International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Area Optimized Double Precision IEEE Floating Point Adder

Elizabeth Joseph Mattam[1], Deepa Balakrishnan[2]

[1]M.Tech Student, Department of Electronics and Communication Engineering
SCMS School of Engineering and Technology, Karukuuty, Cochin, Kerala, India

[2]Assistant Professor, Department of Electronics and Communication Engineering
SCMS School of Engineering and Technology, Karukuuty, Cochin, Kerala, India

Abstract: ***The fields of science, engineering and finance require manipulating real numbers efficiently. Since the first computers appeared, many different ways of approximation real numbers on it have been introduced. One of them, the floating point arithmetic, is the most efficient way of representing real numbers in computers. Representing an infinite, continuous set of (real numbers) with a finite set of (machine numbers) is not an easy task: some compromises must be found between speed, accuracy and efficient use and also implementation and memory cost. Floating Point Arithmetic represent a very good compromise for numerical applications. Floating Point (FP) addition, subtraction and multiplication are widely used in large set of scientific and signal processing computation. Although the concept of Floating-Point addition is easy it imposes a immense challenge while implementation of complex algorithm in hard real-time due to the enormous computational burden with repeated calculations with high precision numbers. A novel technique to implement a double precision IEEE floating-point adder which can complete the operation within two clock cycles. The proposed technique has exhibited improvement the operational chip area management by modifying the carry select adder. Also a decrease in power is also expected since area and power are directly proportional.***

Keywords: Area optimized carry select adder, Floating point adder, area and power reduction,CSLA,Clock cycles.

1. Introduction

Efficiently using the chip area and resources of an embedded system poses a great challenge while developing new algorithm in the embedded platform for hard real-time applications, like the control systems, digital signal processing, vision based sensing. Eventhough, different computational requirement of the algorithms involves different degrees of precision in the engineering and scientific applications, floating point operations are almost always employed in such applications for more accurate and more reliable algorithmic computations.

However, addressing the problem of floating point representation of numbers and the computational resources required while execution of the algorithm, at the software level, may not result in the optimal and dependable solution. Thereby, some hardware based solution at the chip development level is mostly suitable for the case where a dedicatedly digital circuit will be responsible for representing the floating point numbers as well as performing the arithmetic and logical operations as demanded by the algorithms. However, development of such a digital circuit for the purpose of representation of the floating point numbers and as well as performing the arithmetic and logical operations on them is quite difficult at the chip level due to the high level of complexities involved.

The most modern advancements in the area of Field Programmable Gate Array (FPGAs) has provided a lot of useful techniques and tools for the development of dedicated and reconfigurable hardware employing complex digital circuits at the chip level. Therefore, FPGA technology can be fruitfully utilized in order to develop digital circuits so that the hinderence of floating-point representation of numbers and the computational resources required while one performs the arithmetic operations during execution of the algorithm could be solved at the hardware level. This paper presents a unique technique to implement a double precision IEEE floating-point adder that can complete the operation with two clock cycles. A number of works have been reported in the literature with an aim to achieve a reduced latency of floating point operations. [14,11] The algorithm in [5] most effectively finishes the floating-point addition within two clock cycles with the packet forwarding format for handling data hazards in deeply pipe lined floating-point pipelines. The proposed technique has exhibited significant improvement in the optimal chip area management and implementing a dedicated double precision IEEE floating-point adder in FPGA based embedded system.

2. Area Optimized Double Precision Floating Point Adder

A. Modified Carry Select Adder

The main idea of modified work is to use BEC instead of the RCA with Carry=1 in order to reduce the latency and area utilization of the SQRT CSLA. We replace the n-bit RCA, with (n+1) bit BEC is depicted .Figure 1 illustrates how the basic function of the CSLA is obtained by utilizing the 4-bit BEC in conjunction with the mux. In this structure one input of the 8:4 mux gets as it input (B3, B2, B1, and B0) also another input of the mux is the BEC output. This produces the two partial outputs in parallel according to the control signal Cin. The importance of the BEC logic stems from the large silicon area reduction when the CSLA with large number of bits are designed.

