Arithmetic and Logic Unit Design Using VHDL Language

أحمد علي البرق[1]، سمية علي التومي[2]، جمال رمضان البرقلي[3]

كلية التقنية الصناعية - قسم الهندسة الإلكترونية - مصراتة- ليبيا
ahbarag@yahoo.com

Abstract :

This paper deals with designing a model for the Arithmetic and Logic Unit (ALU), which can be said as the basic building block of the Central Processing Unit (CPU) in a computer. A model of the ALU has been designed to perform a set of operations, which includes both arithmetic and logical operations, using the Very High Speed Integrated Circuit Hardware Description Language (VHDL), due to its wide spread and high level of accuracy in the world of digital design. Each hardware component has been designed and tested individually to ensure its functionality, and thus to ensure the functionality of the ALU as a single unit, before its real implementation. The results prove that each component correctly performs all the operations required of it, whether it is alone or within the ALU component.

الملخص:

تناول هذا البحث تصميم نموذج لوحدة الحساب والمنطق (ALU)، والتي يمكن القول بأنها اللبنة الأساسية لوحدة المعالجة المركزية (CPU) في جهاز الكمبيوتر. حيث تم التصميم لأداء مجموعة عمليات والتي تتضمن العمليات الحسابية والمنطقية على حد سواء، باستخدام لغة الوصف المادي VHDL، نظرا لما تمتاز به هذه اللغة من انتشار واسع ومستوى عالي من الدقة في عالم التصميم الرقمي. تم تنفيذ التصميم واختبار كل مكون مادي بمفرده للتأكد من صحة عمله، وبالتالي صحة عمل المكون العام ALU كوحدة واحدة قبل الشروع في تنفيذه بشكل حقيقي. وقد أثبتت النتائج صحة تنفيذ كل مكون لكافة العمليات المطلوبة منه بشكل صحيح، سواء أكان بمفرده أو ضمن المكون العام ALU.

Keywords: Digital design, Arithmetic and Logic Unit (ALU), Very High Description Language (VHDL), Central Processing Unit (CPU).

1. Introduction

An ALU is a digital circuit located inside the central processing unit, which is responsible for performing all arithmetic operations (such as addition, subtraction, division, and multiplication), knowing that the unit uses one operation, which is addition, while other operations such as subtraction are the process of equivalent addition, and multiplication is a recursive addition process, shifting one from right to left, while division is a sequential iterative subtraction process with successive reduction of positions. This circuit also implements logical operations such as (OR, AND, XOR) in digital computers, as well as performing comparison operations to find out the result of logical comparisons, which are (greater than, less than, equal to and not equal) and the derivatives of these comparisons, as well as providing the ability to store information temporarily. In addition to the possibility of processing information, it gives its output depending on a decision taken within it. Since all processing operations are limited to two types of operations, either they are arithmetic or they are logical, this unit is able to address any issue it is asked to address, and it can be said that this unit is the one that actually executes the instructions [1].

Many studies have dealt with the process of designing the arithmetic and logic unit, and its processing of many things, as in [2] the 8-bit arithmetic and logical unit that uses C5 technology were designed, where the design implements possible arithmetic and logical operations.

Also, [3] implemented an ALU using two different types of adders, a ripple carry adder, and a Sklansky type adder, based on Application Specific Integrated Circuits (ASIC).

Furthermore, a design for a low-power ALU were presented in [4] in which work was done to analyze the power flow from ALU circuits and work on optimizing it and optimizing the power supply unit of the ALU (power supply unit).

In view of the importance of this unit, it is necessary to identify it and the way it works, as well as the method of its design. A typical ALU was designed in this paper as follows: Arithmetic section design, Logic section design, and Combine two sections to form the desired ALU.

2. ALU Structure

It is a combinational logical circuit called the Arithmetic and Logic Unit (ALU). Figure (1) shows the general structure of this unit, its working principle, and composition and internal structure. Where A and B are the parameters, R is the output, F is the input from the control unit, and D is the output status.

