**Proceedings of the**
**International Conference on Mechanical Engineering and Renewable Energy 2013**
**(ICMERE2013) 1 – 3 May 2014, Chittagong, Bangladesh**

**ICMERE2013-PI-263**

# DESIGN OF ARETHMETIC LOGIC UNIT USING VHDL

**Md. Rashedul Islam Sarker[1], Raju Ahamed[2], Md. Kamal Hossain[3] and Quazi Delwar Hossain[4]**

[1-4]Department of Electrical & Electronic Engineering, CUET, Bangladesh

[1]rashedul.eee08@gmail.com, [2]rajuahmedeee@gmail.com, [3]kamal_eee_cuet@yahoo.com, [4]quazi@cuet.ac.bd

***Abstract****- Arithmetic logic unit (ALU) is a digital circuit that performs arithmetic and logical operations and executes the commands accordingly. The ALU is a fundamental building block of the central processing unit (CPU) of a computer and the inputs to the ALU are the data to be operated on and a code from the control unit indicating which operation to perform. Its output is the result of the computation. These codes are used to indicate cases such as carry-in or carry-out, overflow, divide-by-zero, etc. We have proposed VHDL environment for ALU design and simulation is done for faster unit. This has further helped in easing the description, verification simulation and hardware realization. VHDL is widely adopted standard and has numerous capabilities that are suited for designs of this sort. The use of VHDL for modeling is especially appealing since it provides formal description of the system and allows the use of specific description styles to cover the different abstraction levels employed in this design. The synthesis of the proposed design is done for high speed and multiple executions to help in the management of large data size.*

**Keywords:** ALU, CPU, VHDL

## 1. INTRODUCTION

VHDL is a hardware description language. It is describe the behavior of an electronic circuit or system, from which the physical circuit or system can then be attained. VHDL stand for VHSIC Hardware Description Language. VHSIC is itself an abbreviation for Very Speed Integrated Circuit, an initiative funded by the United States Department of Defense in the 1980s that led to the creation of VHDL. Its first version was VHDL 87, later upgraded to the so-called VHDL 93. VHDL was original and first hardware description language to be standardized by Institute of Electrical and Electronic Engineers, through the IEEE 1076 standard. An additional standard, the IEEE 1164, war later added to introduce a multi-valued logic system. VHDL is intended for circuit synthesis as well as circuit simulation [1].

The ALU is a building block of any microprocessor or DSP that performs many arithmetic functions based on the control input selection. The ALU is the heart of a microprocessor and performs all the basic operations. Besides the main ALU there are separate units which work independent of the main ALU for performing secondary operations such as address computation. Such units are present in pipelined microprocessors wherein the extra hardware requirement is for achieving greater speed of operation [5]. The ALU can perform basic arithmetic functions such as addition, subtraction etc. and logic functions including add, subtract, logic AND, logic OR, and logic XOR. These various functions of the ALU are implemented using a set of functional units each implementing a function, these may also be done using sharing of same hardware with use of certain additional units like multiplexers. To simulate the circuit "*ModelSim-Altera 6.4a (Quartus II 9.0) Starter Edition*'' was used.

## 2. HOW IT WORKS

The Arithmetic-Logistic Unit (ALU) performs arithmetic (addition, subtraction, increment, decrement etc.) and/or logical operations (AND, OR, XOR, SHIFT, ROTATE, etc.). This unit has two 8-bit inputs and a single 8-bit output. It can perform eight arithmetic and eight logical operation, the operation are selected by the four bit select input 'sel(3:0)'. In figure 1 a(7:0) and b(7:0) are two input of the ALU. cin is carry input and bin is borrow input of the ALU. Getting input from these input signal described above, ALU produce output. These output are obtained from result(7:0), cout and bout. result(7:0) is output result, cout is output carry and bout is the output borrow [1], [2], [3].

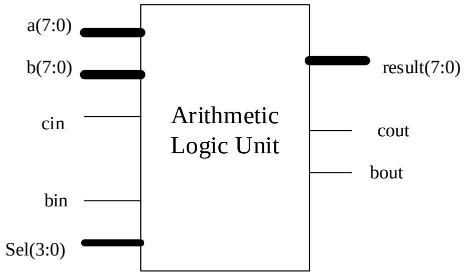


Fig. 1: Top Level Diagram of ALU

The operations which are performed by the ALU are listed in table 1. The op-code determine function performed by the ALU, for an example when the op-code is 0000 AND operation is selected and the ALU performs the specified AND operation. When the selected op-code is 1110 instead of 0000, the arithmetic Addition will be performed.