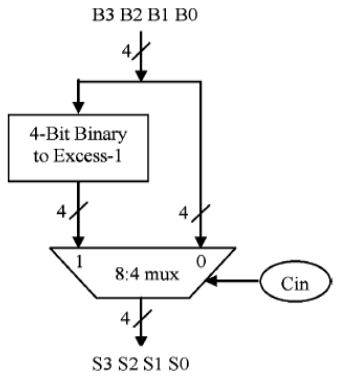

Figure 1: Block diagram of 4 bit BEC with 8:4 mux

The modified 16-bit SQRT CSLA using BEC is shown in Figure 2. The structure is again divided into five groups with different sizes of Ripple carry adder and BEC. The group2, group3, group4 and group5 of 16-bit SQRT CSLA are shown in Figure 3. The parallel Ripple carry adder with Cin=1 is replaced with BEC. One input to the multiplexer goes from the RCA with Cin=0 and other input from BEC. Comparing the individual groups of both regular and modified SQRT CSLA, it is clear that the BEC structure reduces area.

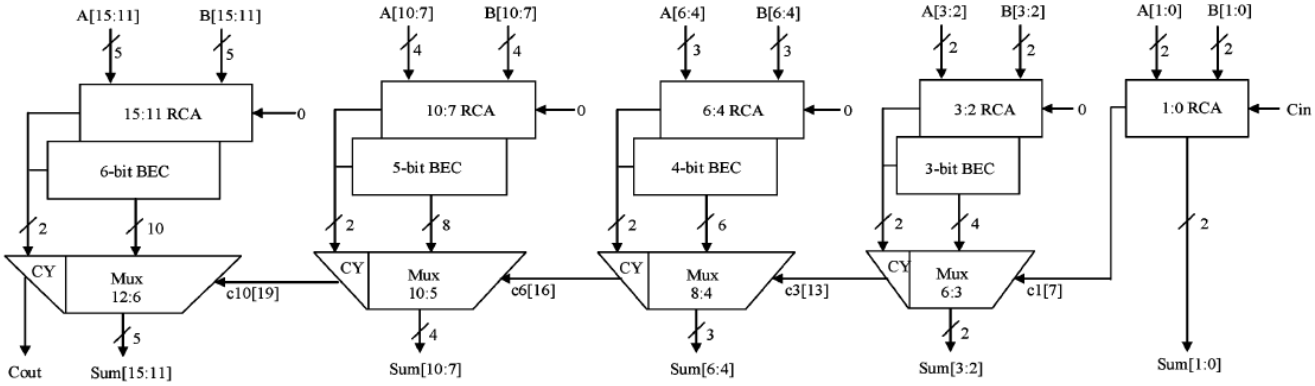

Figure2: Modified 16 bit SQRT carry select adder

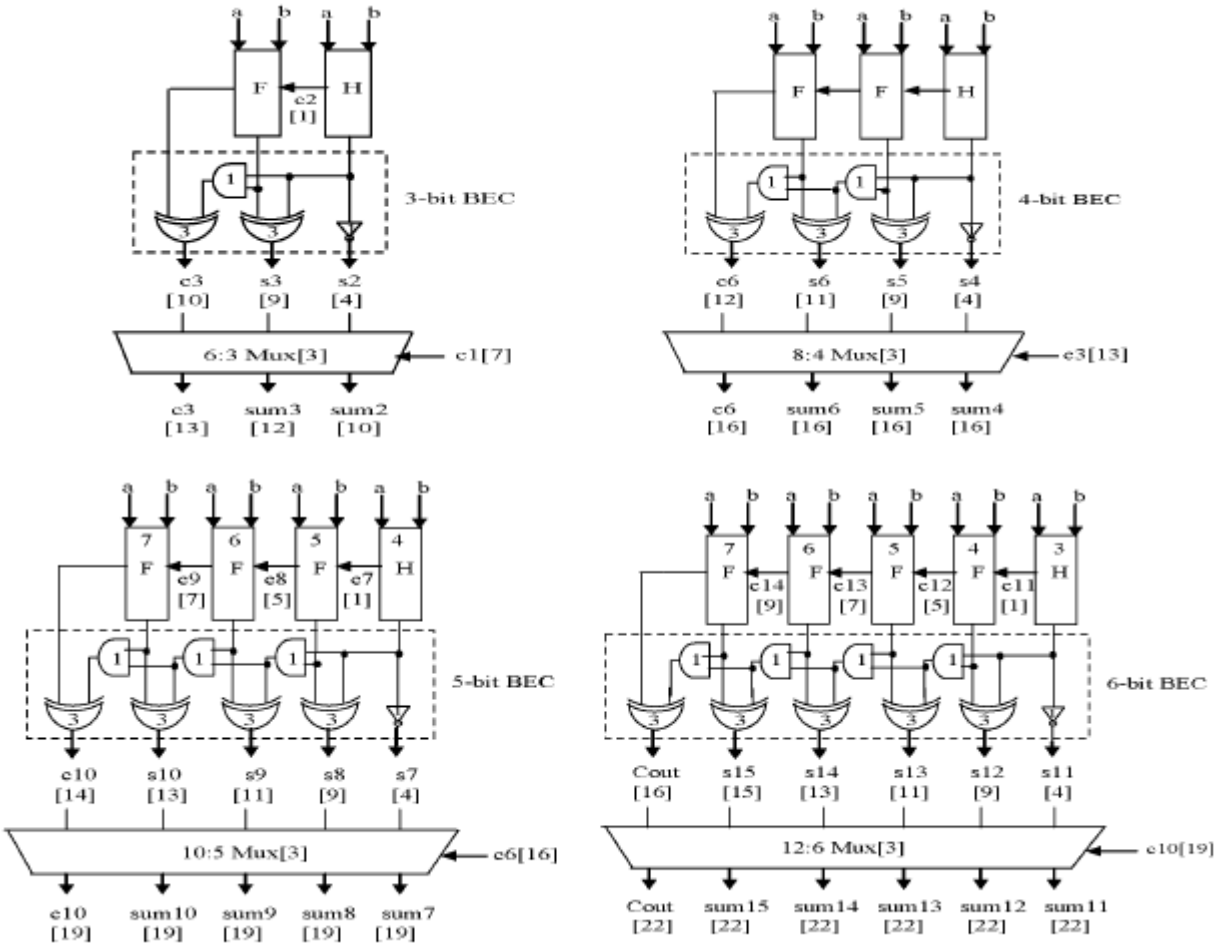

