http://www.jctjournals.com

ISSN(ONLINE):2278 - 3814

Volume-12, Issue-07, July 2023

JOURNAL OF COMPUTING TECHNOLOGIES (JCT)

International Journal

Page Number: 01-05

Designing of 32 bit Floating Point Carry Look Ahead Adder Using VHDL:A Review

[1]Ajay Kumar, [2]Prof. Nishi Pandey, [3]Prof. Abhishek Agwekar

[1]M. Tech Scholar, [2]Assistant Professor, [3]Head of department

[1,2,3] TIEIT, Bhopal,

[1]ajaylion33@gmail.com, [2]nishi.pandey@trubainstitute.ac.in, [3]abhiagwekar@gmail.com

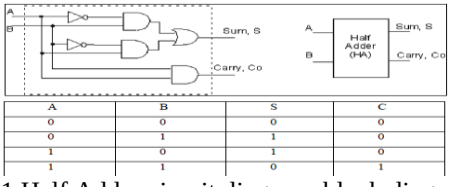
***Abstract*— Considering that the synchronous architectures requires that all clock events happen at the same time over the complete circuit which it not possible due to clock skew also the latency and throughput of the circuit are directly linked to the worst-case delay of the slowest element which increases the delay. Hence our work presents self-timed carry look ahead adder (CLA) based implementation of IEEE 754 32-bit floating point multiplier for FPGA devices. We have done comprehensive simulation testing on FPGA Spartan 3 device through Xillinx 14.4 simulation tool. The simulation results show that the proposed design has lower latency than synchronous design. Also, comparatively power dissipation in our proposed design is low with a synchronous design.**

***Keywords*— FPGA, Xillinx 14.4 simulation, IEEE 754, carry look ahead adder (CLA)**

I. INTRODUCTION

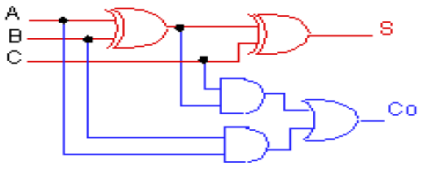
A large number of computer applications(like Computer Graphics, Control Systems, Modeling System, Simulators etc.) neededfloating point arithmetic. However, most of the presently available methods are slow and inefficient because of sequential design, however the recent development in the field of programmable logic devices such as FPLA and CPLD opens the new area of parallel and high speed floating point designs. As per IEEE754 two formats were described as a single and double precision floating point. These arithmetic's were used in different applications in the fields of robotics and automations. Since power consumption is an important consideration in any digital system design. A digital system may consists of several entities such as adder, subtractor, multiplier. However the most basic building block of any digital system is adder. It was observed that the adders used in digital system consume a significant amount of power during computation, and also a significant amount of delay is also attributed. Thus, in order to overcome the above stated problem an efficient adder in terms of power and delay is required. Existing System and Increasingly huge data sets and the need for instant response require the adder to be large and fast. The traditional ripple-carry adder (RCA) is therefore no longer suitable for large adders because of its low-speed performance.

Many different types of fast adders, such as the carry-skip adder (CSK), carry-select adder (CSL), and carry-look-ahead adder (CLA), have been developed. Also, there are many low-power adder design techniques that have been proposed. However, there are always trade-offs between speed and power. Building a low power VLSI system has emerged as significant performance goal because of the fast technology in mobile communication and computation. The advances in battery technology have not taken place as fast as the advances in electronic devices. So the designers are faced with more constraint; high speed, high throughput and at the same time, consuming as minimal power as possible.The goal is to extend battery life span of portable electronics is to reduce the energy expended per arithmetic operation. To execute an arithmetic operation, a circuit can consume very low power by clocking at extremely low frequency but it may take a very long time to complete the operation.



A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Figure 1-1 Half Adder circuit diagram, block diagram and truth table



ISSN(ONLINE):2278 - 3814

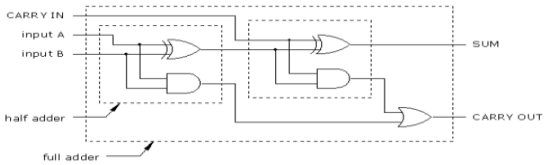


Figure 1-2 Full Adder circuit diagram, combination diagram of 02 Half Adder

The Basic Adder was performing arithmetic addition of two binary bits, generally we prefers a combinational circuit of Half Adders, that combines to form through AND-OR gates. For Full adders we can use two half adder, that will add three bits concurrently and generate sum and carry as output.