Table 1: Function of ALU

| Function | Op-code | Operation |
|---|---|---|
| AND | 0000 | A AND B |
| OR | 0001 | A OR B |
| XOR | 0010 | A XOR B |
| NOT | 0011 | NOT A |
| Shift Right | 0100 | ----- |
| Shift Left | 0101 | ----- |
| Rotate Right | 0110 | ----- |
| Rotate Left | 0111 | ----- |
| Transfer A | 1000 | A |
| Transfer B | 1001 | B |
| Increment A | 1010 | A+1 |
| Increment B | 1011 | B+1 |
| Decrement A | 1100 | A-1 |
| Decrement B | 1101 | B-1 |
| Addition | 1110 | $A+B+C_{in}$ |
| Subtraction | 1111 | $A-B-C_{in}$ |

## 3. BLOCK DIAGRAM REPRESENTATION

There are basically two units, one is Arithmetic unit and another one is Logic unit. Arithmetic unit performs the arithmetic operations and logic unit performs the logical operations. In figure 2 the block diagram of ALU is shown. When inputs a(7:0) and b(7:0) with cin and bin are fed to the ALU, it performs all the sixteen operations simultaneously. But the output result depends on the select signal sel(3:0). When the fourth bit of select input is one, it select the output of the arithmetic unit and when zero it select the output of logic unit. This task is performed by the MUX [1], [4].

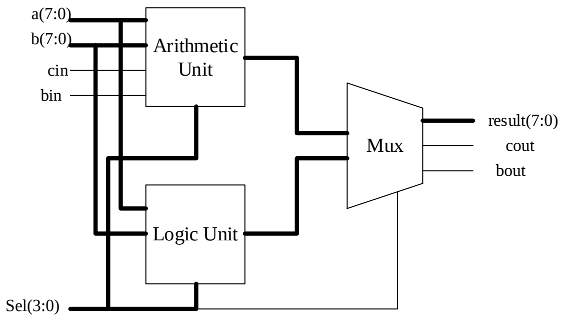


Fig. 2: Block Diagram of ALU

## 4. DESIGN PROCEDURE

### 4.1 VHDL Coding

The first step of VHDL coding of the circuit is library declaration. The std_logic_1164 package of the ieee library specifies a multilevel logic system; std is resource library for VHDL environment and work library where we save our design. The ieee library contain several packages including std_logic _1164 the specifies std_logic and std_unlogic multilevel logic systems [1], [2].

### 4.2 Entity declaration

Entity is a list with specifications of all input and output pins of circuit. Its syntax is shown below.

```
ENTITY alu IS
GENERIC(size: INTEGER :=8);
PORT(a,b: IN STD_LOGIC_VECTOR(size-1 DOWNTO 0);
sel : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
cin : IN STD_LOGIC;
bin : IN STD_LOGIC;
 result : OUT STD_LOGIC_VECTOR(size-1 DOWNTO 0);
cout : OUT STD_LOGIC;
bout : OUT STD_LOGIC;
END alu
```

### 4.3 Architecture

We design the ALU for sixteen operations. In this code sixteen operations are declared as several components. The sixteen components was saved in a library named work. After defining this component we called the component in main ALU architecture. For an example

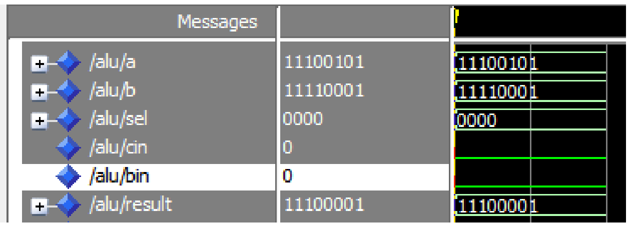
```
COMPONENT addition IS
PORT (a,b: IN STD_LOGIC_VECTOR(7 DOWNTO 0);
cin: IN STD_LOGIC;
s: OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
cout: OUT STD_LOGIC);
END COMPONENT;
```

