IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, February 2026
Impact Factor: 7.67

A Verilog-Based Double-Precision FPU: From Algorithmic Design to Hardware Verification

Rakshitha S V[1], Reetu D[2], Sanjana S[3], Shreeganga H R[4]
Student, Department of Electronics and Communication Engineering[1-4]
Kalpataru Institute of Technology, Tiptur, India
rakshitharakshu935@gmail.com, reetu792005@gmail.com ,
sanjana.s.1526@gmail.com , shreehrganga@gmail.com

Abstract: *Floating Point Units (FPUs) are essential components of modern processors and digital systems, enabling high-precision arithmetic operations required in scientific computing, signal processing, and embedded applications. This paper presents the design, simulation, and hardware implementation of an IEEE-754 compliant single-precision Floating Point Unit using Verilog HDL. The proposed FPU supports basic arithmetic operations such as addition, subtraction, multiplication, and division, along with exception handling and normalization mechanisms. The architecture is modular, enabling easy integration into larger processing systems. Functional verification is performed using Modalism, and hardware feasibility is demonstrated through synthesis results. The proposed design achieves accuracy, modularity, and efficient resource utilization, making it suitable for embedded and high-performance computing applications.*

Keywords*:* Floating Point Unit, IEEE-754, Verilog HDL, ModelSim, Arithmetic Logic Unit, Digital Design

I. INTRODUCTION

Floating point arithmetic plays a critical role in modern digital systems where a wide dynamic range and high numerical precision are required. Applications such as digital signal processing, image processing, scientific simulations, and machine learning heavily rely on floating point computations. Unlike fixed-point arithmetic, floating point representation allows numbers to be expressed with a mantissa and exponent, enabling representation of both very large and very small values.

The IEEE-754 standard defines the format and behaviour of floating-point arithmetic, ensuring consistency and portability across different computing platforms. Designing an efficient Floating-Point Unit that complies with IEEE-754 standards is a challenging task due to complexities such as normalization, rounding, overflow, underflow, and exception handling.

This paper focuses on the design and implementation of a single-precision IEEE-754 compliant FPU using Verilog HDL. The proposed design emphasizes modular architecture, correctness, and suitability for hardware implementation. Simulation and hardware synthesis validate the functionality and feasibility of the proposed FPU.

II. IEEE-754 FLOATING POINT REPRESENTATION

The IEEE-754 single-precision floating point format uses a 32-bit representation, divided into three fields:
Sign bit (1 bit): Represents the sign of the number
Exponent (8 bits): Stored in biased form
Mantissa (23 bits): Represents the fractional part
The value of a floating-point number is given by:
where the bias for single precision is 127.
Special representations are defined for zero, infinity, NaN (Not a Number), overflow, and underflow. Proper handling of these cases is mandatory for IEEE-754 compliance.

III. LITERATURE SURVEY

The design of floating-point units on FPGAs has been widely studied, with various researchers proposing architectures optimized for performance, area efficiency, and configurability. A Verilog-Based Double-Precision FPU: From Algorithmic Design to Hardware Verification 2025-26

Dinechin and Pasca (2009) [1] developed *FloPoCo*, a generator for customizable floating-point data paths. Their tool demonstrated the advantages of parameterized hardware generation, producing efficient arithmetic cores for different precisions. However, the focus remained on configurability and synthesis automation rather than complete IEEE-754 compliance.

Wang and Leeser (2010) [2] introduced *Float*, a variable-precision floating-point library supporting adjustable exponent and mantissa widths. Although it enables flexible precision trade-offs, it requires higher logic utilization and control complexity, making it less suitable for fixed, double-precision designs.

Pasca (2012) [3] proposed a correctly rounded floating-point division architecture using multiply-based reciprocal approximation on DSP-enabled FPGAs. The design achieved significant latency reductions but depended on a large number of DSP blocks and deep pipelines, restricting portability to mid-range FPGAs.

Jaiswal and So (2017) [4] developed a dual-mode division unit that dynamically adjusts pipeline depth for performance and area optimization. While effective, the design complexity increases significantly with additional control logic.

From these studies, it is evident that existing designs largely emphasize speed and throughput on high-end devices, often sacrificing portability and full IEEE-754functionality. Few implementations report end-to-end verification on hardware platforms, leaving a gap for designs targeting mid-range educational SoC boards with verifiable compliance.