II. LITERATURE REVIEW

Various researches have been carried out in order to design adders and a proposed design of 32 bit multiplier using carry look ahead. Carry Look ahead Adder (CLA) is one of the fastest adder structures that is widely used in the processing circuits.There are two types of circuits, named as synchronous andan asynchronous. The synchronous circuits have a clock signal to synchronize the operations of subsystems, while an asynchronous circuit does not have a clock signal, but its operating states, these states may or may not be output of the cascaded sub-system.

[1] P. Balasubramanian, D. Dhivyaa, J. Jayakirthika, P. Kaviyarasi and K. Prasad, "Low power self-timed carry lookahead adders," in Circuits and Systems (MWSCAS), 2013 IEEE 56th International Midwest Symposium , Columbus, OH, 4-7 Aug. 2013. In this paper authors focused on semi-custom design (direct synthesis) of self-timed CLA adders which are physically implemented using standard cells, with an eye on timing optimization. In this context, authos deals with the design of self-timed CLA adders based on the notion of section-carry, where intra-section carries are allowed to ripple within an adder group, while inter-section carries are generated via lookahead. Moreover authors discuss on efficient implementations of 16-bit full adders on FPGA devices using specialized carry-logic. In this paper author focus on the implementation of 16-bit full adder based on Very High Speed Integrated Circuit (VHSIC) Hardware Description Language. And concluded with the design of Hardware Resources mapped on SPARTAN-3 FPGA, implement, simulate and synthesized using VHDL.

[2] PreethiSudhaGollamudi, M. Kamaraju," Design Of High Performance IEEE- 754 Single Precision (32 bit) Floating Point Adder Using VHDL" In this paper Floating Point arithmetic is by far the most used way of approximating real number arithmetic for performing numerical calculations on modern computers. The advantage of floating-point representation over fixed-point and integer representation is that it can support a much wider range of values. Addition/subtaraction,Multiplication and division are the common arithmetic operations in these computations.Among them Floating point Addition/Subtraction is the most complex one.This paper implements an efficient 32bit floating point adder according to ieee 754 standard with optimal chip area and high performance using VHDL .The proposed architecture is implemented on Xilinx ISE Simulator.Results of proposed
architecture are compared with the existed architecture and have observed reduction in area and delay . Further, this project can be extendable by using any other type of faster adder in terms of area, speed and power.

[4] F.-C. Cheng, S. H. Unger and M. Theobald, "Self-Timed Carry-Lookahead Adders," IEEE TRANSACTIONS ON COMPUTERS, vol. 49, no. 7, pp. 659-671, JULY 2000. A Subsystems, in asynchronous circuits, usually need a subsequent start and completion mechanisms that is to be used to synchronize with one another subsystem. The major advantage an asynchronous circuits is it operate at average rate while synchronous circuits are operate at the worst rates.

[5] BehnamAmelifard, FarzanFallah and MassoudPedram, "Closing the Gap between Carry Select Adder and Ripple Carry Adder: A New Class of Low-power High-performance Adders," in Quality of Electronic Design, ISQED 2005. Sixth International Symposium on21-23 March, 2005.In this paper authors discussed on the idea of sharing two adders used in the Carry Select Adder (CSA), aand presented a new design of a low-power high performance adder. Authors compare the speed of a new adder against a Ripple Carry Adder (RCA), and found that new adder is faster than Ripple Carry Adder but slower than a CSA. While in terms of its area and power dissipation new adder occiped a smaller area and lower power than those of a CSA.

[6] Y. Li and W. Chu, "Implementation of Single Precision Floating Point Square Root on FPGAs," in Proc. Of IEEE Symposium on FPGAs for Custom Computing Machines IEEE Computer Society Press, 1997, pp. 226- 232..In paper authors, discuss on a non-restoring square root algorithm and single precision floating point square root implementations based on the algorithm on FPGAs. Authors implemented traditional adder and substractor for low cost with a latency is 25 clock cycles. And implemented a high-throughput pipelined using multiple adder and subtractors with latency of 15 clock cycles . Authors concluded with that the pipelined implementation is capable of accepting a square root instruction on every clock cycle.

[7] B. Fagin and C. Renard, "Field programmable gate arrays and floating point arithmetic," IEEE Transactions on VLSI, vol. 2, no. 3, pp. 365-367, 1994.In this paper authors, discussed on a high speed floating point double precision adder, subtractor and multiplier. Furthermore authors implement design of high speed floating point double precesion adder on a Virtex-6 FPGA. Additionally authors, proposed a designs for IEEE-754 format, to handles problem of over flow, under flow, rounding and various exception conditions.