The VHDL code for addition is given below as an example

```
library ieee;
USE ieee.std_logic_1164.all;
ENTITY addition IS
GENERIC(length: INTEGER :=8);
PORT (a,b: IN STD_LOGIC_VECTOR(length-1 DOWNTO 0);
cin: IN STD_LOGIC;
s: OUT STD_LOGIC_VECTOR(length-1 DOWNTO 0);
cout: OUT STD_LOGIC);
END addition;
ARCHITECTURE addition OF addition IS
BEGIN
PROCESS(a,b,cin)
VARIABLE carry : STD_LOGIC_VECTOR(length DOWNTO 0);
BEGIN
carry(0) := cin;
 FOR i IN 0 TO length-1 LOOP
s(i)<=a(i)XOR b(i) XOR carry(i);
carry(i+1) :=(a(i) AND b(i)) OR (a(i) AND carry(i)) OR (b(i) AND carry(i));
 END LOOP;
cout <= carry(length);
END PROCESS;
 END addition;
```

## 5. RESULT

### 5.1 AND Operation

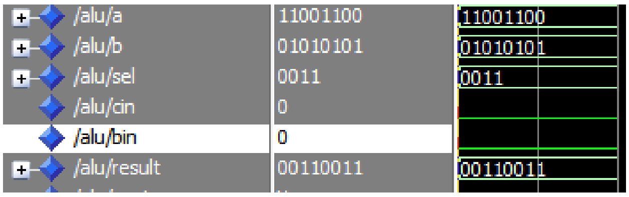


| Messages | |
|---|---|
| /alu/a | 11100101 |
| /alu/b | 11110001 |
| /alu/sel | 0000 |
| /alu/cin | 0 |
| /alu/bin | 0 |
| /alu/result | 11100001 |

Fig. 3: Logical AND Operation

Here input a=11100101 and b=11110001, select input sel=0000 which indicates logical AND operation. The result=11100001

### 5.2 XOP Operation

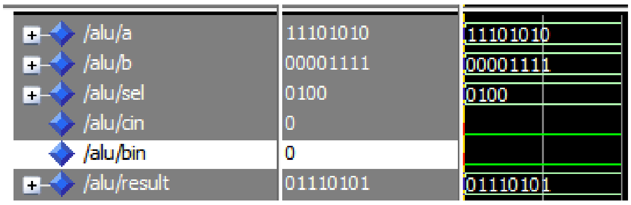


| | |
|---|---|
| /alu/a | 11001100 |
| /alu/b | 01010101 |
| /alu/sel | 0011 |
| /alu/cin | 0 |
| /alu/bin | 0 |
| /alu/result | 00110011 |

Fig. 4: XOR operation

For the logical XOR operation the inputs a=11001100 and b=01010101, sel=0011. The op-code 0011determaines the logical XOR operation. The result of the operation is result=00110011. If we perform the logical XOR operation manually, we will get the same result.

### 5.3 Rotate Right Operation

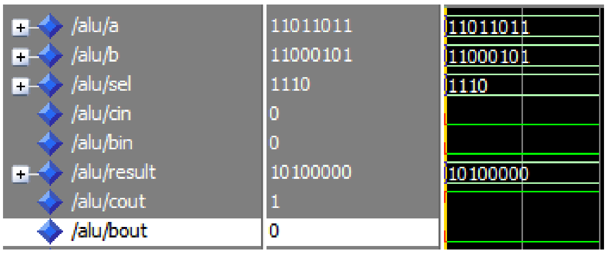


| | |
|---|---|
| /alu/a | 11101010 |
| /alu/b | 00001111 |
| /alu/sel | 0100 |
| /alu/cin | 0 |
| /alu/bin | 0 |
| /alu/result | 01110101 |

Fig. 5: Rotate Right

The rotate operation can be performed on only the input a. In this case a=11101010 and select input sel=0100, result=01110101 which indicates the input a is rotated one bit.

### 5.4 Addition

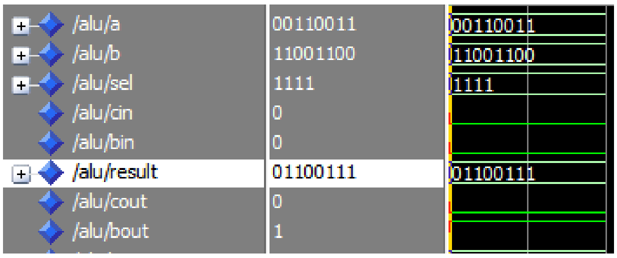


| | |
|---|---|
| /alu/a | 11011011 |
| /alu/b | 11000101 |
| /alu/sel | 1110 |
| /alu/cin | 0 |
| /alu/bin | 0 |
| /alu/result | 10100000 |
| /alu/cout | 1 |
| /alu/bout | 0 |

Fig. 6: Addition

Input a=11011011 and b=11000101, carry input cin=0, sel=1110. Output result=10100000 and carry output cout=1.

