Design of Three-Input Floating Point Adder/Subtractor

[1]Ms. A. Niharika
M.Tech Scholar, Department of ECE, Sree Vidyanikethan Engineering College, Tirupati – 517102, India

[2]Mr. G. Naresh
Assistant Professor, Department of ECE, Sree Vidyanikethan Engineering College, Tirupati – 517102, India

[3]Ms. Neelima K
Assistant Professor, Department of ECE, Sree Vidyanikethan Engineering College, Tirupati-517102,India

***Abstract* – For precision-based applications like satellite communications, floating-point units are used. In single-precision floating-point unit, 32-bit data representation is used. In that MSB is used for sign, the next 8 bits are used for exponent and the remaining twenty three bits are used for mantissa representation. This term paper focuses on the development of a 3-input floating point adder/subtractor by using fast adders. This floating-point unit developed uses optimized LZA and LZC. The design is developed in Xilinx ISE tool using Verilog HDL. The design uses an area of 13.89% with a delay of 49.44 ns and power dissipation of 91mw for spartan3E FPGA.**

Keywords — Floating point Unit, Carry Save Adder, Binary Coded Decimal.

I. INTRODUCTION

In the design of digital processors and application of specific systems, digital arithmetic operations are critical. On modern computers, floating-point arithmetic may be a popular way of introducing real-numbers. The most basic concept behind the floating point representation is that only a finite number of bits are used, and the binary point "floats" to wherever it is required within those bits. On computers, there are many ways to represent real numbers. Floating-point demonstration, especially the quality IEEE format [21], is far the most similar way of on behalf of an estimate to real numbers in computers because it is easily handled in most large computer processors. The benefit of floating-point over an integer demonstration is that it can handle a much wider range of values. The most common solution, floating-point representation, represents reals in technical notation. Numbers are represented in scientific notation as a base number and an exponent.

$$Significant * Base^{Exponent}$$

In floating-point data paths, normalization is one of the most important operations. Normalization transforms the results of floating point procedure in its Normalized form, bestowing to the IEEE-754 standard., i.e.,1.yyy..y, where yϵ{0,1} based on the IEEE-754 standard. Normalization is performed using a detection unit, as well as a normalization shifter, and leading-zero anticipation logic is almost always used to speed up the calculation.

A. Significand

The portion of a binary floating-point number that involves of a fraction field towards the right of the oblique binary point of an obvious primary bit to the left.

B. *Biased Exponent***:**

The exponent plus a constant (bias) that makes the range of the biassed exponent nonnegative.

C. Normalized

The majority of calculations are made with normalized data. A normalized significant value is implied by the normalised number type (hidden bit is 1). The so-called "secret bit" the binary point is located to the left. This bit is inferred rather than stored in the floating point term. This one to the left of the binary point is essential for a number to be normalised.

D. Rounding:

It takes an ∞ precise number and, if essential, transforms it is fit the destination's format whereas signaling the inaccurate concession. As a result, the rounding mode has an impact on the outcome of most arithmetic operations, as well as the overflow and underflow exception thresholds. There are four rounding modes [21] specified in IEEE 754 floating point representation: round towards nearest even, round towards nearest odd. The defaulting rounding method is REN, which is widely used in all software and hardware arithmetic implementations. We are only interested in the default rounding mode is rounding to nearest even, in order to compare various adder algorithms. The representable value that is closest to the infinitely accurate result is chosen in this mode. If the two closest representable values are similarly close, the one with the lower value wins.

II. EXISTING ADDERS

A three input carry save adder is a type of digital adder that is used to compute in computer microarchitecture. In CSA, 3-bits are added parallel at the same period, but the carry doesn't pass up to the points; instead, it is kept in the current stage and reorganized in the next. Hence the interval due to carry is reduced.

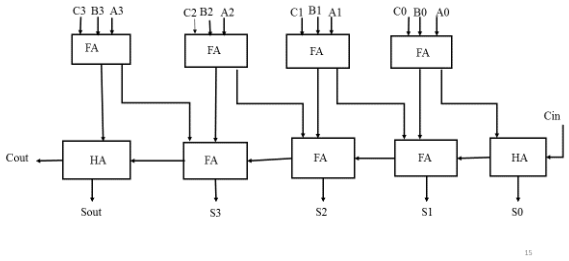

Fig.1. Three input Carry save adder

In this 3-input carry save adder is designed by using arithmetic logic circuits full adder and half adder of single bit number.

The BCD-Adder is a component found in computers and Calculators that perform arithmetic in decimal numbers directly system and accept binary-coded decimal numbers. It necessitates at least nine inputs and five outputs.