[8] A. Jaenicke and W. Luk, "Parameterized Floating-Point Arithmetic on FPGAs," in Proc. of IEEE ICASSP, 2001 vol.2, pp. 897-900.In paper authors presented the double precision floating point adder/subtractor and multiplier supports the IEEE-754 binary interchange format, targeted on a Xilinx Virtex-6 FPGA. And authors compared with a

single precision floating point multiplier, and the multiplier design supports double precision, and cocluded that the double precison provides high speed and gives more accuracy. Additionally the presented designs handles the overflow, underflow, rounding mode and various exception conditions.

[9] Michael L. Overton "Floating Point Representation". [Online].Available:http://homepage.cs.uiowa.edu/~atkinson/m170.dir/overton.pdf..In paper the Michael L. Overton has written a great literature on the details of floating point presentation of different IEEE standards with examples for understanding of such systems.

[10] K. Ramu and B. S. Rao, "Implementation of Area Efficient 16bit Adder in SPARTAN-3 FPGA," International Journal of Engineering Science and Innovative Technology (IJESIT), vol. 2, no. 2, March 2013.Different types of adders (Carry Select Adder, Ripple Carry Adder, etc.) with different approaches (Parallel Adders, High-speed Adders, and Low-power High-performance Adders) have been presented.

[11]L. Gerlach, G. Payá-Vayá and H. Blume, "Efficient Emulation of Floating-Point Arithmetic on Fixed-Point SIMD Processors," in 2016 IEEE International Workshop on Signal Processing Systems (SiPS), 26-28 Oct. 2016. In this paper, an author evaluate the fixed-and floating-point signal processing algorithms through fixed-point implementation and show difference in performance, precision and code size.

In [12] authors discussed on the architecture, design, and testing of the first 16-bit asynchronous wave-pipelined adder and its implementation . In [13][14][15] and [16] the authors describe designs operate on a single bit or a small group of bits sequentially at a very high processing rate with bit-serial or digit-serial architectures. These designs are using clocking and compact structure. Although the latency of serial adders is added together to form a long latencies for 8-/16-/32-/64-/128 bit operations with general purpose processors.

Pongyupinpanich Surapong et al [17] presented the CORDIC Algorithm based Floating-Point Division Operator, for that they developed an efficient architecture of the CORDIC algorithm to make its processing capability much broader input ranges. The optimization algorithm for CORDIC-based FFT units with Fixed-Point Accuracy Analysis is presented by Omid Sarbishei et al [18].The mathematical co-processor core for 32-bit floating-point capable of handling overflow, under flow and special number (infinity and not-a-number) representations, using the CORDIC algorithm is presented by Tanya Vladimirova et al[19] . The co-processor also facilitates thirteen elementary functions (like adding, subtracting etc.). The non- CORDIC approach for floating point multiplier with technologyindependent pipelined design is presented by Mohamed Al-Ashrafy et al [20].

The multiplier implementation also handles the overflow and underflow cases. The performance analysis of different floating point multiplier designs is presented in[21].Self-Timed Asynchronous Sub-threshold Logic systems are studied by NiklasLotze et al [22], with discussionfor necessary timing overheads and approach for their analysis and reduction by the use of circuit techniques. Implementation of Phased Logic (PL) systems by self-timed programmable architecture is discussed by CherriceTraver et al[23].

In [24] and [25] the authors discuss on different aspects of designing Single and double precision Floating point on FPGA with delay analysis for a carry look ahead adder. In[26] the authors share different quantitative analysis of floating point arithmetic and share different solutions for FPGA implementation.

III. PROBLEM FORMULATION

Floating Point (FP) addition, subtraction and multiplication are widely used in large set of scientific and signal processing computation.As we have already seen that multiplier is one of the key component which performs a complex function in the system due to which delay occurs. This lag causes hindrance in other operations too. Further, while we see that major drawback of a conventional multiplier is its speed, the main reason behind it is the process of multiplication. Suppose if we consider shift and add multiplier we can see that the number of partial products generated are quite high. Due to which more number operations have to be performed. This is one of the major problem which needs to be overcome. Secondly, we have observed that multiplier can be made using different types of adder. Speed and efficiency of a multiplier largely depends on which adder is used. Therefore, we had to make a comparative study using different adders. Power and area requirement is of other adders is also large. Hence, we had to select such type of adder which provides optimization of the basic requirements.

