Analysis of effectiveness of power on refined numerical models of floating point arithmetic unit for biomedical applications

Dinesh Kumar J R¹ᵃ, Ganesh Babu C²ᵇ, Balaji V R¹, Visvesvaran C¹
¹Department of Electronics & Communication Engineering, Sri Krishna College of Engineering & Technology, Coimbatore, Tamil Nadu, India
²Department of Electronics & Communication Engineering, Bannari Amman Institute of Technology, Sathyamangalam, Tamil Nadu, India

Email: ¹dineshkumarjr@skcet.ac.in, ²bits_babu@yahoo.co.in

Abstract: “The technology defined today might be redefined tomorrow”- from the quotes, in the modern era redefining the architecture with the optimization of area, delay and reduction of power is focused more than fixed models. This paper describes the floating point arithmetic unit and insists the importance of it, in real time signal processing. The present trends of portable devices for medical applications need occupancy of smaller area with long span time life of battery. The bio medical signals are converted into IEEE 754 format as BIF. Hence it is easily analyzed in FPGAR devices and also this work insists the importance of redefining the architecture with reduction of critical path delay. The FPU arithmetic unit has the blocks to perform the computations like addition, Multiplication and division operation. The main architecture is constructed using top level design with individual blocks defined as a part of it. The floating point unit defined with modern FPGAR (Reconfigurable FPGA) increase the speed of computation and enables the flexibility and adopted to the hardware reconfigurable models. A 32 bit representation of IEEE 754 results of FPU is stored in the data storage and this process optimize the critical path between blocks. The way its programmed consumes lower level of power consumption in the range of 50.4 nw and delay of 25.2 ns and also modern adder structure with 6T type was used to implement the intermediate blocks of multiplier in FPU block as part of computing partial products. By this process of FPU redefining, reduce the number of slice to 206 and increase the operating frequency of 46.12 MHz.

1. Introduction

In the modern era of digital world all the process of computation plays important role. The computations are done by the arithmetic unit of the processors [1]. The need for the accuracy in floating point computations are increased compared to Integer computation techniques [2]. The processors which is defined to perform the specific signal processing applications like speech signal processing[8], digital signal processing are expected to have maximum efficiency and more accurate with precision values. But the in Bio medical applications the real time data are much complex to analyze as this data’s are having low bit variation and highly sensitive to external noises generated by the device. The error present in the signals will not define in precise way to identify the type of
problem associated with the human body. Hence defining the biomedical system with precision and high accurate data plays a vital role in biomedical processors. The main idea is to refine the processors in which floating point operations is optimized in terms of power and area occupancy of the device. The arithmetic unit of the processors is defined to perform the computation in floating point or integer way of representation. The accuracy of the processing data is in depended of the data obtained and it is depends on the no of bits taken for its representation [4].

The computation of any process is done through the addition operations. Even though computing operation of multiplication can be done using repetitive addition. Hence the application of real time signals and processing of it through floating point unit should provide more accurate and précised results of computation. Mostly the solution required for the different problems associated with the real time signals needs to be regularized with high level computation accuracy [5]. A Floating point representation of number should be mentioned in IEEE 754 format. The IEEE 754 format representation is implemented in most of the modern processors and controllers and this format representation is classified as three blocks connected in cascaded manner to indicate sign-bit, exponent-bit and mantissa-bit with bit size of 1,8 and 23 bits and the total size of 4 bytes (32 Bits) as shown below Fig.1 [16].

The effective use of silicon chip area, power reduction and clock tree synthesis are major issues in focus to the embedded processors which are designed to operate with real time signals. The allocation of silicon area has significant impact on the power consumption level of the processor in integer computations, but the advancement in integrated circuit fabrication techniques narrated that the Integer operation ALU and FPU has same constraints with respect to the area, delay and power [12]. So the implementation of floating point computation was done with the help of FPGAs available in today’s market with different algorithm. The different architecture of FPU can be simulated in the same FPGA using reconfigurable architecture [10]. The complexity of computation depends on the algorithm in programming levels of abstraction. The FPGA which is designed to implement the FPU operations can be redefined to develop the systems in the field of industrial control systems, Robotics, Digital computations and traffic controllers and biomedical [15] / DSP processors [8]. This paper is organized as sections of survey, computation methodology, result analysis and simulation results with conclusion in the following sections.

