ABSTRACT: A floating point arithmetic and logic unit design using pipelining is proposed. By using pipeline with ALU design, ALU provides a high performance. With pipelining plus parallel processing concept ALU execute multiple instructions simultaneously. Floating point ALU unit is formed by combination of arithmetic modules (addition, subtraction, multiplication, division), Universal gate module. Each module is divided into sub-module. Bits selection determines which operation takes place at a particular time. The design is and validated using vhdl simulation in the xilinx13.1i software.

Keywords - ALU - Arithmetic Logic Unit, Top-Down Design, Validation, Floating Point, Test-Vector.

I. INTRODUCTION

Floating point describes a system for representing numbers that would be too large or too small to be represented as integers. Floating point representation is able to retain its resolution and accuracy compared to fixed point representation. IEEE specified standard for floating-point representation known as IEEE 754 in 1985.

The IEEE 754 floating point format consists of three fields.

Sign bit: 1 bit. It is 1 for a negative number and 0 for positive number.

Exponent: 8 bits. The exponent represents a power of two.

Mantissa: Final portion of word (23 bits) is the significant that is also called as mantissa. Mantissa is a Fractional part.

Arithmetic logical unit is a combinational network that performs arithmetic and logical operation on the data. For computation input data is given to A.L.U. code is also given from control unit, according to that code it compute the result. The ALU with floating point operations is called a FPU.

Pipelining plus parallel processing execute is used to execute multiple instructions simultaneously. The cycle time of the processor is reduce. If pipelining is use, the CPU Arithmetic logic Unit can be design faster. It increases the overall performance of a system.

In the floating point ALU with universal logic gate we can perform addition, subtraction, multiplication, division operation and logical operation with less delay and less area.

II. LITERATURE REVIEW

By referring first this paper 16 bit floating point ALU is design using pipelining. Pipelining is use to execute multiple instructions simultaneously. Top-Down design approach is use. In top-down design approach, four arithmetic modules, addition, subtraction, multiplication and division are combined to form a floating point ALU unit. Each module is divided into sub-modules. Two selection bits are combined to select a particular operation. Each module is independent to each other. All modules in the ALU design are realized using VHDL, design functionalities are validated through VHDL simulation. All components and module is successfully run. Synthesis and Simulation in the Xilinx. The problem in this ALU is that hardware complexity in terms of synthesis is more. [1] 32 bit floating point ALU is design using pipelining. The design approach is same as that of design of 16 bit ALU. Concept of pipelining is use to execute multiple instruction simultaneously. Top down approach is also use. It performs four operations, Addition, subtraction, Multiplication and Division. Each module is independent to each other, all modules in the ALU design are realized using VHDL, design functionalities are validated through VHDL simulation. All components and module is successfully run, Synthesis and Simulation in the Xilinx. The problem in this ALU is that hardware complexity in terms of synthesis is more. [2]

A floating point arithmetic and logic unit design using pipelining. By using pipeline with ALU design, ALU provides a high performance. With pipelining concept ALU execute multiple instructions simultaneously. Top-Down design approach is use. Floating point ALU unit is formed by combination of arithmetic modules (addition, subtraction, multiplication, division), logical operation module (AND, OR, NOT). Each module is divided into sub-module is shown...
in fig.1. Bits selection determines which operation takes place at a particular time. The design is validated using vhdl simulation in the xilinx12.1i software. The problem in this ALU is that hardware complexity in terms of synthesis is more.[3]

32 bit floating point ALU is design. Floating point operations are hard to implement on Field Programmable Gate Arrays (FPGA ) because of the complexity of algorithms is more. Then again, many scientific applications require floating point arithmetic because of high accuracy in their calculations. In this paper an efficient implementation of an IEEE 754 single precision floating point arithmetic unit is designed in Xilinx SPARTAN 3E FPGA. VHDL environment is performed for floating point arithmetic unit design using pipelining, which provides high performance. Pipelining is used to execute multiple instructions simultaneously. In top-down design approach, four arithmetic modules, addition/ subtraction, multiplication and division are combined to form a floating point arithmetic unit. FP addition is implemented using Leading-One-Detector (LOD), Leading-One-Predictor (LOP) and two-path algorithms. In this ALU foradder module clock period is (LOD- 33.159ns, LOP-28.358ns, Tow-path- 22.313ns), for FP multiplier it is 10.402 ns, for FP divider it is 7.058ns. And the area in slices is for Adder module it is (LOD- 694, LOP- 731, Two-path- 1020), for FP multiplier it is 272, for FP divider it is 185. Synthesis and simulation results are obtained by using Xilinx13.1i platform. [4]

III. OVERALL ANALYSIS OF REPORTED WORK

From the overall analysis we can say that floating point ALU was design that performed arithmetic operation which include addition, subtraction, Multiplications, and Division operations. That ALU was designed using pipelining because of that the speed of ALU is increases and it executes multiple instructions simultaneously. That ALU was designed with Top down approach. In that ALU problem is that a more delay, more area, and hardware complexity in terms of synthesis is also more. This conclusion is made in [1], [2] and [4].After that one ALU was designed that performed Addition, subtraction, multiplication, division, AND, OR, NOT operation [3]. Same designing method was used. Same problem is present in this ALU also. Our approach is that to design 32 bit floating point ALU with universal logic gate, which can perform Addition, subtraction, multiplication, division operation with less delay and less area.