The CLA adder is a fast carry-propagate adder which reduces carry propagation time to a logarithmic order by predicting future carries based on a priori knowledge of the input signals, instead of ascertaining carries on a stage-bystage basis unlike the ripple carry adder (RCA).In our work we have describedour objective to design and implement of Carry Look Ahead Adder for 32 bits with minimum area and transition delay. To fulfil our objectives, we have gone through with existing floating point architecture with their specification, different adder such as serial and parallel form. We have selected IEEE 747 adder architecture to compare synchronous and our proposed carry look ahead adder. We have to identify different recursive functions for carry generation and propagation from one stage to another. It includes nesting of same function and higher order of lookup to get resultant bit representation with standardIEEE 754. Here we are focused on a single precision standardized binary exchange format (32 bit floating point).

Apart from this we have to understand technical and semantic concept of hardware description language called as VHDL. Also learn about simulation tools and its different modules for analysis and testing. Here we have selected Xillinx 14.4 as our EDA tool.

IV. CONCLUSION AND FUTURE WORK

Multiplication is the most important arithmetic operation in many applications . As speed is always constraint in the multiplication operation, hence increasing the speed is very important . The speed of the multiplier determines efficiency of the system . All the signal and data processing operations involve multiplication. In any system design, the three main constraints which determine the performance of the system are speed, area and power requirement. We have done designing of carry look ahead adder through Xillinx EDA tool with a comprehensive simulation testing with all aspects. We have a comparative different reports for time delay and power dissipation analysis with configuration report. As shown in table 1, 2 and 3, multiplier with self-timed CLA shows much better results than with synchronous CLA. The Simulation result shows that multiplier with self-timed CLA takes less time to generate final product than with synchronous CLAbecause of worst case delay. Similarly, result shows about 20% enhancement in power consumption in case of self-timed CLAbased multiplier.Although our work presents architecture for self-timed 32 bit floating point multiplierin future, it can be extended for 64 bit multiplication.

References

[1] I. Koren, Computer Arithmetic Algorithms, New Jersey: Prentice-Hall Inc., Englewood Cliffs, 07632,, 1993.

[2] J.-C. Lo, "A fast binary adder with conditional carry generation," IEEE Trans. Computer, vol. 46, no. 2, p. 248.253, Feb. 1997.

[3] P. Balasubramanian, D. Dhivyaa, J. Jayakirthika, P. Kaviyarasi and K. Prasad, "Low power self-timed carry lookahead adders," in Circuits and Systems (MWSCAS), 2013 IEEE 56th International Midwest Symposium , Columbus, OH, 4-7 Aug. 2013.

[4] F.-C. Cheng, S. H. Unger and M. Theobald, "Self-Timed Carry-Lookahead Adders," IEEE TRANSACTIONS ON COMPUTERS, vol. 49, no. 7, pp. 659-671, JULY 2000.

[5] BehnamAmelifard, FarzanFallah and MassoudPedram, "Closing the Gap between Carry Select Adder and Ripple Carry Adder: A New Class of Low-power High-performance Adders," in Quality of Electronic Design, ISQED 2005. Sixth International Symposium on21-23 March, 2005.

[6] Y. Li and W. Chu, "Implementation of Single Precision Floating Point Square Root on FPGAs," in Proc. Of IEEE Symposium on FPGAs for Custom Computing Machines IEEE Computer Society Press, 1997, pp. 226- 232..

[7] B. Fagin and C. Renard, "Field programmable gate arrays and floating point arithmetic," IEEE Transactions on VLSI, vol. 2, no. 3, pp. 365-367, 1994.

[8] A. Jaenicke and W. Luk, "Parameterized Floating-Point Arithmetic on FPGAs," in Proc. of IEEE ICASSP, 2001 vol.2, pp. 897-900.

[9] Michael L. Overton "Floating Point Representation". [Online]. Available: http://homepage.cs.uiowa.edu/~atkinson/m170.dir/overton.pdf..

[10] K. Ramu and B. S. Rao, "Implementation of Area Efficient 16bit Adder in SPARTAN-3 FPGA," International Journal of Engineering Science and Innovative Technology (IJESIT), vol. 2, no. 2, March 2013.

[11] L. Gerlach, G. Payá-Vayá and H. Blume, "Efficient Emulation of Floating-Point Arithmetic on Fixed-Point SIMD Processors," in 2016 IEEE International Workshop on Signal Processing Systems (SiPS), 26-28 Oct. 2016.