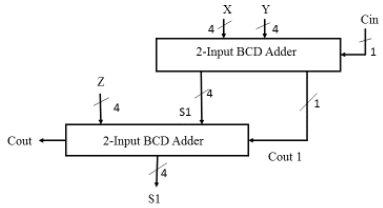

Fig.2. Three input Binary coded decimal adder

If the output result is less than nine, no correction is required. If the output result is greater than nine, add six to the output because each hexadecimal digit has 16 different values, while BCD only has ten. In BCD math, it's the same way.

III. PROPOSED DESIGN

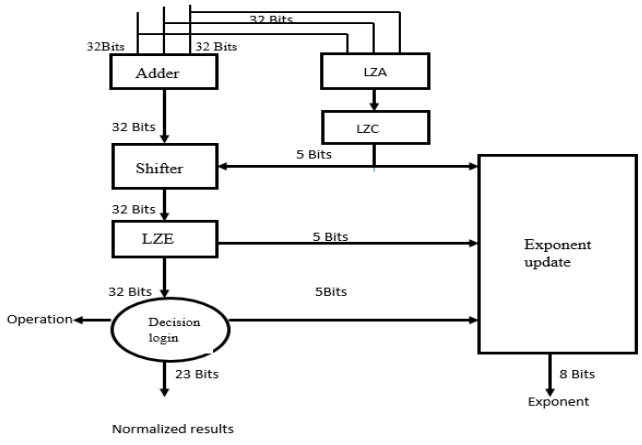

Fig.3. Three Input floating point adder

The adder and the leading zero expectation each get three 32-bit operands. The floating point addition operations are sped up using LZA and LZC. The LZA and LZC circuits encode the number of zeros in the result and feed it to the shifter along with the true adder result. The shifted result is fed to the leading zero anticipation error handler. The LZE verify for the existence of any leading Zeros and rectifies it. The output of LZC circuit is also fed to the exponent update unit. The exponent is reduced by the amount of number of leading zeros.

The decision logic it verifies the bit field and update the both significand and exponent field. In this unit output significand is selectively observed and rest bits are discarded. i.e.,32 bit is converted back to the 23 bit significand.

IV. RESULTS AND DISCUSSION

All the codes are modeled in Verilog HDL. They are functionally verified for Spartan3E series FPGA with part number XC7Z020-1CLG484, which is a 28nm FPGA, in Xilinx ISE 14.5 Tool.

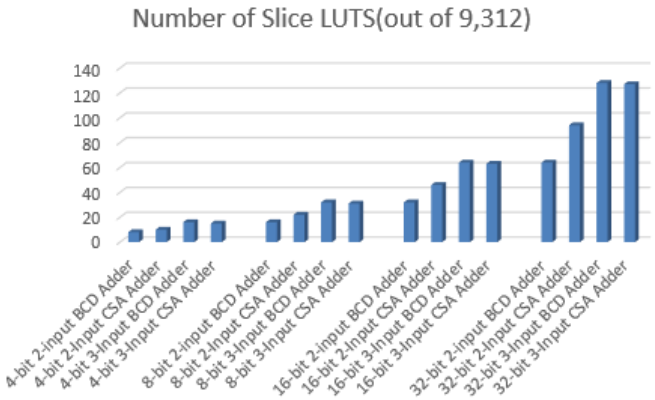

Fig.3. Comparion of Area for two input and three input Carry save adder and Binary coded decimal adders

In fig.3, the 2-input CSA area is decreased by 31.9% when compaired to 2-input BCD adder.

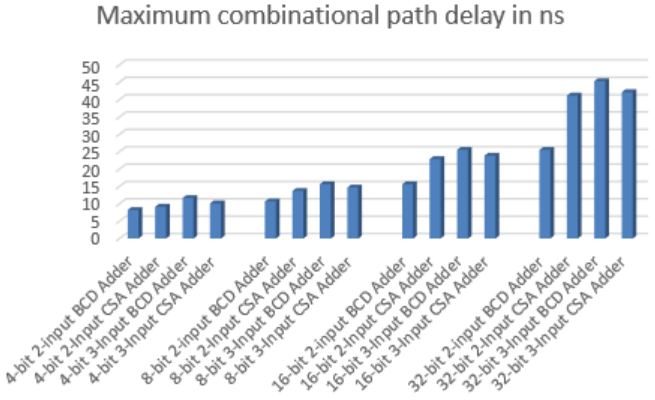

Fig.4. Comparion of delay for two input and three input Carry save adder and Binary coded decimal Adders

From Fig.4, 2-input CSA delay is decrease by 0.380% when compaired with 2-input BCD adder.

