International Journal for Interdisciplinary Sciences and Engineering Applications

Orange Publications

IJISEA

International Peer-Reviewed Journal

www.ijisea.org

ISSN:2582 - 6379

Contact : editor@ijisea.org

ISSN: 2582 - 6379
Orange Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2024, May, Volume 5 Issue 1
www.ijisea.org

Design and Implementation of 32-bit ALU with 32 Operations for FPGA-Based RISC Architecture Applications

Chukka Sameena[1], CH Sri Lakshmi[2]
PG Schalor[1], Assistant Professor[2]
Department of Electronics and Communication Engineering
Amrita Sai Institute of Science and Technology
Kanchikacherla,NTR District,AP,India

ABSTRACT

The project focuses on the design and the implantation of a 32-bit Arithmetic Logic Unit (ALU) capable of performing 32 different arithmetic and logic operations based on select lines. The ALU is implemented using Verilog hardware description language (HDL) and synthesized on FPGA hardware. Each operation is selected using a 5-bit control input, enabling a wide range of arithmetic and logical computations. Primary objective is to showcase the ALU's functionality and versatility across diverse computing tasks. To optimize power consumption and hardware utilization, the design incorporates clock gating techniques and pipelining, ensuring efficient operation in resource constrained environments. Pipelining enhances throughput by breakdown operations into sequential stages, maximizing alu utilization. Through extensive simulation and synthesis, the ALU design is rigorously validated demonstrating both correctness and efficiency in operation. Vivado was utilized for a synthesis and implementation process, ensuring compatibility with FPGA hardware. The ALU is synthesized on Artix 7. FPGA platform, containing 61 cells, 135 i/o port and 1005 nets. Its design to carry out various arithmetic and logic tasks with each operation chosen using a 5-bit control input.

Keywords: Arithmetic Logic Unit, Verilog HDL, FPGA, Digital Design,Arithmetic Operations, logic Operations, Clock Gating and Pipelining

I.INTRODUCTION

The designed ALU module accepts two 32-bit operands (A and B), along with the control signal for selecting the desired operation. It supports 32 different operations, including arithmetic operations (addition, subtraction, multiplication, division, etc), bitwise operations (AND, OR, XOR), logical operations (NOT, NAND, NOR), shift operations (left shift, right shift), Additionally, specialized functions such as 2's complement, ADC (Add with Carry), parity checker and generation, conversion between Binary and Grey codes and Manchester encoding are also supported. The ALU operates on a positive edge-triggered clock and delivers the computed results on a 64-bit output bus, ensuring compatibility with a wider range of digital computing system.

II. LITERATURE SURVEY

The literature review for the 32-bit Arithmetic Logic Unit (ALU) design project encompasses several key areas relevant to digital logic circuitry and computer architecture. Firstly, a thorough investigation of ALU design techniques reveals the desire methodologies and architectural approaches used in constructing efficient ALUs. This includes the exploration of traditional architecture like ripple-carry adders and carry – look ahead adders as well as modern design incorporating pipelining and parallel processing for enhanced performance and throughput. Additionally, an examination of fundamental digital logic circuits, such as logic gate and flip flops sheds lights on the underlying components utilized within ALU architectures. Combinational and sequential logic circuits are studied for their integration into ALU designs, contributing to overall functionality and efficiency. Within the context of computer architecture, the review delves into the role of the ALU within processor system and its interaction with other components like registers, control units and memory units. This includes an analysis of instruction set architectures (ISAs) and their influence on ALU design principles and capabilities. Furthermore, optimising techniques aimed at improving ALU performance, area efficiency and power consumption are exploded. This entails investigating design automation tools synthesis strategies and verification methodologies essential for the successful development and validation of ALU designs. Recent advances and research trends in ALU design constitute another critical aspect of the literature review, highlighting novel approaches and emerging technologies shaping the further of ALU architectures. By examining recent research papers and academic publications, the project team gains insights into key challenges and further directions in ALU 3 research, including, scalability reliability and support for specialized computation tasks. Moreover, the review encompasses an analysis of existing solution and implementations, including open-source projects and academic resources providing reference designs and code examples for educational research purposes. Through this comprehensive literature review, the project team is equipped with valuable knowledge and resource to inform their ALU design and implementation process effectively.