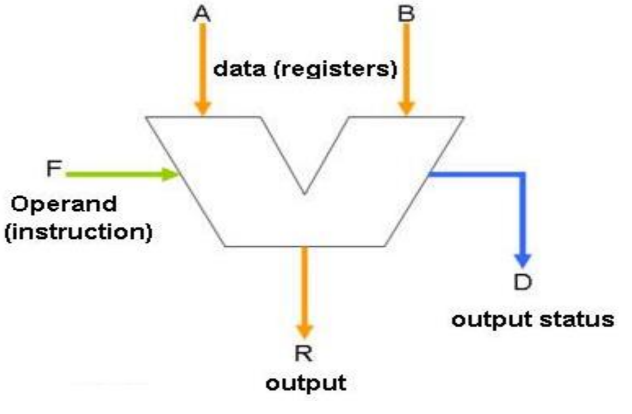

Figure 1. Arithmetic and logic unit symbol.

2.1 ALU Sections

An ALU circuits divided into two main parts as follows:

A. Integer Numbers Unit

This unit deals with the processing of arithmetic operations that consist of integer numbers that do not contain a decimal point. These operations are used in two-dimensional applications such as Word, Excel, Power point, two-dimensional drawing programs and most of the programs we use, meaning that this unit is used by two-dimensional applications, so it is very important because most of the software we use is based on this unit.

B. Floating Point Unit

This unit processes arithmetic operations that contain a decimal point, and this unit plays a major role in the speed of running programs that rely heavily on decimal numbers, such as three-dimensional games and engineering design programs such as AutoCAD. Nowadays this unit has come to play an important role, since modern games depend on this unit for their speed [5].

2.2 Working principle of ALU

The arithmetic and logic unit is the unit that actually executes the programming instructions, it is the one that adds, multiplies, divides, subtracts and performs all comparison operations. It receives commands from the Control Unit to execute the instruction stored in the Instruction Register, executes it, and then gives the result, which is usually stored in the main memory RAM, and it also uses the registers to complete its work.

2.3 Internal structure of ALU

The operations performed by the arithmetic and logic unit are related to the internal structure of this unit. In general, there are two basic types of building operational units:

- Complex Instruction Set Computer (CISC).
- Reduced Instruction Set Computer (RISC) [5].

2.4 Registers

The registers are a fast memory used in which the processor stores the numbers on which it wants to perform its calculations. The arithmetic and logic unit cannot perform any arithmetic operation until after the numbers to be performed are brought to the registers, as the size of the registers is important because it determines the size of data that the computer can perform calculations on it [1].

2.5 ALU Operations

ALU operations can be categorized into following:

A. Operations with single factor, such as:

- Clear: Clears a register content.
- Increment: increase content of a register one by one.
- Decrement: decreased a register content by one.
- Transfer: transferring contents of a register.
- Shifting: shifting a register content to the right or left.
- Logical NOT: change the logical level of a register content.

B. Operations with two factors, such as:

- Addition: Add the content of the Accumulator with other register content.
- Subtraction: Subtraction of a register content from the content of the Accumulator.
- Logical AND: Performs a logical multiplication between the accumulator content and a register and stores the result in the accumulator.
- Logical OR: The process of logical add between the accumulator content and a register and stores the result in the accumulator.
- Comparison: comparing the value of a register content with the value of the accumulator, but the result of the comparison is not stored in the accumulator, but the value is stored as one or zero depending on the comparison result in the comparison bit [6].

2.6 Inputs and Outputs

The inputs to the ALU are the data being operated, called coefficients, as well as the control unit code, which indicates the performance of the operation, and its output is the result of calculating these coefficients. In many ALU designs, the input or output can also take a set of symbols. These symbols are used to indicate states such as: carry in, carry out, overflow, divider by zero,….. etc.