IV. PROBLEM DEFINITION

An A.L.U. performed addition, subtraction, multiplication, and division operation, latter on this, they add the logical operation which include AND, OR, NOT gate but the hardware complexity in terms of synthesis, delay and area is also more with less accuracy. So to avoid this problem we are designing 32 bit floating point A.L.U. which perform arithmetic operation which include addition, subtraction, multiplication, division operation and design of universal logic gate with high accuracy, high speed, less delay and less area.

V. PROPOSED METHODOLOGY

1) Designing of multiplication module.
2) Designing of addition/subtraction module.
3) Designing of Division module.
4) Designing of universal gate module.
5) To study the concept of pipelining, Parallel processing which is used to execute multiple instructions simultaneously
6) Comparison and study of the results

VI. PROBABLE OUTCOME

Proposed work will result 32 bit floating point A.L.U. with universal logic gate, which perform addition, subtraction, multiplication, division and designing of universal logic gate. This will meet the following specifications:
• Reduce Area
• Less delay

VII. AVAILABLE TOOLS

A) PIPELINED FLOATING POINT MULTIPLICATION MODULE:

In this 32 bit pipelined floating point A.L.U. is designed with four stage pipelining. Which has two 32 bit numbers given as input and got result is a 32 bit number, which is a multiplication of that two 32 bit numbers.

RTL SCHEMATIC
B) ADDER/SUBTRACTER MODULE:
1) Unpack: The sign, exponent and mantissa of both operands are separated. A flag, aequalb flag, is set if both inputs are equal. The aequalb flag will be used if the effective operation, determined in the adder/subtracter module, was subtraction to set a flag indicating a zero output. This prevents unnecessary addition/subtraction and pre normalization operations from taking place.
2) Swap: Inputs are swapped if necessary such that operand A carries the larger floating point number and operand B is the smaller operand to be pre-normalized. A swap flag is set if the operands were swapped to be used in determining the effective operation in the adder/subtracted module.
3) Zero Detect: An appropriate flag is set if one or both input operands is a zero. This helps avoid unnecessary calculations and normalizations when a zero operand is detected. The resultant exponent and the difference between the two exponents are determined here.
4) Pre-normalize: The smaller mantissa, of operand B, is pre-normalized, that is it's shifted by the difference between the two input exponents. Three extra bits the guard bit, the round bit, and the sticky bit are added to both mantissas to increase the accuracy of the performed operation (addition or subtraction) and to be used in the rounding process. Sticky bit is the logical "Or"ing of any bits that are dropped during the pre-normalization of operand B.
5) Adder/Subtracter: The effective operation to be performed is calculated according to the signs of operands A & B, the input operation and the swap flag. The effective operation is performed and the zero flag is updated if the effective operation is subtraction and the aequalb flag is set.
6) Post Normalize: The resultant mantissa is normalized after the leading one is detected using the LOD method. The resultant exponent is adjusted accordingly.
7) Rounding: The resultant mantissa is rounded using the REN technique. The final output is given in IEEE format.
In the 32 bit floating point ADDER/SUBTRACTOR we are giving two 32 bit numbers, and clock signal is also given for synchronization purpose. ADD_SUB signal is given when this signal is ‘1’ it perform addition operation, When ADD_SUB signal is ‘0’ it perform subtraction.
IX. CONCLUSION

High speed floating point multiplier based on the IEEE-754 single precision format is developed based on four stage pipeline technique and also designed a 32 bit ADDER_SUBTRACTOR module. Next I will design DIVISION MODULE, UNIVERSAL GATE MODULE, and then A.L.U.

ACKNOWLEDGMENT

I would like to thank my guide Dr. S. D. Chede and Prof. B.J.Chilke, for their guidance and support and Department of Electronics Engineering (Comm.), S. D. College of Engineering, Wardha.

REFERENCES

[1] Rajit Ram Singh 2Vinay Kumar Singh 3poornima shrivastav 4Dr. GS Tomar “VHDL environment for floating point Arithmetic Logic Unit – ALU design and simulation”
[3] Shuchita Pare*1 Dr. Rita Jain*2 “32 Bit Floating Point Arithmetic Logic Unit ALU Design and Simulation” international journal of imerging trend in electronics and computer science volume 1 issue, 2012
[5] Prashanth B.u.v \ P.Anil Kumai, G Sreenivasulu3 iMaintenance Engineer, DST-PURSE, 2M. TECH Student , 3Associate Professor, S. V University, Tirupati-517502 “Design & Implementation of Floating point ALU on a FPGA Processor” International Conference on Computing, Electronics and Electrical Technologies [ ICCEET],2012