III. METHODOLOGY

The ALU executes 32 different arithmetic and logical operations based on the select line, including addition, subtraction, multiplication, division, bitwise and, or, xor, as well as shift and comparison operations. By decoding the select line, the alu dynamically switches between arithmetic and logical operations, providing versatility and efficiency in executing a wide range of competitions within processor architecture.

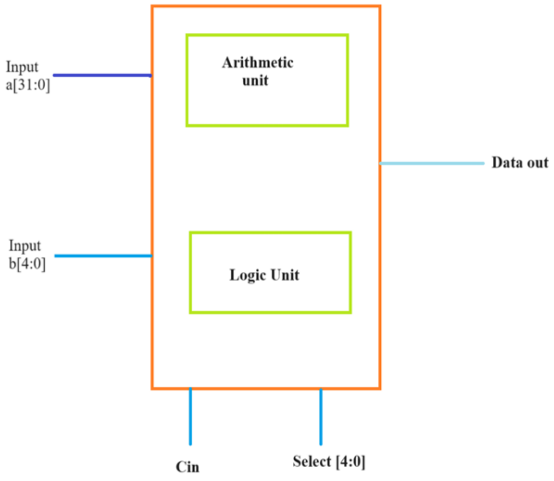


Figure 1: Shows Basic Block Diagram of ALU

The methodology employed for developing the 32-bit Arithmetic Logic Unit (ALU) involves a schematic approach, progressing through several key stages. Initially the project in conducts an in- depth analysis of

ALU requirements, encompassing supported operations, operand widths and architectural constraints. This analysis informs the creation of a detailed functional specification document, outlining the ALU intended behaviour and capabilities.

Following conceptualization phase, the design process commences with the definition of the ALU architecture. Factors such as operand width supported operation and data path organisation are carefully considered. Critical components like adder, multiplier and logic gates are then selected based on performance and efficiency criteria. Subsequently, Register Transfer Level (RTL) descriptions of the ALU module are developed using a hardware description language (HDL) like Verilog, capturing the functional behaviour and interconnection of the ALU components. Moving to implementation the RTL description are translated into synthesizable Verilog code, adhering to coding standards to ensure reliability and maintainability. Testbenches are developed to verify the correctness and functionality of ALU design through simulation-based validation. Various verification techniques including functional and formal verification, are employed to ensure the design meets specific requirements.

IV. RESULTS AND DISCUSSION

We carefully looked at how well Alu work by testing it under a different situation we made sure it worked accurately and reliably. We measured how much of the ALU’s abilities our tests checked, like making sure all parts of it were used. If we found any problem where the results did not match what we expected we looked into why that happened. Then we fix those problems quickly to keep it working well. The test results demonstrate that the implemented ALU met the specified performance requirements and exhibited reliable functionality across various test scenarios, validating its suitability for use in practical applications.