Figure 3: Individual groups of Modified 16 bit SQRT CSLA

The reduced number of gates of this work offers the great advantage in the reduction of area and also the total power. The comparison of results show that the modified SQRT CSLA has a slightly larger delay, but the area and power of the 64-bit modified SQRT CSLA are significantly smaller.

B. Modified Higher Level Representation of the Adder Algorithm

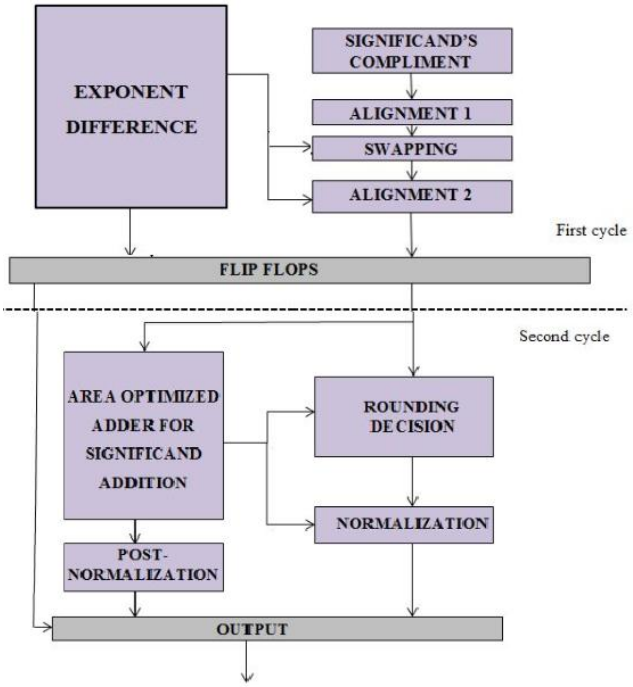

Figure 3: Modified higher level representation of the algorithm

Here all the steps followed in the previous versions are followed. But the only difference is we keep an area optimized adder for significand addition. The area optimized adder is explained above which is modified SQRT carry select adder.