2. Related works and survey
The development of optimized model for floating point computation is always gets much priority, especially defining the processors associated with real time application and bio medical signal processing [6]. The various architectures are developed for forming the floated value computation, but the issues are related to either power or area. Importance of the architecture is defined in terms of blocks connected to it. Simple adder structure may not be used for floated value operations. The floating point unit should perform the computation of real time signals. Initially the real time signals are dynamic and it should be represented to certain defined format. The IEEE 754 standard format is referred with single floating point precision with the bit size of 32 bits, where as the double precision method consumes more storage area (64 bits) [7]. The full adder circuit which performs bit wise addition is extended for the bit operation of 32bit. The blocks of floating point representation which defines in the previous sections, is taken into the part of computation as sign-bit, mantissa-bit and exponent bit. The sign-bit is calculated by the Ex- OR operation, the sign-bit is asserted to “0” for negative number and “1” for positive number and the remaining block of IEEE 754 format to be taken to compute the other progression based on the computation [9]. The two different forms of floating point representation are there, like DIF (Decimal Interchange Format) & BIF (Binary Interchange
format) in VLSI, however the BIF format is referred mostly as ease of computation with 32 bits and also representing the binary number (1 & 0) is much easier then decimal representation. Hence the implementation of real time signals in BIF (IEEE 754) format reduce no of cycle of operation in terms of critical path delay computation.

Figure 2. Full adder circuit designed using 6T Structure

The figure 2 shows how the high speed adders (RPcY) are constructed using basic block of full adder with 6T as explained in [1]. The usage of this architecture is defined only for 4 bits of operation, so this structure is redefined to perform the computation of exponent bit(biased) of size 8 bits [3]. This structure is very useful as the number of transistor is reduced so the area of FPU can be reduced to the minimize level.

Figure 3 shows the another high speed adder (CSA) based computation for extracting the exponent bit [3]. The same part of the circuit can be used to redefine the multiplier block usually the multiplier is calculated by partial product (PP) calculations and this PP is stored in the floated buffer.

Figure 3. Carry save adder designed from FA of 6 T

The blocks of multiplier is redefining here using the high speed adder blocks and this ensures the reduction of delay and improves the speed of computation. The high speed operation of circuits ensures optimization on area and critical path delay. The biomedical signal [6] obtained from the human body is converted into 32 bits, by using various process like quantization and precision. After the conversion the corresponding DIF is converted to BIF, to perform the floating computations.

3. Computation methodology

3.1 Multiplication and Addition of Floated Values

The floating point computation is performed as specified in the flow graph. The operations involved in various steps. The initial process of operation is to get the numbers (1 & 2) to perform the desired operation [5]. The second step is to calculate the sign bit value using Ex-OR operation. Then check bit is used to decide the operation is addition or multiplication. This step is operated to identify the operation. If the selected operation is Addition of two floating numbers then MSB polarity is to be identified, whether it is addition of two (+) ve number or(-)ve number or the mixed of the polarity. Suppose if (I) both the MSB are positive then calculate the difference between the exponent bit and conclude the operation by checking the difference (d) bit, if this bit is equals to zero then interchanging of the number is not permitted otherwise shift the number 2 by the no of bits founded as difference [14]. Then calculate the new value of is added with the Number 1 after the shift of number 2 by the d bits and then add the significant bits of both numbers. Check for the carry is generated, if the carry is found add that with the “1” found in the previous steps.
Figure 4. Floating point Addition/ Multiplication

Align the number by 32 bits of standard IEEE 754 format. If the numbers are differ then compute the 2’s complement of number 2 and add them with the number 1. If the carry is generated, here it can be neglected and shift the result on left till the MSB becomes “1”, also make a note on the number cycles it is taken to do this operation. Then subtract the exponent bit from the count calculated in the above steps. To make the both exponent bits are equal append “D” (Exponent – Count) no of zero to this.

Similar to the steps the floating point multiplication is done by calculating the sign bit using EX-OR gate ad initial step. Then the exponent bits of both numbers were added and mantissa bit are multiplied. Generally the mantissa bit is represented in 23 bit. So the multiplication of mantissa might give total resultant bit of 46 bits [7]. Hence it requires the normalization process. The 46th bit is “1” then no need to shift the mantissa as its in the prescribed format, else the 47th bit is “1” then obtained mantissa has to be rotated to right and 1 bit is added with the exponent bit. The intermediate products obtained through the steps fix the decimal points when it is converted to decimal format representation. The overflow / underflow in the multiplier is not allowed since the FPU follows the standard form representation. The overflow is raised when the result is infinity, underflow is raised if the exponent bit is less than “0” [16]. So the compensation steps is done by adding bias or removing bias value depends on the value of exponent obtained from the previous step.