3. VHDL Definition

Very High Description Language (VHDL) is a standard programming language designed by the United States Department of Defense that is used to describe, design, and simulate complex digital systems and VHSIC HDL circuits. It is an abbreviation of Very High Speed Integrated Circuit Hardware Description Language. It has been an IEEE standard since 1987 and was revised in 1997.

This language allows the description of parallelism and correlation in addition to the detailed and direct expression of time and also allows the description of analog circuits, but it is often used in the design of Field Programmable Gate Arrays (FPGA) circuits [7].

3.1 VHDL Design Styles

VHDL is characterized by flexibility and versatility in the ways of defining the architecture, and Figure (2) shows the types of design in VHDL:

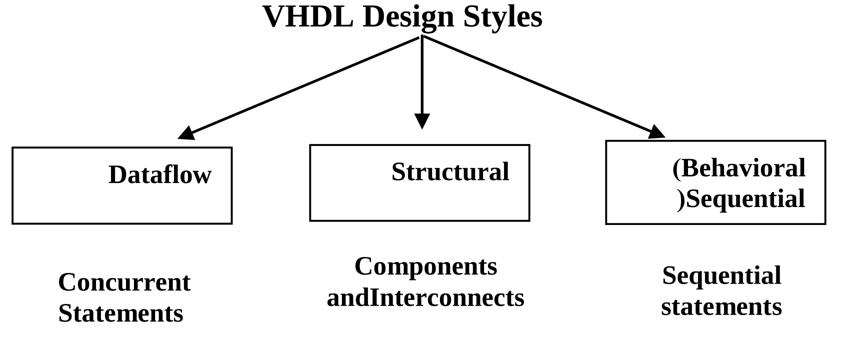

Figure 2. Design patterns VHDL.

A. Data flow

This pattern describes how data moves through a system and processing steps. The data flow method uses a series of synchronization statements to achieve the logic.

B. Structural

It can describe the structure of the design in terms of its secondary components and how they are interconnected, where the sub-components are created and built first and then called to build the system.

C. Behavioral Description

Behavioral modeling is used as a model of what happens to the inputs and outputs of a system diagram, regardless of what is inside the system and how it works, allowing the use of sequential data to describe the behavior of the system over time [8].

4. The Used Model

4.1 General Organization of ALU

The process of designing a unit that performs a set of arithmetic and logical operations by describing the hardware component to convert the description into small electronic circuits, then integrating these circuits and some auxiliary circuits such as multiplexers circuits to form the arithmetic and logic unit. Figure (3) shows a diagram of an 8-bit arithmetic and logic unit. It consists of an arithmetic unit and a logic unit, addition a multiplexer circuit and a shift circuit, which will be shown how each of them is designed separately and then combined to form the ALU.

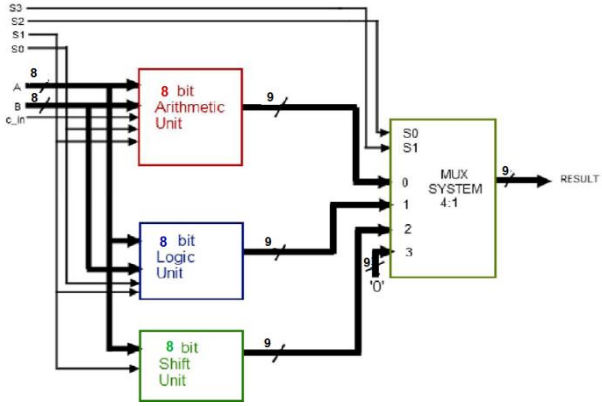

Figure 3. Schematic diagram of 8-bit arithmetic and logic unit.

1. 8- BIT ARITHMETIC UNIT

It is an arithmetic unit that performs the following tasks: Addition, Addition with carry, Subtraction, Decrement, Increment, and Transfer functions. Figure (4) shows the schematic diagram of the arithmetic unit in the arithmetic and logic unit and its component circuits.

