Single Precision Floating Point Co-Processor For Statistical Analysis

Remya Jose
Department of Electronics & Communication Engineering
Rajagiri School of Engineering and Technology
Kochi, Kerala

Dhanesh M S, Asst.Professor
Department of Electronics & Communication Engineering
Rajagiri School of Engineering and Technology
Kochi, Kerala

***Abstract*— Floating point arithmetic is widely used in many areas, especially scientific computation and signal processing. The main applications of floating point numbers are in the field of medical imaging, biometrics, motion capture and audio applications.This paper presents the design and implementation of single precision floating point co-processor that can be used for statistical analysis.The design is coded in Verilog hardware description language at Register Transfer Level (RTL) and synthesized in virtex 5 device with the help of Xilinx ISE tool.**

Index Terms—floating point; single precision; co-processor; RTL;

I. Introduction

Representing and manipulating real numbers efficiently is required in many fields of science, engineering, finance and more.Since the early years of electronic computing, many different ways of approximating real numbers on computers have been introduced.One of the best ways of representing numbers is floating point representation.The term floating point is derived from the fact that there is no fixed number of digits before and after the decimal point. There are also representations in which the number of digits before and after the decimal or binary point is fixed and these numbers are called fixed-point representations. The advantage of floating-point representation over fixed point representation is that it can support a much wider range of values.

Floating point numbers generally follows IEEE-754 standard.The way floating point operations are executed depends on the data format of the operands. IEEE standard supports both single precision and double precision data formats. The main difference between single precision and double precision format is that single precision consists of 32 bits and the double precision consists of 64 bits.

Our goal is to design and implement a floating point co-processor that can be used for statistical analysis. With this goal of flexibility in mind, our processor was designed so that it can be configured to perform several useful functions.The main functions are addition, subtraction, multiplication, division, mean, median, and mode .The design is implemented using efficient adders and multipliers with reduced area and delay.

II. Single precision floating point number

Single-precision floating-point format is a computer number format that is used to represent a wide dynamic range of values.It is generally represented in IEEE-754 standard.In IEEE 754-2008 the 32-bit base 2 format is officially referred to as binary32.Fig:1 shows a single precision floating point format.Single precision floating point number consists of three fields

Sign:It is used to denote the sign of the number i.e. 0 represent positive number and 1 represent negative number.

Mantissa:It is part of a floating point number which represents the magnitude of the number.

Exponent:It is part of the floating point number that represents the number of places that the decimal point (binary point) is to be moved.

The general way of representing a single precision floating number using the IEEE 754 format is

$$x = 1^{s} \quad 1{:}f \quad 2^{e} \quad 127 \tag{1}$$

32 bits		
sign	exponent	mantissa
1 bit	8 bits	23 bits

Fig. 1. IEEE 754 single precision format

III. Floating point ALU

Floating point ALU mainly performs addition, subtraction, multiplication, division, mean, median, and mode.The input to the ALU is 32-bit operands represented in IEEE 754 standard.There are seven control signals in the floating point ALU.A multiplexer can be used to select the required operation.

A. Addition

One of the most complex operation which uses major delay and considerable area is floating point addition.From table 2, it is better to choose kogge stone adder than carry lookahead adder,since it provides less delay.

TABLE I
FLOATING POINT OPERATIONS

Control signal	Operation
3'b000	Addition
3'b001	Subtraction
3'b010	Multiplication
3'b011	Division
3'b100	Mean
3'b101	Mode
3'b110	Median

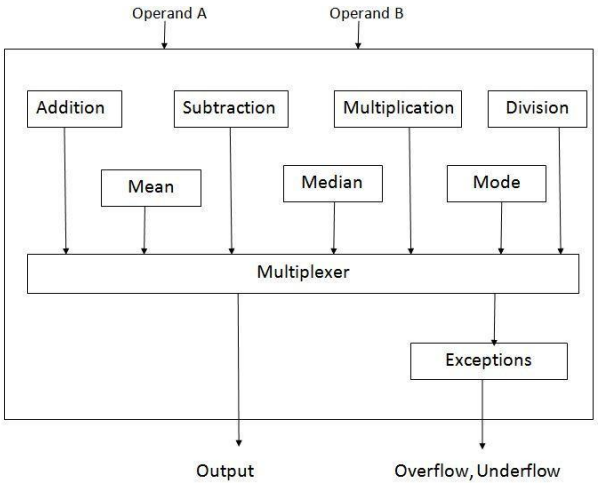

Fig. 2. Floating point ALU

The floating-point addition consists of five stages exponent difference, pre-normalisation, addition, post-normalization and rounding.