3.2 Division and Subtraction of Floated Values

The previous section discussed on how the floating point multiplication is performed. This section describes how the division is performed in floating point computations. All the basic computation of floating uses the same operation to calculate the sign value. Obtained the subtraction of two exponent bit and perform the division operation of mantissa of number 1 & 2 also perform the round off values and normalize the value of mantissa bit by shifting “1” to left or right. Overflow may occur if mantissa of number 2 is zero, ie, Divide by zero case will update the overflow operation. The floating point subtraction performed same as explained in floating additions. The floating point subtraction was done by performing addition of numbers 1 & the 2’s complement of number 2.
The developed architecture of Floating point computation methods like addition-multiplication & subtraction - divider were combined to refine the floating point arithmetic unit. This unit can consumes very less power compare to other structures since the optimization was performed on the silicon area allocation. The most critical path in building the final structure is supply distribution and clock synthesis along with critical path delay. The supply problem was compensated by constructing common rail at the core die area of the device [6]. The clock synthesis problem was nullified by performing de clone the clock gating element at the manufacture level. The critical path delay is reduced by connecting the register are closer and cutest the longest circuit path.

The floating point computation was defined in previous section. The need for floating point unit for real time signal processing applications depends on the power consumed and delay generated in critical path. In the redefined architecture the multiplier unit was constructed with the adder whose internal transistor count is less, 6 Transistor. The internal transistor count was reduced to improve the effectiveness of area of silicon die area and also the reduced no in operating transistor consumes less power, also the transistor was defined to operate with low operating voltage approximately 1.2V. The low transistor (6T) adder is replaced in the multiplier block of floating point unit to ensure the high speed in computation [1] and suppress the critical delay for generating the computed results [15]. The proposed FPU consists of floating point registers, Stack unit and buffers for accessing the data and moving the data’s to the further computation purposes. The decoder, rounding and normalization unit are placed in the purpose of maintain no of bits to be equal to IEEE 754 standard format [7], after the computation which defines in the precious section, calculate the 32 bit format IEEE 754 format which was divided into three blocks of sign, mantissa and exponent bit. The blocks of FPU is designed initially and suing the top order architecture implementation method from the base to final architecture was constructed. Here the common bus designed to have the operation of 32bits. The architecture was defined in programming as the per the HDL description [17] and the optimization was done to reduce the critical path delay and less switching operation ensures the low power operation of the entire circuit.

Figure 5. Floating Point Division
4. Results & discussions
The proposed methodology flow graph was constructed using System Verilog (C) language. The code was simulated in Model sim at the initial level. The top level model and the combined structure was simulated in Xilinx 14 and the results obtained shows as this methodology provides high reduction of power in the range of 50.42 nW and also the delay by 25.2 ns. The critical path time delay is vary from 0.24us to 0.30us, and this variations was depends on the computation operation preferred by the programmer.

This architecture was implemented in Spartan VI processor which has the operating core voltage of 1.2V and supports the maximum I/O and adapted to 45nm Technology with high logic pin connections.

Table 1: Delay & power comparison

| Types of Unit   | Delay (S) | Power (W) |
|-----------------|-----------|-----------|
| **Existing Method** | **##** | **##** |
| FP Adder        | 25.9μ     | 18.7n     | 203m   | 34.42 n |
| FP subtractor   | 27.0μ     | 21.6n     | 203 m  | 40.12 n |
| FP multiplier   | 18.5μ     | 16.9n     | 24m    | 60.81 n |
| FP Divider      | NA        | 16.2n     | NA     | 60.36 n |
| FP Arithmetic Unit | NA      | 25.2 n   | NA     | 50.42 n |

## Proposed Method
**Figure 7.** Comparison graph of existing system and proposed system

**Table 2.** Parameter of Refined multiplier in FPU

| Adders Replaced in Multiplier Unit | Speed (MHz) | No of Slices |
|------------------------------------|-------------|--------------|
|                                    | **          | ##           |
| Carry Save                         | 44.72       | 52.4         |
|                                    | 776         | 542          |
| Pipelined                          | 151.2       | 94.4         |
|                                    | 126         | 154          |
| FPU with Carry save adder based Multiplier | 58.72 | 36.48 |
|                                    | 446         | 168          |
| FPU with pipelined Multiplier      | 141.7       | 46.12        |
|                                    | 140         | 204          |

**Figure 8.** Comparison graph of refined FPU architecture

**5. Conclusion**

This paper explains the various steps involved in floating point computation like addition, multiplication and division. The process of computation and usage of optimization of programming
leads to reduction of critical path delay and area. The field of bio medical needs much accuracy with the real time signal process. The FPU defined on FPGA can be implemented to the DSP applications. Such that a way this system can be converted into single block of Simulink models in SCI lab or MAT lab tools. The FPGA was used here so that the hardware can be reconfigured using programming techniques so that architecture was modified with less no of transistor count and less occupancy of area in silicon Die. Hence this FPU can be used in real time (bio signal) devices for the improvement of computation speed and reduce the other parameters of VLSI like delay, area and power. The bio medical devices which uses this FPU have high span of time in power with more portable devices

