IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 7, May 2023

An Efficient VLSI Design of Pipelined Half Precision Floating Point ALU Design

[1]Dr. S. A. Arunmozhi, [2]S. BharathHari, [2]S. Hari Ganesh, [2]K. Kaviya, [2]M. P. Kaviya
[1]Associate Professor, Department of Electronics and Communication Engineering
[2]UG Students, Department of Electronics and Communication Engineering
Saranathan College of Engineering College, Panjappur, Trichy, Tamil Nadu, India
Affiliated to Anna University

Abstract: *The IEEE Standard 754 floating point number is the most popular format for real numbers in modern computers. This work provides an overview of IEEE floating point and its representation is provided in this work. Using half-precision arithmetic in Verilog for VLSI design involves translating data models into half-precision floating-point number and creating the appropriate arithmetic operations. The IEEE 754 standard defines the structure of half-precision floating-point number consisting of 16 bits divided into three parts: a sign bit, a five-bit exponent, and a ten-bit mantissa. The sign bit represents the sign of the number, the exponent represents the magnitude of the number, and the signific and or mantissa represents the precision of the number. To use half-precision arithmetic in Verilog, you must create a module that defines the half-precision floating-point number data structure and uses the arithmetic operators for addition, subtraction, multiplication, and square. This pipeline, described on Verilog, is built on a Xilinx Spartan 3 FPGA. Xilinx Timing Analyzer was used measure the runtime. Also, when compared to cutting-edge technology, our offering outperforms them in terms of latency and throughput. ALU Design is known by Proposed Arithmetic Operation*

Keywords: ALU Design

I. INTRODUCTION

To the complexity of the algorithms,floating point operations are very difficult to implement in FPGAs.Although floating point operations encompass computations with a wide dynamic range, they need more resources than integer operations. Unsigned/signed multipliers can be used to multiply binary data, whereas floating point division can be used to multiply floating point numbers. Binary interchange format and Decimal interchange format are the two unique floating-point formats supported by the IEEE 754 standard. The binary representation of real values can be expressed using floating point numbers, for example. Divide floating point values is crucial for DSP applications with high dynamic range.

II. RELATED WORK

A low-power, high-radix floating-point divider architecture is suggested. The divider has certain drawbacks, including the ability to estimate partial quotient digits and inaccuracies in each cycle of repetition[2].On the basis of the Taylor-series expansion method, a fused floating-point multiply, divide, and square root unit is developed. Taylor Series implementation has the drawback of being complex[1] The hardware implementations employ single precision 32-bit floating-point technology. Division and the multiplicative inverse are both used in the implementations. Processing speed is a drawback that is less[5] by adding a corrective phrase that can be searched up during the early iteration in place of the last iteration. The greater error value is a drawback[4] modified booth multipliers are used in exponential Portion Disadvantage is Its having higher Delay[3]. Compared to previous methods, our concept provides significant improvements in terms of latency and throughput.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 7, May 2023

III. IEEE 754 FLOATING POINT NUMBERS REPRESENTATION:

IEEE 754 is a standard that specifies how floating-point numbers are represented in binary format and how arithmetic operations on these numbers are performed. The standard defines several formats for representing floating-point numbers, including single-precision (32-bit) and double-precision (64-bit) formats.

3.1.To convert a decimal number to its IEEE 754 binary representation, follow these steps:

1. Convert the decimal number to binary.
2. Determine the sign bit. If the decimal number is positive, the sign bit is 0. If it is negative, the sign bit is 1.
3. Normalize the binary number by shifting the binary point to the left until the most significant bit is 1. Count the number of bits shifted, which is the exponent.
4. Add the bias to the exponent. The bias for single-precision format is 127, and for double-precision format is 1023.
5. Convert the exponent to binary and pad with zeros to the left if necessary to match the size of the exponent field in the format.
6. Combine the sign bit, exponent field, and normalized fraction (mantissa) into a single binary number.
7. For single-precision format, the resulting binary number should be 32 bits long. For double-precision format, it should be 64 bits long.

3.2.To convert an IEEE 754 binary representation to its decimal equivalent, follow these steps:

1. Determine the sign of the number based on the leftmost (most significant) bit.
2. Determine the exponent by subtracting the bias from the value of the exponent field.
3. Convert the mantissa to decimal by summing the product of each bit in the mantissa with its corresponding power of 2, starting with 1/2 and doubling each time. The sum is multiplied by 2 to the power of the exponent.
4. Apply the sign to the result obtained in step 3 to obtain the final decimal value.
5. The special values, such as infinity and NaN (not a number), are also defined by the IEEE 754 standard and have specific representations in the binary format.