[12] Mnaka, H. Akaike, A. Fujimaki, Y. Yamanashi, N. Yoshikawa, S. Nagasawa, K. Takagi and N. Takagi, "100-GHz single-flux-quantum bit-serial adder based on 10-kA/cm2 niobium process," IEEE Trans. Appl.Supercond., vol. 21, no. 3, p. 792–796, Jun. 2011.

[13] A. F. Kirichenko and O. A. M. T. Mukhanov, " Implementation of novel "pushforward" RSFQ Carry-Save Serial Adders," IEEE Trans. Appl. Supercond., vol. 5, no. 2, p. 3010–3013, Jun. 1995.

[14] A. Y. Kidiyarova-Shevchenk, K. Y. Platov, E. M. Tolkacheva and I. A. Kataeva, "RSFQ asynchronous serial multiplier and spreading codes generator for multiuser detector," IEEE Trans. Appl. Supercond., vol. 13, no. 2, p. 429–432, Jun. 2003.

[15] S. V. Polonsky and A. V. Rylyakov, "RSFQ arithmetic blocks for DSP applications," IEEE Trans. Appl. Supercond., vol. 5, no. 2, p. 2823–2826, Jun. 1995.

[16] H. Park, Y. Yamanashi, N. Yoshikawa, M. Tanaka and A. Fujimaki, "Design of fast digit-serial adders using SFQ logic circuits," IEICE Electronics Express, vol. 6, no. 19, pp. 1408-1413, 2009.

[17] PongyupinpanichSurapong and F. A. Samman, "Floating-Point Division Operator based onCORDIC Algorithm," ECTI Transactions on Computer and Information Technology, vol. 7, no. 1, May 2013.

[18] OmidSarbishei and KatarzynaRadecka, "On the Fixed-Point Accuracy Analysis and Optimization of FFT Units with CORDIC Multipliers",," in 20th IEEE Symposium on Computer Arithmetic., 2011.

[19] T. Vladimirova, D. Eamey, S. Keller and P. S. M. Sweeting, "Floating-Point Mathematical Co-Processor for a Single-Chip On-Board Computer," in Surrey Space Centre School of Electronics and Physical Sciences University of Surrey, Guildford, UK, GU2 7XH, Guildford.

[20] M. Al-Ashrafy, A. Salem and WagdyAnis, "An Efficient Implementation of Floating Point Multiplier," in Electronics, Communications and Photonics Conference (SIECPC), 2011 Saudi International, 24-26 April 2011..

[21] R. P. Singh, P. Kumar and B. Singh, "Performance Analysis of 32-Bit Array Multiplier with a CarrySave Adder and with a Carry-Look-Ahead Adder,"

International Journal of Recent Trends in Engineering, vol. 2, no. 6, November 2009.
[22] NiklasLotze, MauritsOrtmanns and YiannosManoli, "A Study on Self-Timed Asynchronous Subthreshold Logic," in Computer Design, ICCD 2007. 25th International Conference on 7-10 Oct. 2007, 2007.
[23] CherriceTraver, R. B. Reese and M. A. Thornton, "Cell Designs for Self-Timed FPGAs," in ASIC/SOC 14th Annual IEEE International , 2001.
[24] N. Shirazi, A. Walters and P. Athanas, "Quantitative Analysis of Floating Point Arithmetic on FPGA Based Custom Computing Machines," in Proc. of IEEE Symposium on FPGAs for Custom Computing Machines IEEE Computer Society Press, 1995, pp. 155-162.
[25] L. Lourca, T. A. Cook and W. H. Johnson, "Implementation of IEEE Single Precision Floating Point Addition and Multiplication on FPGAs," in Proc. of IEEE Symposium on FPGAs for Custom Computing Machines, IEEE Computer Society Press,, 1996, pp. 107-116..
[26] M. E. Louie and M. D. Ercegovac, "Mapping Division Algorithms to Field Programmable Gate Arrays," in Proc.26th Asilomar Conference on Signals, Systems, and Computers, 1992, pp. 371-375.
[27] S. K. V. K. V. A. M. P. Nagaraj Y and Dr. Chirag Sharma, "FPGA implementation of different adder architectures," International Journal of Emerging Technology and Advanced Engineering, vol. 2, no. 8, pp. 362-364, August 2012.
[28] Z. Navabi., VHDL Analysis and Modeling of Digital Systems, Mc Graw-Hill, Inc, 1993.