B. First Clock Cycle

This is the first stage in the pipeline mechanism. The components of the Floating Point number, in terms of bit vector, are,

$$(S, E[0: 10], F[0: 52])$$

The basic algorithm operates only with normalized FP numbers. The basic operation is performed within two clock stages, and is determined by the parameter,

$$SOP = \{0,1\}$$

It is supplied as an input to the algorithm. The mathematical operation to be performed is determined by calculating the effective sign of operation.

$$S.EFF = sa \text{ xor } sb \text{ xor } SOP$$

After this, some initial pre-processing operations are done before adding or subtracting the two numbers. Then the exponent difference is obtained and is represented as

$$\delta = ea - eb,$$

Then the number with the smaller magnitude is sorted out through various operations based on conditions derived from the effective sign and the resultant of the exponent difference. In case the exponent difference is in the range [-63, 64] the smaller significand is shifted by MAG_MED positions to the right. The amount of alignment shift in medium range is determined by the modular value of the exponent difference δ, i.e. MAG_MED. The alignment shift can be formulated as:

$$(-1)^{SIGN\_MED}.[MAG\_MED] = \delta\text{-}1 \text{ if } 64 \geq \delta \geq 1$$
$$\delta \text{ if } 0 \geq \delta \geq -63$$

C. Second Clock Cycle

This is the second step of the pipelining mechanism. The two "pre-processed" significands are added and the result is rounded according to the IEEE standard rounding algorithm. Here the rounding algorithm from has been implemented. At the ending, it is normalized. The output result is a 64 bit binary floating point number.

$$rnd(sum) = rnd((-1)^{sa}.\ 2^{sa}.fa + (-1)^{(SOP+sb)}.\ 2^{sb}\ .fb)$$

A detailed block level representation of the second cycle of the algorithm is given in Figure 3.3. This approach is similar to [10] where the floating point arithmetic is a two stage pipelined and divided into two paths, namely "R-Path" and "N-Path". The two paths are selected on the basis of the exponent difference. The proposed algorithm was arrived at by following a few implemental changes in the algorithm of [2]. In [1], the dividing of the algorithm into two paths has been avoided, instead the algorithm has been modified to handle all the variations of input with agility

SIMULATION RESULTS

The waveform shown below gives the simulation result waveform of the double precision adder. The inputs are A and B of 64 bits each and clk and en. SOP is also given as the input.

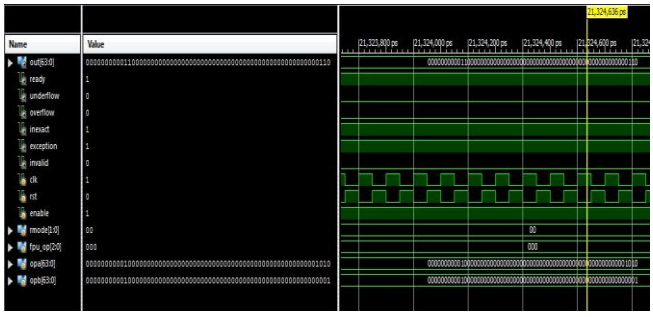

Figure 4: Output waveform of the IEEE double precision floating point adder

Device Utilization Summary				
Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	2,296	19,200	11%	
Number used as Flip Flops	2,296			
Number of Slice LUTs	1,544	19,200	8%	
Number used as logic	1,544	19,200	8%	
Number using O6 output only	1,302			
Number using O5 output only	52			
Number using O5 and O6	190			
Number of route-thrus	54			
Number using O6 output only	52			
Number using O5 output only	2			
Number of occupied Slices	755	4,800	15%	

Figure 5: Device utilization summery for area

The device utilization summary is depicted in Figure 5. It shows the number of flip flips, LUTs, slices used etc. Comparing with [1] a tremendous decrease in area can be observed in the proposed architecture.