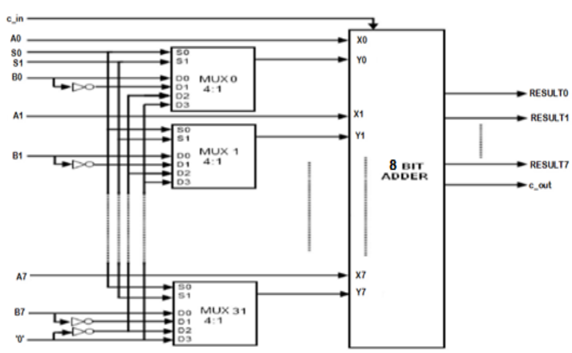

Figure 4. Schematic diagram of 8-bit arithmetic unit.

The basic unit of the arithmetic operation in ALU is the full adder unit. In this design, a full adder circuit will first be designed and used in the design of a 4-bit adder circuit, then a multiplexer circuit is formed and the circuits are then integrated to form the arithmetic unit.

2. 8 - BIT LOGIC UNIT

The logic unit in this design performs the following functions: Logical Multiplication, Logical Addition, Inversion or Complementation, and XOR Logical Operation.

The circuit that will be designed can perform the four basic logical operations: OR, NOT, XOR, AND, and from these operations all other logical operations can be derived. Figure (5) shows the schematic diagram of the logic unit in the ALU.

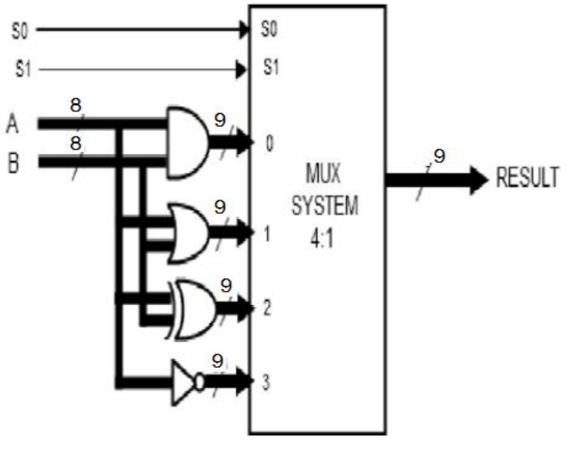

Figure 5. Schematic diagram for 8-bit logic.

Logic gates are the basic elements for the formation of simple logical circuits on which digital devices used in all communication systems and networks. The logic unit in this design consists of four logic gates, OR, NOT, XOR, AND, in addition to a Multiplexer circuit (MUX) to choose the required operation.

3. 8- BIT SHIFTER UNIT

The shift unit is used to perform the function of a logical shift for the values of a register. The transmission of bits in the register can be in two directions, right or left, and Figure (6) shows the schematic diagram of the 8-bit shift unit.

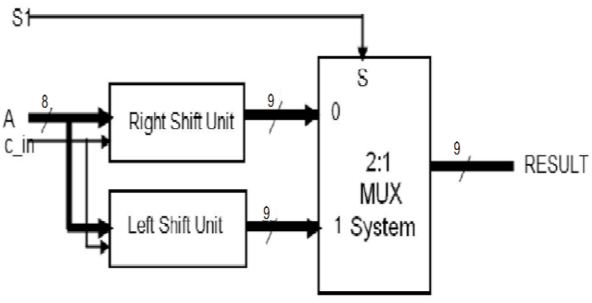

Figure 6. Schematic diagram of 8-bit shift unit.

In this design the shift unit is configured with one shift circuit to the right and one to the left and a circuit selector to choose which of the two circuits to operate [9].

5. Implementation and Results

After designing all the auxiliary units and circuits that make up the ALU, these components were combined to form an integrated circuit that can perform a set of arithmetic and logical operations between two different inputs of 8-bit length, which are (A0….A7) as the first input, and (B0….B7) as the second input. The function is selected and the type of operation performed by the circuit is determined by placing the operation code on the inputs (S0…S3) and the input C_in, and the output of the process is taken on the outputs (RESULT0……RESULT8). If there is a carry-out from a process, its value will be on the output RESULT8.