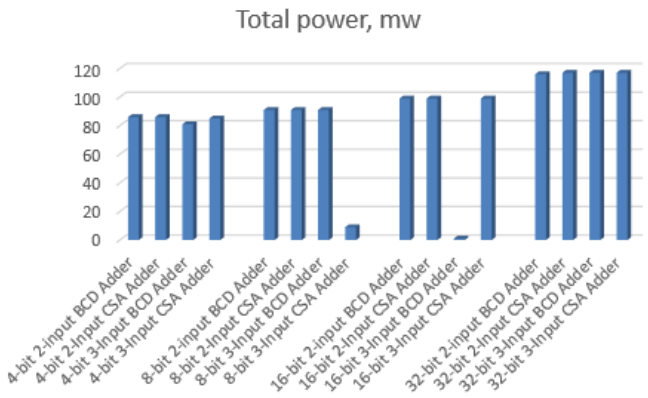

Fig.5. Comparion of Power Dissipation for two input and three input Carry save adder and Binary coded decimal Adders

From Fig.5, 2-input CSA power is decrease by 0.08% when compaired with 2-input BCD adder.

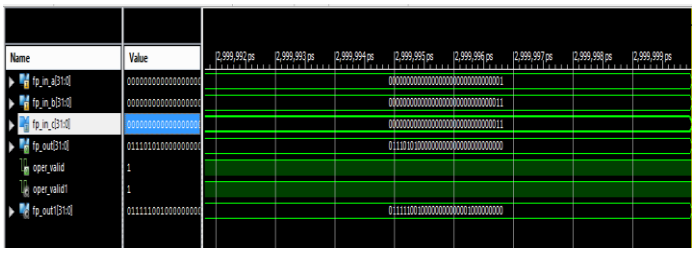

Fig.6. Three-input floating point Adder/subtractor

From fig.6, the simulation result shows the 3-input floating point Adder/ Subtractor. Where a=1, b=1 and c=1 then sum i.e., fp_out=3. Similarly, subtractor is also verified by using 3 – input CSA. The design is developed in Xilinx ISE tool using Verilog HDL. The design uses an area of 13.89% with a delay of 49.44 ns and power dissipation of 91mw for spartan3E FPGA.

CONCLUSION

In scientific calculations, the floating point unit is extremely useful. With the requirement of three point adder and subtractor for these units, we concentrate on developing 3-input adder/ subtractor. The adder for 3-input addition are done by BCD, CSA and floating point adder. The developed codes are modelled in Verilog HDL and verified in Xilinx ISE 14.5 Tool. Among them, CSA is found to be a better choice if BCD addition is not a criteria. This design uses an area of 13.89% with a delay of 49.44 ns and power dissipation of 91mw for spartan3E FPGA. In future the design is optimization by using fused floating-point ALU will be developed.

REFERENCES

[1] Kailash chandra ray and Amit kumar panda, "High-speed area-efficient VLSI architecture Of three-operand binary adder", Ieee transactions on circuits and systems–i: regular papers, vol. 67, no. 11, november 2020

[2] E. Prabhu , Dasari lakshmi prasanna, "An efficient fused floating-point dot product Unit using vedic mathematics", Proceedings of the third international conference on trends in electronics and informatics (ICOEI 2019)

[3] Diganta Sengupta, Atal Chaudhuri, Mahamuda Sultana,"Proposal for Fast BCD Addition", 2017 Third International conference on research in computational intelligence and communication network.

[4] Vassilis Paliouras and Giorgos Tsiaras, "Multi-operand logarithmic addition/subtraction based on Fractional Normalization", 2017 6th International Conference on Modern Circuits and Systems Technologies (MOCAST)

[5] Aditi Sharma, AbhaySharma, Sukanya Singh, "Implementation of Single Precision Conventional and Fused Floating Point Add-Sub Unit Using Verilog.

[6] Vinodkumar Jacob, Drusya P M, "Area efficient fused floating point three term adder", International Conference on Electrical, Electronics and Optimization Techniques (ICEEOT) – 2016.

[7] Jyoti sharma, sambangi Satishkumar, Pabbisetty tarun, and sivanantham S "Fused Floating-Point Add and Subtract Unit" Online International Confernece on Green Engineering andTechnologies (IC-GET 2015)

[8] Vinayak Patil, Aneesh Raveendran, A.David Selvakumar, P. M. Sobha,and D. Vivian. "Out of order floating point coprocessor for RISC VISA,"19th International Symposium on VLSI Design and Test, 2015.