RESEARCH ARTICLE OPEN ACCESS

AN EFFICIENT VLSI DESIGN OF PIPELINED FLOATING POINT ALU FOR PROCESSORS

THIRUMUTHY.N*, Dr. V.Kalaipoonguzhali,**

*(PG Scholar VLSI/Sembodai Rukmani Varatharajan Engineering College, and Sembodai, Email:thiru881@gmail.com)

**(Professor and HOD of ECE, /Sembodai Rukmani Varatharajan Engineering College, and Sembodai, Email: hodeeesrvec@gmail.com)

----------**********----------

Abstract:

IEEE Standard 754 floating point is the most prevalent format today for real numbers on computers. An overview of IEEE floating point and its representation is provided in this publication. In order to improve time and area performance, this study describes a single precision floating point divider. The main goal of this work is to create a specific method for splitting two floating point integers in order to decrease power consumption and boost execution speed. In evaluation to earlier Dividers, this one is the quicker and more exact because this is to the pipelining process. This Verilog-described pipelined design is built on a Xilinx Spartan 3 FPGA. Xilinx Timing Analyzer is used to assess timing performance. To reduce the critical path, a compressor and an adder also compute the total of the partial products and other data. The combined findings demonstrate that our design's highest attainable frequency is superior to that of the current approaches. Moreover, when compared to existing techniques, our concept clearly outperforms them in terms of latency and throughput. ALU (Arithmetic Logic Unit) is Implemented by Proposed Arithmetic Process.

Keywords —Encryption,Decryption,FPGA,ALU.

----------**********----------

I. INTRODUCTION

Floating point operations are exceedingly challenging to implement on an FPGA due to the intricacy of the algorithms. Large dynamic range computations are included in floating point operations, yet these operations take more resources than integer operations do. Binary values can be multiplied using an unsigned/signed multiplier, while floating point numbers may be multiplied using floating point division. The IEEE 754 standard offers two distinct floating point formats: Binary interchange format and Decimal interchange format. Floating point numbers are one possible approach to express real values in binary format. For DSP applications with high dynamic range, the ability to divide floating point values is essential.

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.

II. LITERATURE SURVEY-1:

Title: Architecture of area effective high radix floating-point divider with low power consumption

Author: Y. Yang, Q. Yuan, and J. Liu

Year: 2021

Description: 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.

III. LITERATURE SURVEY-2:

Title: Floating-point division and square root using a Taylor-series expansion algorithm.

Author: T.J. Kwon and J. Draper

Year: 2009

Description: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 calculations.

IV. LITERATURE SURVEY -3:

Title: High throughput floating-point dividers implemented in FPGA

Author: P. Malik

Year: 2015

Description: 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.

V. EXISTING SYSTEM:

- DES is now considered to be insecure for several applications.
- This is chiefly due to the 56-bit key size being too small to DES keys have been broken.
- There are also some analytical results which determine theoretical weaknesses in the cipher, although they are unfeasible to mount in practice.

VI. EXISTING SYSTEM DRABACKS:

- The 56 bit key size is the major defect of DES and the chips to implement one million of DES encrypt or decrypt operations.
- Hardware operations of DES are very quick.
- DES was not designed for application and therefore it runs comparatively slowly.

VII. PROPOSED SYSTEM:

Floating point number is a numerical representation used in computer programming to store and manipulate real numbers. It is called "floating point" because the decimal point can float to different positions in the number, allowing for a wide range of values to be represented with a relatively small number of bits.

In most computer systems, floating point numbers are represented using the IEEE 754 standard. This standard specifies the format for representing floating point numbers, including the number of bits used for the mantissa (the significant digits of the number) and the exponent (which determines the scale of the number).

Floating point numbers are useful for a wide range of applications, including scientific calculations, financial modeling, and graphics processing. However, they have limitations, such as limited precision and the potential for rounding errors. Therefore, it's important to be aware of these limitations and use appropriate techniques when working with floating point numbers.

VIII. PROPOSED SYSTEM ADVANTAGES:

- Low-power Design
- Low-energy high-throughput

IX. SYSTEM ARCHITECTURE:

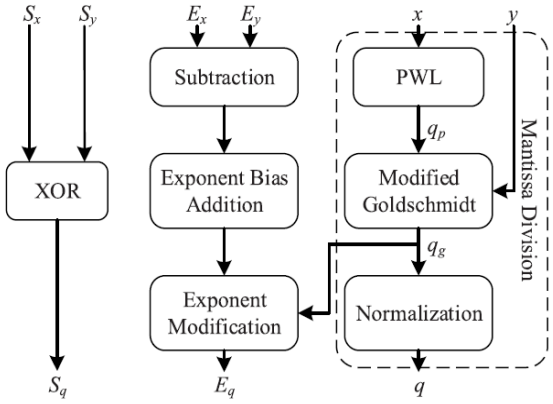

Fig. 1System Architecture

X. CONCLUSIONS

Using a modern universal PWL approach and a modified Goldschmidt algorithm, 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, booth encoding 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] [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] [2] 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.
[3] [3] 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.
[4] [4] P. Malik, "High throughput floating-point dividers implemented in FPGA," in Proc. IEEE 18th Int. Symp. Design Diag. Electron. Circuits
[5] Syst., Apr. 2015, pp. 291–294.
[5] 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.