Pre-normalisation:The operands are transformed into formats that makes them easy and efficient to handle internally.

Post-normalisation:The result will be normalized if possible (leading bit before decimal point will be 1, if normalized) and then transformed into the format specified by the IEEE standard.

B. Subtraction

Subtraction operation is implemented by taking 2s com-plement of second operand. Similar to addition operation, subtraction consists of five major tasks exponent difference, pre-normalisation, addition, post-normalization and rounding.

C. Multiplication

In multiplication process 2n digit product will be produced by multiplying two n-digit operands.In IEEE 754 floating-point multiplication, the two mantissas are multiplied, and

TABLE II
COMPARISON OF AREA AND DELAY OF FLOATING POINT ADDERS

Adder	Area(Number of LUTs)	Delay(ns)
koggestone adder	410	54.261
carry lookaheadadder	356	67.451

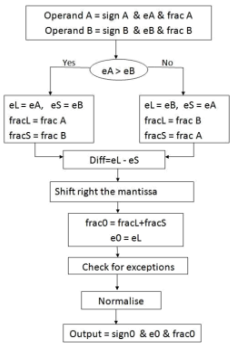

Fig. 3. Floating point addition

the two exponents are added. Here first the exponents are added from which the exponent bias(127) is removed. Then mantissas have been multiplied using feasible algorithm and the output sign bit is determined by exoring the two input sign bits. The obtained result has been normalized and checked for exceptions[2].

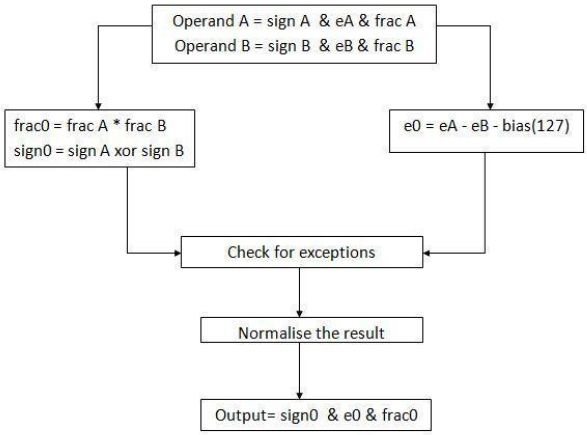

Fig. 4. Floating point multiplication

D. Division

Division is one of the most complex floating point oper-ation.In division two operands namely dividend and divisor is used to produce quotient and remainder.Here the exponent of result has been calculated by using the equation, e0 = eA eB + bias (127) -zA + zB followed by division of fractional bits[2].Sign of result has been calculated from exoring sign of two operands. Then the obtained quotient has been normalized. For division process Goldsmith (GDM) algorithm is used. For this algorithm to be applied needs both the inputs to be normalized first. In this both the multiplication of inputs is independent of each other, so can be executed in parallel, therefore it will increase the latency[1].

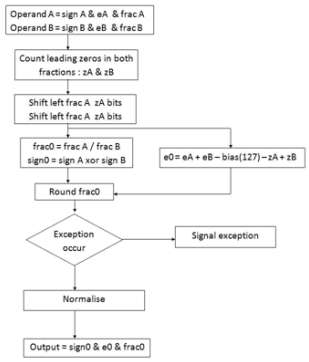

Fig. 5. Floating point division

E. Mean

The mean is also known as average.The mean can be found by adding up all of the given data and dividing by the number of data entries.The addition can be performed using the same procedure as floating point addition.For addition, koggestone adder is used.After obtaining the result, the result is being fed to floating point divider.The floating point divider divides the result by the number of data entries.

F. Median

The statistical median is the middle number in a sequence of numbers. To find the median,arrange the numbers in ascending order and then the number in the middle is the median. If there is an even set of numbers, average the two middle numbers.