The present work addresses this gap by implementing a fully verified, resource balanced double-precision FPU for the Intel Cyclone V SoC, emphasizing functional completeness, portability, and verifiable hardware operation

IV. ARCHITECTURE DESIGN OF THE FPU

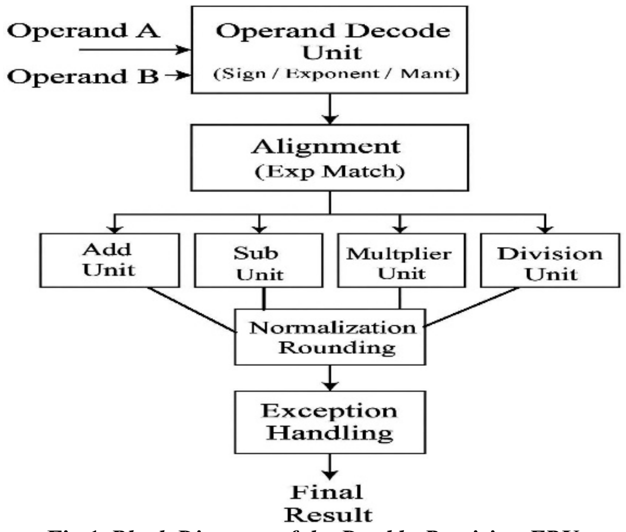


Fig 1:*Block Diagram of the Double-Precision FPU*

The concept, motivation, and objectives of designing a Double- Precision Floating-Point Unit on FPGA hardware. It outlined the systematic design methodology and the tool chain used for simulation, synthesis, and hardware testing.

The proposed design bridges theoretical arithmetic models and physical implementation, ensuring numerical accuracy and hardware efficiency. The following section presents a brief literature survey to contextualize this work within prior FPGA- based floating-point research.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, February 2026
Impact Factor: 7.67

V. MODULE DESCRIPTIONS

5.1 Addition Module

Performs IEEE-754 double-precision addition. Both operands are decomposed into sign, exponent, and mantissa. The smaller mantissa is right-shifted according to exponent difference for alignment. After addition, normalization is applied if overflow occurs, and intermediate results are forwarded to the rounding unit.

5.2 Subtraction Module

Implements floating-point subtraction by aligning exponents, subtracting smaller mantissa from larger, and re-normalizing. If the result underflows, the exponent is decreased accordingly. Sign inversion is applied when operand ordering requires it.

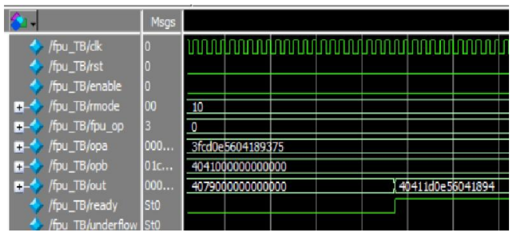
Multiplication Module

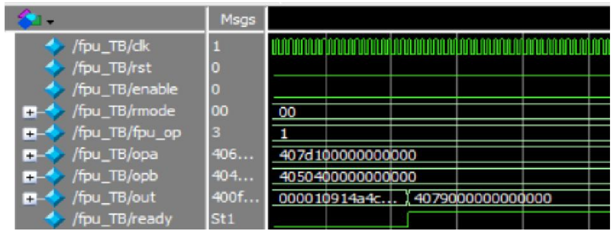
The floating-point multiplication operation is performed through a sequence of exponent and mantissa manipulations. Initially, both operands are decomposed into sign, exponent, and mantissa. The sign of the result is determined by XOR of the operand signs. Exponents are added and bias (1023) is subtracted.

Division Module

The division module employs an iterative *restoring division algorithm*. The divisor mantissa is compared and subtracted from the dividend mantissa bit by bit to form the quotient. Each iteration shifts the partial remainder left and appends one quotient bit per clock cycle, for a total of 54 iterations. The result is normalized, the exponent adjusted, and rounding is applied to produce a 64-bit IEEE-754-compliant output.