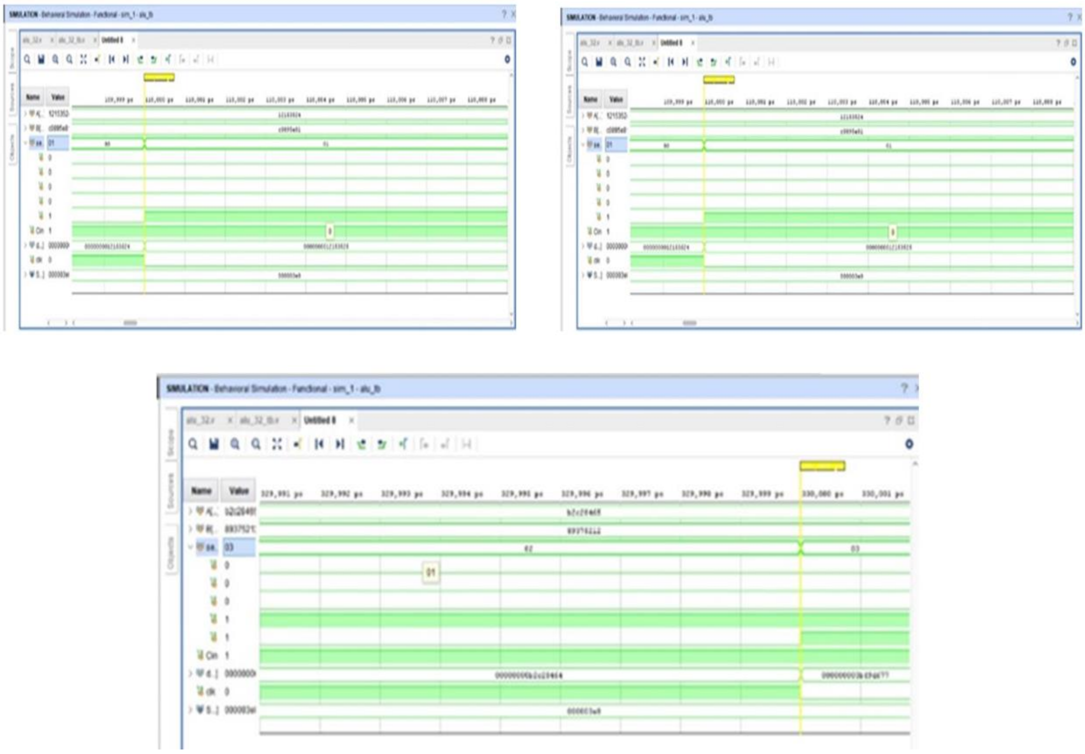

SIMULATION - Behavioral Simulation - Functional - sim_1 - alu_tb
SIMULATION - Behavioral Simulation - Functional - sim_1 - alu_tb
SIMULATION - Behavioral Simulation - Functional - sim_1 - alu_tb
Name
Value

Figure 2 : Shows Arithmetic operations of PASS,INC,DEC, like wise 32 Operations will done

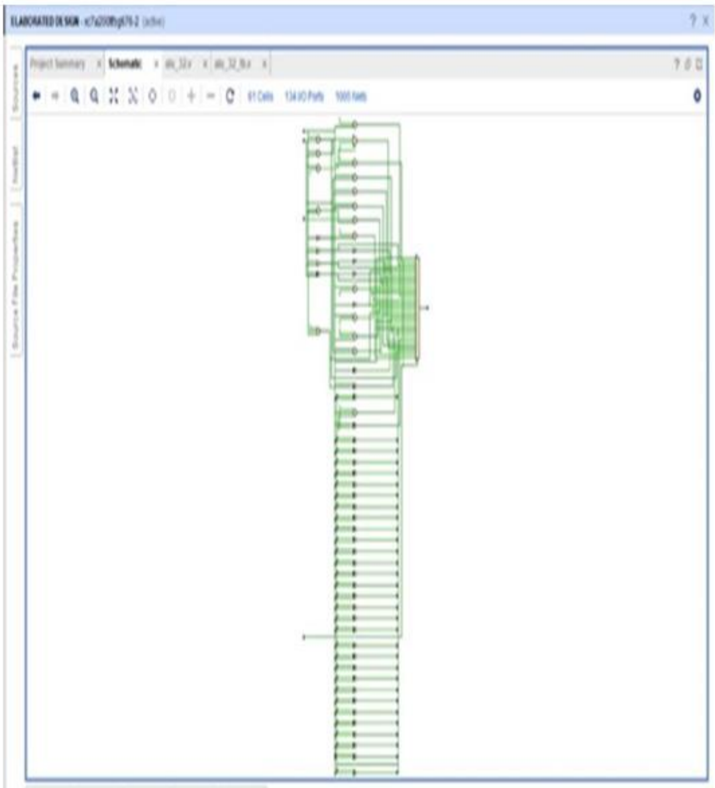
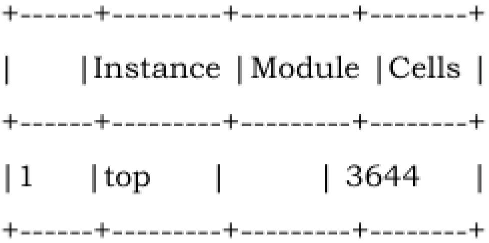