G. Mode

The mode is the number that occurs most often within a set of numbers.Mode helps identify the most common or frequent occurrence of a characteristic. It is possible to have two modes (bimodal), three modes (trimodal) or more modes within larger sets of numbers.

H. Range

The range is the difference between the highest and lowest values within a set of numbers. To calculate range, subtract the smallest number from the largest number in the set.Range shows how much the numbers in a set vary.

IV. RESULTS

The design is carried out in Verilog HDL, synthesized and simulated using Xilinx ISE software.The simulation is done in the Xilinx Virtex 5 device.Timing and area analysis are also done to get optimised result.All the operations are embedded to form a floating point co-processor.

A. Floating point adder

In addition,two 32 bit operands are given as the input.Of this 32 bit,23 bits[22:0] consists of mantissa part,8 bits[30:23] consists of exponents and 1bit[31] for sign representation.At first exponents are made equal and then adding the mantissa using koggestone adder.For example, Let
w= 32'b00000000110001110010011011011110 0,
q=32'b00000000100000001100110011100001;then the output will be 32'b00000001111011101000000011110100

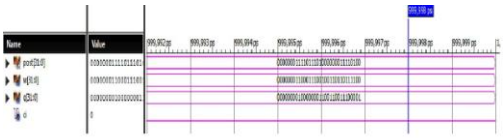

Fig. 6. Floating point adder

B. Floating point multiplier

In floating point multiplication,two 32-bit operands are used.Here the 23 bit mantissa is multiplied using booth multiplier.Since booth multiplier is used, the number of partial products will be less.The partial products are added using koggestone adder.The remaining exponent part is obtained by adding the exponents.Sign bit is obtained by exoring the two sign bits.For example, Let
m=32'b00000000101101100000000000000000,
n=32'b00000001000100001100000000111111;
then the outputs will be
product=45'b100001110100010101100110010100000000000000000,
diff=8'b10000100.

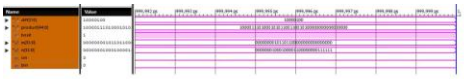

Fig. 7. Floating point multiplier

V. CONCLUSION

This paper deals with various arithmetic modules used to implement an efficient co-processor and also a comparative analysis of various floating point adders and multipliers on the basis of area and delay.Delay is one of the key elements in the implementation of processor.On the basis of comparative analysis, koggestone adder and modified booth multiplier were selected.Using these arithmetic modules an efficient co-processor was implemented.The design is completely done on Xilinx and synthesized using Xilinx tools.

REFERENCES

[1] Manisha Sangwan, A Anita Angeline,Design and Implementation of Sin-gle Precision Pipelined Floating Point Co-Processor, 2013 International Conference on Advanced Electronic Systems (ICAES).

[2] Ushasree G, R Dhanabal, Dr Sarat Kumar Sahoo, VLSI Implementation of a High Speed Single Precison Floating Point Unit using Verilog, pro-ceedings of 2013 IEEE Conference on Information and Communication Technologies(ICT 2013).

[3] Prashanth B.U.V, P. Anil Kumai, G. Sreenivasulu, Design and Implemen-tation of Floating Point ALU on a FPGA Processor, International Con-ference on Computing, Electronics and Electrical Technologies (ICCEET 2012), 2012

[4] Subhajit Banerjee Purnapatra, Siddharth Kumar, Subrata Bhattacharya, Implementation of Floating Point Operations on Fixed Point Processor An Optimization Algorithm and Comparative Analysis, IEEE 10th Inter-national Conference on Computer Information Technology (CIT 2010), 2010

[5] S. F. Oberman and M. J. Flynn, Division algorithms and implementations, IEEE Transactions on Computers, vol. 46, pp. 833854, 1997.

[6] Joy Alinda P. Reyes, Louis P. Alarcon, and Luis Alarilla, A Study of Floating-Point Architectures for Pipelined RISC Processors, IEEE International Symposium on Circuits and Systems, 2006.

[7] Yu-Ting Pai and Yu-Kumg Chen, The Fastest Carry Lookahead Adder, Department of Electronic Engineering, Huafan University.

[8] Ghassem Jaberipur, Behrooz Parhami, and Saeid Gorgin, RedundantDigit Floating-Point Addition Scheme Based on a Stored Rounding Value, IEEE transactions on computer, vol. 59, no.