IV. SOFTWARE REQUIREMENT

4.1.SIMULATION TOOL

MODELSIM 6.4C:

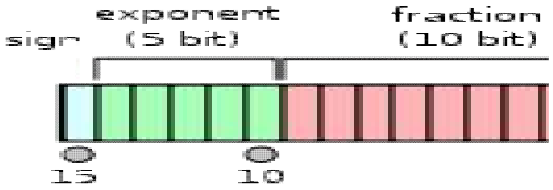
Modelsim is a simulation tool for hardware design which provides behavioral simulation of several languages, i.e., Verilog, VHDL, and System C. Verilog HDL is an industry standard.

4.2.SYNTHESIS TOOL:

XILINX 9.1/13.2:

Xilinx Tools are integrated tools for designing digital circuits using Xilinx Field Programmable Gate Arrays (FPGAs). Digital design can be entered in different ways using the CAD tools mentioned above: using the schematic input tool, u sing hardware description language (HDL) -Verilog or VHDL, or both.

V. HALF PRECISION FORMAT

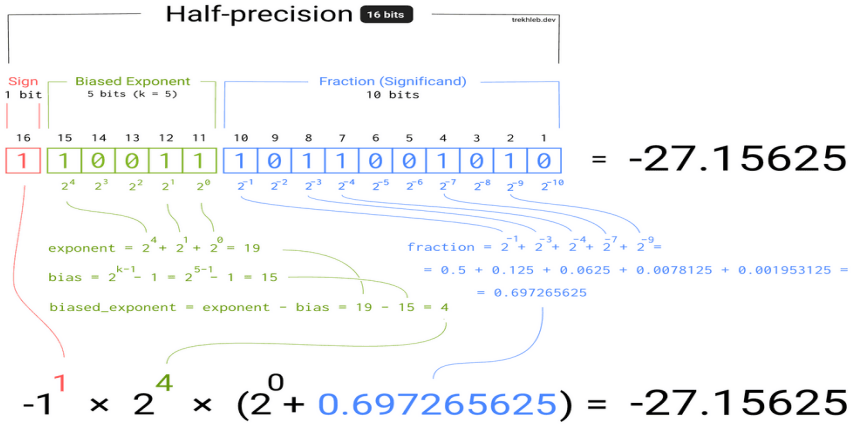


Half precision is a data format used to represent numerical values using a smaller amount of memory compared to other formats, such as single precision or double precision. It is also known as binary16 format, and it uses 16 bits of data to store each value.In half precision format, the 16 bits are divided into three components: 1 sign bit, 5 exponent bits, and

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 7, May 2023

10 mantissa bits. The sign bit determines whether the number is positive or negative, while the exponent bits represent the exponent of the number in a biased notation. The mantissa bits represent the significand or the fractional part of the number.Half precision format offers a compromise between memory usage and precision. While it provides less precision compared to other formats, it can represent a wide range of values and is suitable for many applications where memory usage is a critical factor, such as mobile devices or real-time graphics.One of the main advantages of half precision format isits compactness, as it requires only half the amount of memory compared to single precision format, which uses 32 bits of data. This makes it ideal for applications where memory usage is limited, such as embedded systems or mobile devices.However, it is important to note that half precision format has limitations, and it may not be suitable for all applications. For example, it may not provide enough precision for scientific calculations or simulations that require high accuracy. Nonetheless, half precision format is widely used in many applications, and it has become a popular choice for representing numerical values in computer graphics and gaming.

5.1NUMBER REPRESENTATION IN HALF PRECISION:



Floating point numbers are used to represent non-integer numbers and are used in most engineering calculations such as 3.256, 2.1, 0.0036. The most commonly used floating point standard is the IEEE standard.Semi-precision data types use less memory than other floating-point types, such as single and double precision. Although it only occupies 16 bits of memory, it allows floating point to be wider than integers or constant data of the same size. We use "pipelining" to increase the efficiency of half of the arithmetic operations.