The program was simulated and tested and the results were presented, by tracking the behavior of the components within the arithmetic and logic unit, achieving its working theory and understanding how to deal with it, as the components were simulated separately, after which the ALU program was simulated as a single unit.

A. Arithmetic Unit

At this stage, the work of the arithmetic unit was tested and the correctness of its implementation of some calculations that will be performed on the inputs. For example, when the value of the control signal s=0 and the signal c_in=0, the output value (result) is the sum of the two inputs i_a and i_b , and since the values of the input and the control signals were in the Hexadecimal system, the result of the execution is as follows:

i_a $(40)_{16} = (01000000)_2$
i_b $(92)_{16} = (10010010)_2$
Result = i_a + i_b = $(0D2)_{16}=(011010010)_2$

Figure (7) shows the result of the arithmetic unit executing alone the process of summing the values on its inputs i_a and i_b.

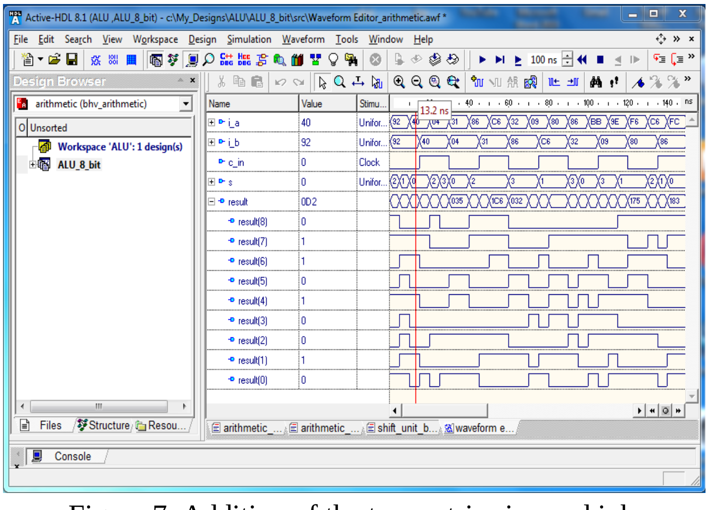

Figure 7. Addition of the two entries i_a and i_b.

B. Logic Unit

Figure (8) shows the execution of the logic unit program by performing the logical AND operation on the input ports i_a and i_b, where the values of the input and output ports after execution were as follows:

i_a $(3B)_{16} = (00111011)_2$
i_b $(F6)_{16} = (11110110)_2$
Result = i_a · i_b = $(Z32)_{16} = (Z00110010)_2$

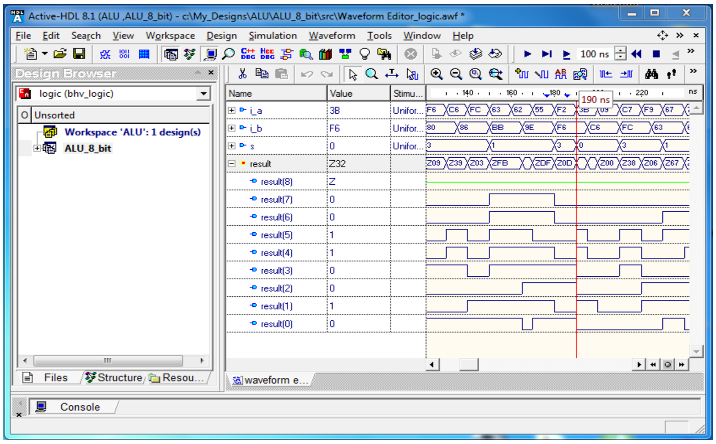

Figure 8. AND operation for input i_a and input i_b.

C. Shifter Unit

It is one of the circuits that make up the arithmetic and logic unit and works to shift the value of the port i_a to the right or to the left Shift, according to the value of the control signal s, which was its value in this process 0.