Result


Msgs
/fpu_TB/clk 0
/fpu_TB/rst 0
/fpu_TB/enable 0
/fpu_TB/rmode 00 10
/fpu_TB/fpu_op 3 0
/fpu_TB/opa 000... 3fcd0e5604189375
/fpu_TB/opb 01c... 4041000000000000
/fpu_TB/out 000... 4079000000000000 40411d0e56041894
/fpu_TB/ready St0
/fpu_TB/underflow St0
Fig 2:Floating point Addition operation


Msgs
/fpu_TB/clk 1
/fpu_TB/rst 0
/fpu_TB/enable 0
/fpu_TB/rmode 00 00
/fpu_TB/fpu_op 3 1
/fpu_TB/opa 406... 407d100000000000
/fpu_TB/opb 404... 4050400000000000
/fpu_TB/out 400f... 000010914a4c... 4079000000000000
/fpu_TB/ready St1
Fig 3:Floating point Subtraction operation

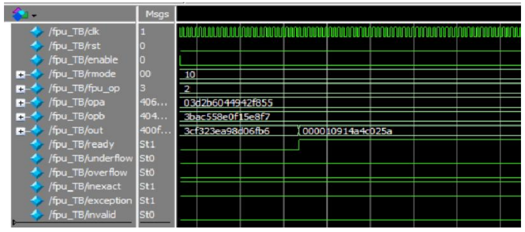


Fig 4: *Floating point Subtraction operation*

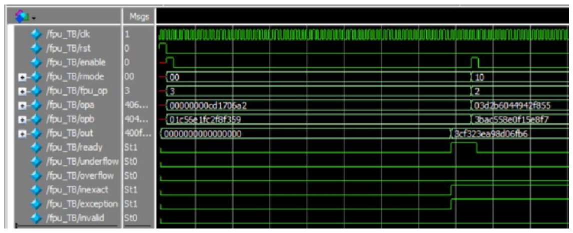


Fig 5: Floating point division operation

VI. CONCLUSIONS

This project presented the complete design and FPGA implementation of a Double- Precision Floating-Point Unit (FPU) using Verilog HDL. The work encompassed algorithmic study of IEEE-754 arithmetic, structural RTL development, functional simulation, synthesis, timing analysis, and on-board verification on the Terasic DE10-Standard SoC (Intel Cyclone V 5CSXFC6D6F31C6) platform. The design successfully integrates the four fundamental arithmetic operations—addition, subtraction, multiplication, and division—along with rounding and exception handling as prescribed by the IEEE-754 double-precision standard.

REFERENCES

[1]. F. de Dinechin and B. Pasca, "Custom Arithmetic Datapath Design Using the FloPoCo Core Generator," *Proc. 19th Int. Conf. on Field Programmable Logic and Applications (FPL)*, IEEE, pp. 170-175, 2009. DOI: 10.1109/FPL.2009.5272553.

[2]. X. Wang and M. Leeser, "VFloat: A Variable Precision Fixed- and Floating-Point Library for Reconfigurable Hardware," *ACM Trans. on Reconfigurable Technology and Systems (TRETS)*, vol. 3, no. 3, pp. 1-34, 2010. DOI: 10.1145/1839480.1839486.

[3]. B. Pasca, "Correctly Rounded Floating-Point Division for DSP-Enabled FPGAs," *Proc. Int. Conf. on Field Programmable Logic and Applications (FPL)*, IEEE, pp. 223-228, 2012. DOI: 10.1109/FPL.2012.6339189.

[4]. S. Liebig and D. Koch, "PolyGSopt: High-Performance Double-Precision Division on FPGAs," *Proc. Int. Conf. on Field-Programmable Technology (FPT)*, IEEE, pp. 284-287, 2014. DOI: 10.1109/FPT.2014.7082762.

[5]. M. Jaiswal and H. So, "An Area-Efficient Architecture for Dual-Mode Double- Precision Floating-Point Division," *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 64, no. 3, pp. 665-676, Mar. 2017. DOI: 10.1109/TCSI.2016.2607227.

[6]. IEEE Standard for Floating-Point Arithmetic, IEEE Std 754-2008, Aug. 2008

[7]. [Intel Corporation, Cyclone V Device Handbook, Vol. 1: Device Overview andDatasheet, 2020.

[8]. Terasic Technologies Inc., DE10-Standard Development and Education BoardReferenceManual, Rev. 1.3, 2019.

[9]. Intel Corporation, Quartus Prime Lite Edition 20.1 User Guide: Design Compilation2020.