VI. EXISTING SYSTEM

Conventional half-precision number refers to a format of representing numerical values using 16 bits of data. It is also known as binary16 format, and it is commonly used in computer graphics and other applications that require fast processing of numerical data. In this format, the 16 bits of data are divided into three components: 1 sign bit, 5 exponent bits, and 10 mantissa bits. The sign bit indicates whether the value is positive or negative, while the exponent bits represent the exponent of the number in a biased notation. The mantissa bits represent the significand or the fractional part of the number. The range of values that can be represented in half-precision format is limited compared to other formats, and the precision is also lower. The largest positive number that can be represented is 65504, and the smallest positive number that can be represented is 6.1035 x 10^-5. The precision is approximately 3 decimal digits.
Despite its limitations, half-precision format is useful in applications where memory and processing speed are important, such as mobile devices and real-time graphics.

6.1.EXISTING SYSTEM TECHNIQUE:

Conventional Half Precision Method

6.2.DRAWBACK:

It has more delay and Power

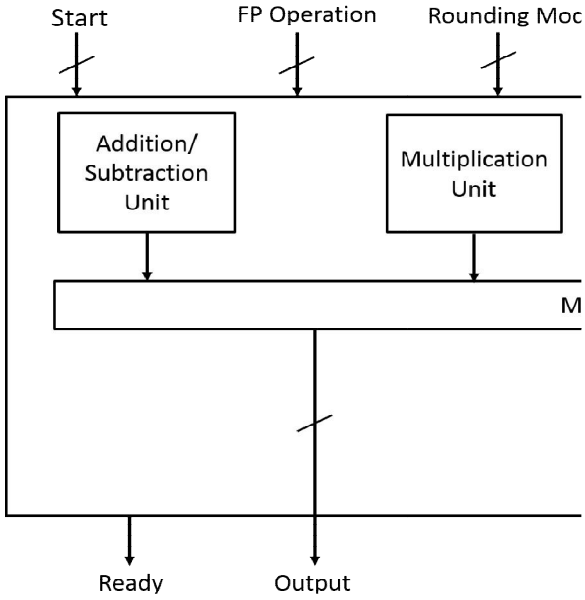
IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 7, May 2023

VII. PROPOSED SYSTEM:

The Arithmetic Logic Unit is one of the main components in the ALU computer processor. Performs arithmetic operations such as addition, subtraction, multiplication and division. The pipeline allows multiple instructions to be executed simultaneously. Pipelined ALUs have better performance, measured in terms of the number of clock cycles required to perform each arithmetic operation. Floating point notation is based on IEEE Std 754.In this paper, it is proposed that the ALU pipeline performs four arithmetic operations in an HDL environment, including addition, subtraction, multiplication, and division. Some fields, such as machine learning and graphics processing, can benefit from using partial precision. The motivation for this work will arise from the need to develop custom applications by creating a custom ALU to meet their needs. The ALU can better support these applications by providing suitable floating point operations at half precision.

7.1.PROPOSED ALGORITHM

Pipelined Half Precision Floating Point Designs



The floating-point adder in the ALU performs the addition of two half-precision floating-point numbers, taking into account the sign, exponent, and mantissa of both numbers. It follows the rules specified in the IEEE 754 standard for handling underflow, overflow, and rounding. Similarly, the floating-point subtractor subtracts one half-precision floating-point number from another, considering the necessary adjustments for signs, exponents, and mantissas. The floating-point multiplier in the ALU performs the multiplication of two half-precision floating-point numbers. It multiplies the mantissas and adds the exponents, adjusting the result based on the sign of the numbers being multiplied. The square operation calculates the square of a half-precision floating-point number. It squares the mantissa and doubles the exponent, accounting for the sign of the input. Additionally, the ALU includes logic operations such as AND, OR, and XOR, which operate on the individual bits of the input numbers. The Verilog implementation of the ALU utilizes the structural design approach, where the ALU module is divided into smaller sub-modules responsible for specific operations. These sub-modules are interconnected to form the complete ALU. The ALU design is intended to provide efficient and accurate computation for half-precision floating-point numbers, enabling various mathematical and logical operations in digital systems.

7.2.PROPOSED SYSTEM ADVANTAGES:

Low-Area Design
Low-energy high-throughput hardware
Less Complexity

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 7, May 2023

VIII. RESULT

The Fig.8.1 shows that the results of arithmetic operations(Addition, Subtraction, Multiplication, Square of A, Square of B) and logical operations (AND, OR, NOT, NAND, NOR, XOR, XNOR)computed for taking values of A and B as 16.0 and 25.0 respectively.