Figure 3: Shows Schematic of ALU

```
+------+------------+------+

|      | Cell       | Count |

+------+------------+------+

|1     |CARRY4  |   640 |

|2     |DSP48E1 |     4  |

|3     |LUT1       |   186 |

|4     |LUT2       |   226 |

|5     |LUT3       |  1992 |

|6     |LUT4       |    38  |

|7     |LUT5       |    58 |

|8     |LUT6       |   334|

|9     |MUXF7     |    32 |

|10    |IBUF       |    70 |

|11    |OBUF      |    64 |

+------+-----------+--  ----+
```

Figure 4 : Shows the Report of Cell Usage



```
+------+---------+---------+--------+
|      |Instance |Module   |Cells   |
+------+---------+---------+--------+
|1     |top      |         | 3644   |
+------+---------+---------+--------+
```

Figure 5:Shows usage Area Instances

V.CONCLUSION

In the conclusion, the development of 32-bit alu represents a significant achievement in advising computational capabilities within digital system. The project has successfully realized its objectives by designing and implementing a versatile and efficient alu capable of performing 32 different operations. Through rigorous testing and validation. The alu demonstrates reliability, accuracy and robustness in executing arithmetic and logical tasks. The project outcome underscores the importance of alu in digital computing, highlighting its potential for enhancing computational efficiency and performs in various domains. The Artix 7 FPGA platform's synthesised arithmetic Logic Unit (ALU) exhibits its ability to perform a wide variety of arithmetic and logic tasks with efficiency. Utilising a 5-bit control input, the ALU uses a design consisting of 61 cells, 135 I/O ports, and 1005 nets to dynamically select each operation. The adaptability and efficiency of FPGA-based hardware solutions in digital design and computation are demonstrated by this flexibility, which enables the smooth completion of a wide range of computational tasks.

Future Scope:

Moving forward, several areas of improvement and future works can be identified to enhance the alu functionality and performance. This includes refining the clock division and Manchester coding implementation to ensure full functionality across all supported operations. Additionally, optimization techniques can be exploded to improve the alu speed and efficiency making it more suitable for high performance computing applications. Furthermore, integrating additional features and expanding the alu capabilities could enhance its versatility and applicability in various computing domain.

REFERENCES:

[1] P. A. David and H. L. John “Computer Organisation and design” Morgan Kaufmann, 2005.

[2] D. Michel, A. Murali and S. Per, ‘Parallel Computer and Organisation Design’, Cambridge University Press, 2012.

[3] D. Neha, G. Nidhi M. Anu “Hardware Efficient AES for Image Processing with High Throughput”, IEEE conference on NGCT. 2015, Sept 2020, pp 932-935.

[4] Y. Shao-Ying, L. Yuan-Te, L. Wei-Chi, and H. Terng- Yin “Cost Efficient Frequency-Domain MIMO-OFDM Modern with a SIMD ALU-Based Architecture”, IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION. (VLSI) SYSTEM, VOL. 23, NO. 12, DECEMBER 2015. pp 2791-280.

[5] D. Michel, A. Murali and S. Per, ‘Parallel Computer and Organisation Design’, Cambridge University Press, 2012.

[6]D. Neha, G. Nidhi M. Anu “Hardware Efficient AES for Image Processing with High Throughput”, IEEE conference on NGCT. 2015, Sept 2015, pp 932-935.

[7] T. Yoshida, G. Matsuhara S. Yoshioka, H. Tago, S. Suzuki and N. Goto, “A 500MHz 1-stage 32bit ALU with self-running test circuit”, Symposium on VLSI Circuit Digest of Technical Paper, 1995, pp 11-12.

[8]G. Jeong and J. Park,” Design of 32-bit RISC Processor and efficient verification proceedings of the 7th Korea-Russian International Symphony KORUS, 2003, pp. 220-227.

[9] G. Jeong and J. Park and H. Jo,” ARM 7 compatible 32bit RISC processor design and verification” KORUS, 2005, pp. 761-764.

[10]Y. Kui and D. Yue-Hua, “32 Bit multiplication and division ALU design based on RISC structure”, International joint Conference on Artificial Intelligence, 2009, pp. 761-764

[11]P. Bhavina and S. Vandana, “To implement Cryptographic model for secure communication on FPGA using 32-bit ALU unit”, IEEE International Conference on Signal and Image Processing, pp. 440 443

IJISEA
www.ijisea.org

International Peer- Reviewed Journal