of delay, then CLA has delay of 26.686 ns and CSkA has delay of 14.170 ns. No. of slices and number of look up tables used by CLA are 18, 32 respectively. Thus in every aspect CSKA is beneficial over CLA). Carry skip adder uses the ripple carry adder block, generate and propagate signals (and of course the carry skip logic). Because of its great topological regularity and layout simplicity Carry skip adder is considered a good compromise in terms of area and performance.

9. Future Scope

This paper consists of various bit Carry skip adders. Carry skip adders are useful when we needs our addition to be done in less time (Delay efficient) and less area consuming and it can be done by using variable block carry skip adder – by choosing less size block at the ends and the more size block in the middle delay can be efficiently reduced also the

because of the area efficiency it can be used in various Digital integrated circuits.

References

[1] D. C. Chen, L. M. Guerra, E. H. Ng, M. Potkonjak, D. P. Schultz, and J. M. Rabaey, "An integrated system for rapid prototyping of high performance algorithm specific data paths," in Proc. Application Specific Array Processors, Aug 1992, pp. 134-148.

[2] Kenny Johansson, Oscar Gustafson, and LarsWanhammar, "Power Estimation for Ripple-Carry Adders with Correlated Input Data", E. Macii et al. (Eds.): PATMOS 2004, LNCS 3254, pp. 662–674, 2004. c Springer-Verlag Berlin Heidelberg 200.

[3] Weinberger, J. L. Smith, "A logic for high-speed addition", National bureau of standards circular, 1958, pp. 3-12.

[4] May Phyo Thwal, Khin Htay Kyi, and Kyaw Swar Soe, "Implementation of Adder-Subtracter Design with Verilog HDL" World Academy of Science, Engineering and Technology Vol. 2, March 2008.

[5] Scott Hauck, *Member, IEEE*, Matthew M. Hosler, and Thomas W. Fry, "High-Performance Carry Chains for FPGA's", IEEE transactions on very large scale integration (VLSI) systems, Vol. 8, No. 2, April 2000, pp. 138 – 147.

[6] V. G. Oklobdzija and E. R. Barnes, "On implementing addition in VLSI technology," J. Parallel Distrib. Comput. vol. 5, no. 6, Dec. 1988, pp. 716–728.

[7] Raminder Preet pal singh, Parveen Kumar, Balwinder Singh, "Performance analysis of 32- bit array Multiplier with a Carry Save adder and with a Carry Look Ahead adder", International journal of recent trends in Engineering, Vol. 2, No. 6, November 2009.

[8] James E. Stine, "Digital Computer Arithmetic Data path Design Using Verilog HDL", Kluwer academic publishers, 2004.

[9] Silvio Turrini, "Optimal Group Distribution in Carry-Skip Adders", Western Research Laboratory 1989.

Author Profile

Sangeeta Rani received her B. Tech degree in Electronics and Communication Engineering from M.R.I.E.M College under Maharishi Dayanand University, Rohtak. Currently pursuing M. Tech from M.E.R.I. college under Maharishi Dayanand University.