$i\_a = (86)_{16} = (10000110)_2$,$c\_in = 0$

$Result = (043)_{16} = (001000011)_2$

Figure (9) shows the result of the right-shifted value of port i_a, taking into account the value of port c_in.

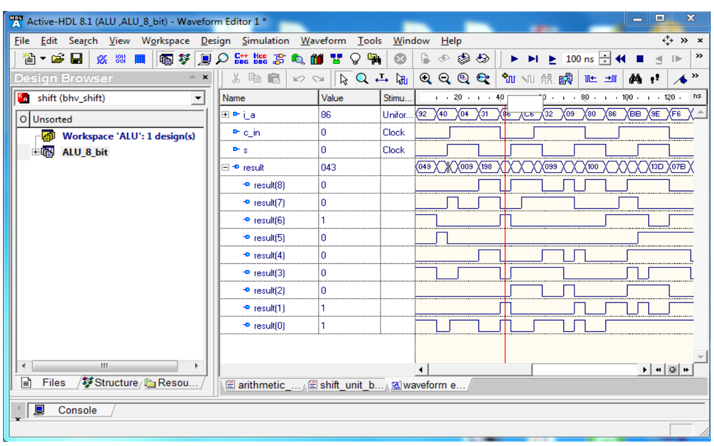

Figure 9. Shift right the input i_a.

D. Arithmetic and Logic Unit

After simulating the programs of the units comprising the arithmetic and logic unit individually, and displaying the results of some operations performed by these units individually, the ALU program consisting of the previous circuits together was simulated and testing the design and execution of the operations it performs, ensuring its behavior, displaying and testing results. Table (1) Describes the operations performed by the ALU component.

Table (1) Functions of the Arithmetic and Logic Unit

S3	S2	S1	S0	c_in	Result	Operation
0	0	0	0	0	A+B	Addition
0	0	0	0	1	A+B+1	Addition with carry
0	0	0	1	1	A+B'+1	Subtraction
0	0	1	0	0	A-1	Decrement
0	0	1	0	1	A	Transfer
0	0	1	1	0	A	Transfer
0	0	1	1	1	A+1	Increment
0	1	0	0	x	A·B	AND
0	1	0	1	x	A+B	OR
0	1	1	0	x	$A \oplus B$	XOR
0	1	1	1	x	NOT A	Complement
1	0	0	x	x	LSR A	Shift Right
1	0	1	x	x	LSA A	Shift Left

All cases have been tested and the results are valid, and Figure (10) shows the result of the execution of the addition with carry, this operation is performed on the values of the input ports A, B with the presence of a carry from a previous addition process.

i_a = $(85)_{16} = (10000101)_2$
i_b = $(09)_{16} = (00001001)_2$, c_in = 1
Result = $(08F)_{16} = (010001111)_2$

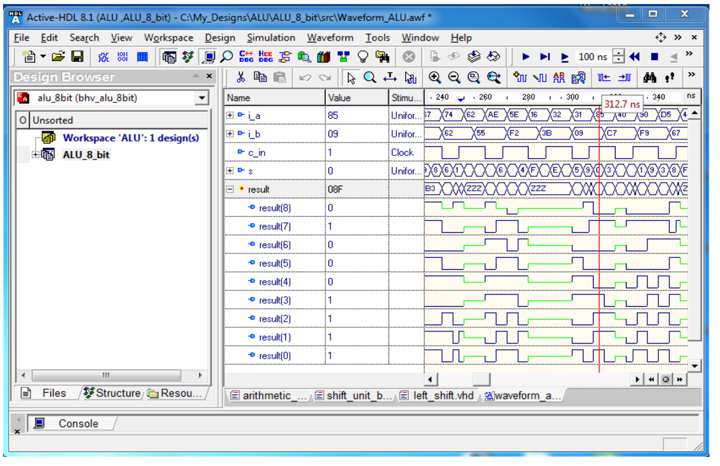

Figure 10. Addition with carry input.