6. REFERENCES

[1] Dinesh Kumar J R, Ganesh babu C, Karth S P 2019 Performance Analysis of 16 bit Adders in high speed computing applications” IEEE –Processings of ICACE
[2] Mandal. S.K. , Smaranika Routi “ Implementation of Low Power and High Speed Single Precession Floating Point Multiplier using Kogge-Stone Adder and Dadda Multiplier” IJVDCS, Sep2016
[3] Dinesh Kumar, C. Ganesh Babu “Survey on Consummation of Subsided Power and High Speed Adders For Mixed Signal Processing Applications” IJPAM, 2018.
[4] Sonali Mehta, Balwinder Singh, Dilipkuamr “Performance Analysis of Floating Point MAC Unit” International journal of computer applications, Sep 2013
[5] Lathapriya V and Sivagurunathan P “Design And Implementation Of Low Power Floating Point Unit Using Reconfigurable Data Path: A Survey” IJTCE, Feb 2016
[6] Dinesh Kumar, C. Ganesh Babu “Performance Investigation of Multiplier for Computing & Control Applications” in Jour of Adv Research in Dynamical & Control Systems, Vol. 10, 2018
[7] R. Prakash Rao “Implementation Of The Standard Floating Point Mac Using IEEE 754 Floating Point Adder” Proceedings of the Second International Conference on Computing Methodologies and Communication (ICMC 2018)
[8] Balaji, V R & Subramanian, S 2014, ‘A Novel Speech Enhancement Approach Based On Modified DCT and Improved Pitch Synchronous Analysis’, American Journal of Applied Sciences(SciencePub), vol.11, no.1, pp.24-37.
[9] Naresh Grover, M.K.Soni” Design of FPGA based 32-bit Floating Point Arithmetic Unit and verification of its VHDL code using MATLAB”Information Engineering and Electronic Business, 2014
[10] Michael Nachtigal, Himanshu Thapliyal and Nagarajan Ranganathan., “Design of a Reversible Floating-Point Adder Architecture,” proceeding of the 2011 11th IEEE International Conference on Nanotechnology Portland Marriott August 15-18, Portland, Oregon, USA, pp.451-456,2011.
[11] Yee Jern Chong, Sri Parameswaran, “Configurable multimode embedded Floating point unit for FPGA,” IEEE Transactions on Very large scale integration (VLSI) Systems, vol. 19 no. 11, 2011
[12] Balaji, V R & Thangamoni kha 2016, ‘Implementation Of An Adaptive Fir Filter Using High Speed Distributed Arithmetic”, Proceedings of International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST), Vol. 2, No. 10, pp. 191-201.
[13] X. Wang and B. E. Nelson, “Tradeoffs of designing floating-point division and square root on virtex fpgas,” in Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines, (Napa Valley, CA), pp. 195–203, April 2003.
[14] KKK.Deergha Rao, P.P. V. Muralikrishna “FPGA Implementation of 32 Bit Complex Floating Point MultiplierUsing Vedic Real Multipliers with Minimum Path Delay” 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering,(UPCON), 2018
[15] Balaji V R, Sathiya Priya.J 2018 “Enhancement of Speech Signal Using Modified Binary Mask Based Algorithm for Vehicular Noise”, Journal of Advanced Research in Dynamical & Control Systems, Vol. 10, 12-Special Issue, 2018
[16] Vladik Kreinovich and Siegfried Rump, “Towards Optimal Use of Multi-Precision Arithmetic: A Remark”, published in Reliable Computing, 12:365-369, 2006.

[17] Bhaskar J. “A VHDL Primer” Pearson Printice Hall Publication Third Edition 2007.

[18] A. M. Shams, T. K. Darwish and M. A. Bayoumi “Performance Analysis of Low Power 1-Bit CMOS full adder cells”, IEEE Transaction on VLSI Systems, Vol. 10, Feb. 2002.

[19] Rakesh Chowdary Gutta “Design and Analysis of one bit Hybrid full Adder using pass transistor Logic” IJPAM 2017

[20] Kazi Nikhat Parvin, Md.Zakir Hussain, Md.Ali Ghazi Islam “High precision arithmetic unit for biomedical and radar system Applications” by IEEE NetACT .978-1-5090-6590-5/17/ ©2017 IEEE.