3. Conclusion

floating point adders uses area and power intensively, but essential in high performance systems. in the design of integrated circuits, occupancy of area plays a important role because of increasing the necessity of portable systems. Efficiently using the chip area and resources of an embedded system poses a great challenge while we develop new algorithm in embedded platform for hard real-time applications, like control systems, digital signal processing, vision based sensing, and more. even though, different computational requirement of the algorithms involves different degrees of precision in most engineering and scientific applications, the floating point operations are almost always employed in such applications for accurate and reliable algorithmic computations. The design of an ieee double precision floating point adder is implemented here with lesser area. a modified version of sqrt csla is used for this purpose since it provides an optimal trade-off between area and delay. as area reduces, power also decreases, but it in turn results in a slight delay which can be ignored. hence this architecture is area and power efficient. the whole design was captured entirely in verilog using xilinx ise

References

[1] Ghosh, Somsubhra, Bhattacharyya, Prarthana and Dutta, Arka, “FPGA Based Implementation of a Double Precision IEEE Floating-Point Adder”, Jan 2013.
[2] Maroju SaiKumar et al, Design and Performance Analysis of Various Adders using Verilog, International Journal of Computer Science and Mobile Computing Vol.2 Issue. 9, September- 2013, pg. 128-138
[3] B. Ramkumar and Harish M Kittur, Low-Power and Area-Efficient Carry Select Adder, IEEE transactions on very large scale integration (VLSI) systems, vol. 20, no. 2, February 2012.
[4] R.Uma, Vidya Vijayan, M.Mohanapriya, and Sharon Paul, Area, Delay and Power Comparison of Adder Topologies, International Journal of VLSI Design & Communication Systems, vol.3, no.1, pp.153-168,Feb 2012.
[5] Karan Gumber, Sharmelee Thangjam, "Performance Analysis of Floating Point Adder using VHDL on Reconfigurable Hardware", International Journal of Computer Applications, vol. 46, no. 9, pp. 1-5, May 2012.
[6] Sarabdeep Singh, Dilip Kumar, Design of Area and Power Efficient Modified Carry Select Adder, International Journal of Computer Applications, vol.33, no.3, pp.14-18,Nov 2011.
[7] Padma Devi, Ashima Girdher, and Balwinder Singh, Improved Carry Select Adder with Reduced Area and Low Power Consumption, International Journal of Computer Applications, vol.3, no.4, pp.14-18,June 2010.
[8] Raminder Preet Pal Singh, Praveen Kumar, and Balwinder Singh, Performance Analysis of 32-Bit Array Multiplier with a Carry Save Adder and with a Carry Look Ahead Adder, Letters of International Journal of Recent Trends in Engineering, vol.2, no.6, pp. 83-89,Nov 2009.
[9] IEEE Computer Society, "IEEE Standard for Floating-Point Arithmetic", IEEE Std. 754T>1-2008 (Revision of IEEE Std 754-1985), Aug. 29, 2008.
[10] N. Kikkeri, P.M. Seidel, "An FPGA Implementation of a Fully Verified Double Precision IEEE Floating-Point Adder", Proc. of IEEE International Conference on Application-specific Systems, Architectures and Processors, pp. 83-88, 9-11 July 2007.
[11] Peter-Michael Seidel, Guy Even, "Delay-Optimized Implementation of IEEE Floating-Point Addition", IEEE Trans. on Computers, vol. 53, no. 2, pp. 97-113, Feb. 2004.
[12] A. Nielsen, D. Matula, e.N. Lyu, G. Even, "IEEE Compliant Floating-Point Adder that Conforms with the Pipelined Packet-Forwarding Paradigm," IEEE Trans. on Computers, vol. 49, no. 1, pp. 33-47, Jan. 2000.
[13] G. Even and P.-M. Seidel, “A Comparison of Three Rounding Algorithms for IEEE Floating-Point Multiplication,” IEEE Trans. Computers, vol. 49, no. 7, pp. 638-650, July 2000.
[14] A. Beaumont-Smith, N. Burgess, S. Lefrere, C. Lim, "Reduced Latency IEEE Floating-Point Standard Adder Architectures," Proc. of 14th IEEE Symposium on Computer Arithmetic, pp. 35-43, 1999.
[15] P.-M. Seidel, "On the Design of IEEE Compliant FloatingPoint Units and their Quantitative Analysis", PhD thesis, Univ. of Saarland, Germany, Dec. 1999.

Author Profile

Elizabeth Joseph Mattam received the B.Tech degree in Applied Electronics and Instrumentation Engineering from Mahatma Gandhi University, Kerala at Rajagiri School of Engineering and Technology 2012and now she is pursuing her M.Tech degree in VLSI and Embedded systems under the same university in SCMS School of Engineering and Technology, Cochin.