As for the subtraction process, it is done using two's complement, which is the most common and used in computers, and Figure (11) shows the result of the process.

i_a $(61)_{16} = (01100001)_2$
i_b $(3B)_{16} = (00111011)_2$
i_b' $(C4)_{16} = (11000100)_2$
Result = i_a - i_b = i_a + i_b' +1= $(126)_{16}$= $(100100110)_2$

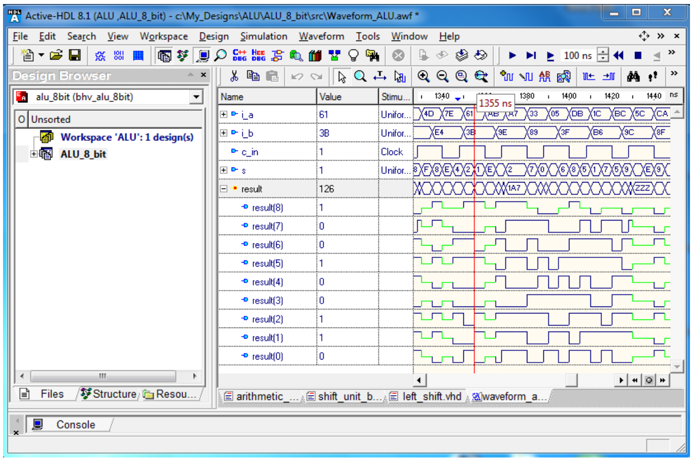

Figure 11. Subtraction using two's complement.

6. Conclusions

In this paper, electronic circuits and units that perform arithmetic and logical operations were built, and then these circuits were assembled and integrated to obtain an ALU component. The results of the implementation and simulation of the design proved to achieve the expected performance of the ALU model significantly, as each component gave the correct results for all the operations required of it, whether it was alone or within the general component ALU, and the possibility of simulating and testing each hardware component alone to ensure the correctness of its work, and thus the correctness of the work of the general component before actually starting to implement it.

7. References

[1] Muhammad T., Muhammad N., Fatin A., Muhamad Y.,AbdKadirM., Sharifah S., 2020, Design of Multiplication and Division Operation for 16 Bit Arithmetic Logic Unit (ALU). Journal of Electronic Voltage and Application Vol. 1 No. 2 (2020) 46-54.

[2] Priyal G., HemantV.,2015, Design, Layout Simulation of 8Bit Arithmetic and Logic Unit Using C5 Process Technology.

International Journal of Electrical and Electronics Engineers Vol. No. 7 Issue 2, July-December 2015.

[3] Abdul Rehman B.,Liguo S., Abdullah B., 2016, Comparative Analysis of ALU Implementation with RCA and Sklansky Adders In ASIC Design Flow.(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 7, 2016.

[4] ArindamB., AkanshaA., 2017, Design and Implementation of Low Power ALU Design. International Journal of Engineering and Computer Science Vol. 6 Issue 4, April 2017.

[5] Yi G., Shilang T., Zhongli D.,2000, Comparison between CISC and RISC.

[6] Shikha K., Kanika K., 2012, Implementation of ALU using FPGA. International Journal of Emerging Trends & Technology in Computer ScienceVol 1. Issue 2. July – August 2012.

[7] Abdelouahab B., Adel B., Farid D., Abdeslem S., 2018, FPGA based Hardware-in-the-Loop Simulation for Digital Control of Power Converters using VHDL-AMS. International Journal of Advanced Computer Science and Applications Vol. 9, No. 12, 2018.

[8] IEEE-SA Standards Board.,2000, IEEE Standard VHDL Language Reference Manual.

[9] DishaM., RichaR., 2013, 32-bit Arithmetic Logical Unit (ALU) using VHDL. International Journal of Science Engineering and Technology Vol.1 Issue 1, 26 NOV 2013.