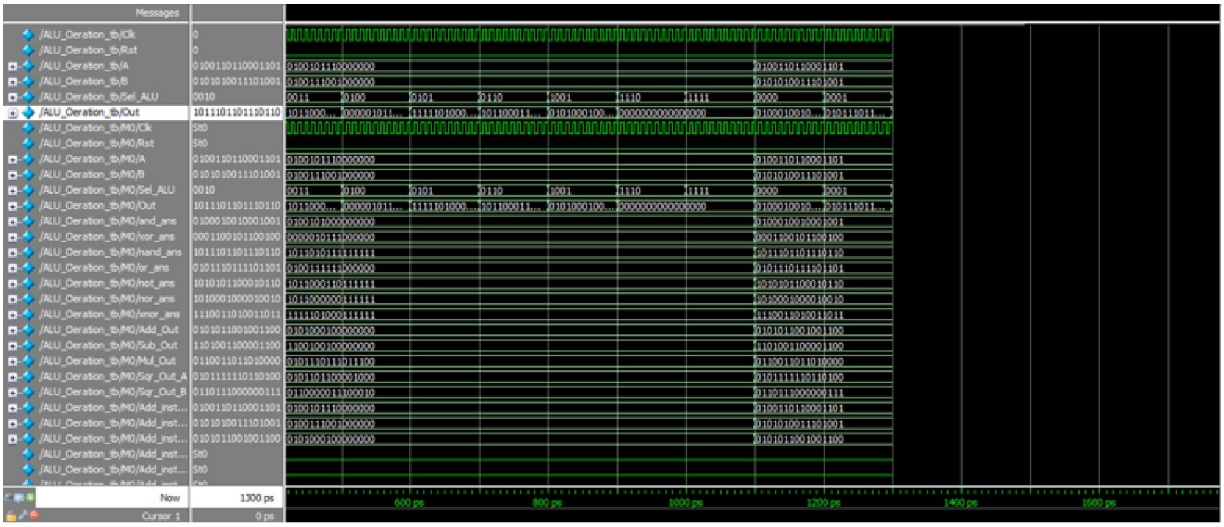


Fig.8.1

IX. CONCLUSION

Using a modern Pipelined approach and a Half Precision Arithmetic unit we offer a revolutionary division method in this article. A single-precision divider with a fully pipelined design is implemented using this technique. We introduce all the Arithmetic Unit and ALU Design. To cut down on the quantity of incomplete products is used in the hardware implementation. A compressor also combines the partial products' sums and other inputs before computing it. When compared to previous techniques, our concept provides significant improvements in terms of latency and throughput.

REFERENCES

[1] S. Galal and M. Horowitz, “Energy-efficient floating-point unit design,” IEEE Trans. Comput., vol. 60, no. 7, pp. 913–922, Jul. 2011.
[2] Y. Yang, Q. Yuan, and J. Liu, “An architecture of area-effective high radix floating-point divider with low-power consumption,” IEEE Access, vol. 9, pp. 40039–40048, 2021.
[3] P. Surapong and F. A. Samman, “Floating-point division operator based on CORDIC algorithm,” ECTI Trans. Comput. Inf. Technol. (ECTI-CIT), vol. 7, no. 1, pp. 79–87, Jan. 1970.
[4] K.-N. Han, A. F. Tenca, and D. Tran, “High-speed floating-point divider with reduced area,” Proc. SPIE Math. Signal Inf. Process., vol. 7444, Oct. 2009, Art. no. 74440O.
[5] P. Malik, “High throughput floating-point dividers implemented in FPGA,” in Proc. IEEE 18th Int. Symp. Design Diag. Electron. Circuits Syst., Apr. 2015, pp. 291–294.
[6] P. Kornerup, “Reviewing high-radix signed-digit adders,” IEEE Trans. Comput., vol. 64, no. 5, pp. 1502–1505, May 2015.
[7] K. Papachatzopoulos and V. Paliouras, “Reduction of delay variations in arithmetic circuits using a redundant representation,” in Proc. 5th Int. Conf. Modern Circuits Syst. Technol. (MOCAST), Thessaloniki, Greece, 2016, pp.1–4
[8] Y. S. Mehrabani and M. Eshghi, “Noise and process variation tolerant, low-power, high-speed, and low-energy full adders in CNFET technology,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 24, no. 11, pp. 3268–3281, Nov. 2016.
[9] M. Alioto and G. Palumbo, “High-speed/low-power mixed full adder chains: Analysis and comparison versus technology,” in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), New Orleans, LA, USA, 2007, pp. 2998–3001.