### 5.5 Subtraction

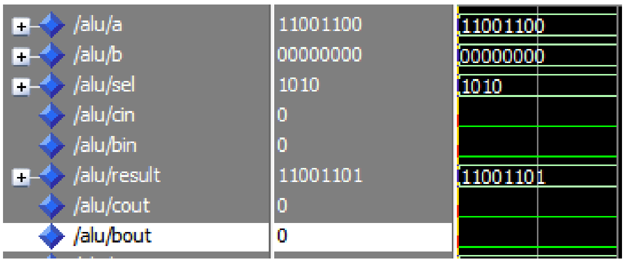


| | |
|---|---|
| /alu/a | 00110011 |
| /alu/b | 11001100 |
| /alu/sel | 1111 |
| /alu/cin | 0 |
| /alu/bin | 0 |
| /alu/result | 01100111 |
| /alu/cout | 0 |
| /alu/bout | 1 |

Fig. 7: Subtraction

Input a=00110011 and b=11001100, borrow input bin=0, sel=1111, result=01100111 and borrow output bout=1.

### 5.6 Increment



| | |
|---|---|
| /alu/a | 11001100 |
| /alu/b | 00000000 |
| /alu/sel | 1010 |
| /alu/cin | 0 |
| /alu/bin | 0 |
| /alu/result | 11001101 |
| /alu/cout | 0 |
| /alu/bout | 0 |

Fig. 8: Increment

The increment operation can be performed on only input a. Here a=11001100, select input sel=1010 and output result=11001101, which is incremented by 1.

## 6. DATAFLOW ANALYSIS

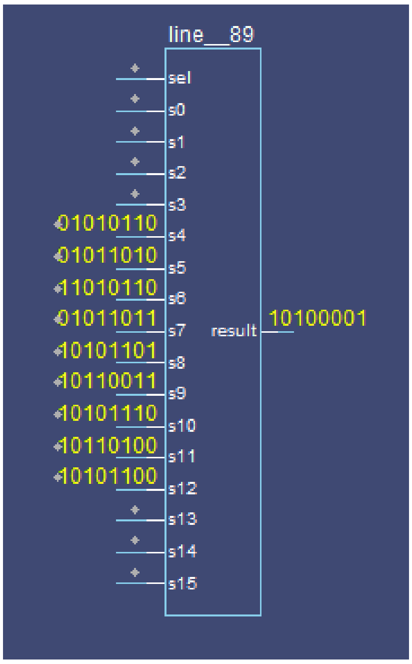


Fig. 9: Internal Dataflow of ALU

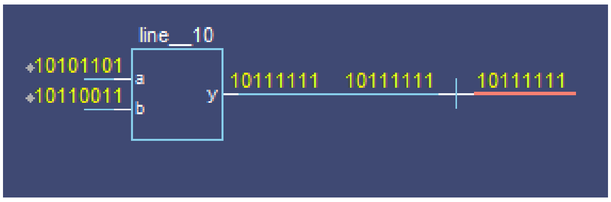


Fig. 10: Internal Dataflow of OR Operation

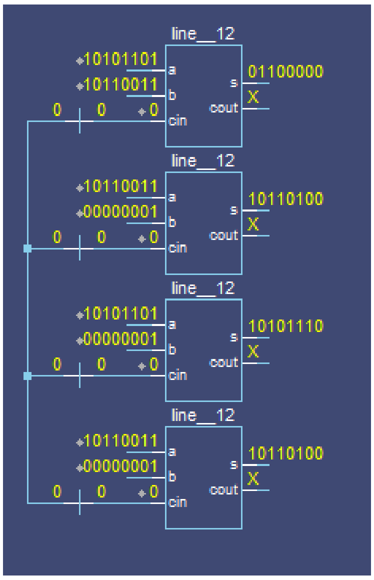


Fig. 11: Internal Dataflow Stracture for Addition Operation

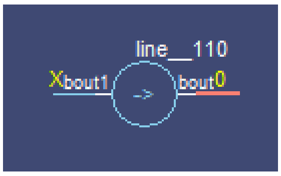


Fig. 12: Internal Borrow Flow

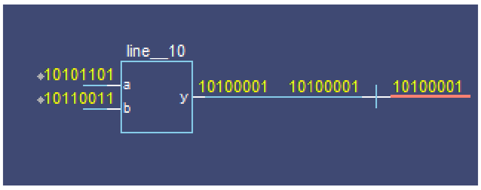


Fig. 13: Internal AND Operation

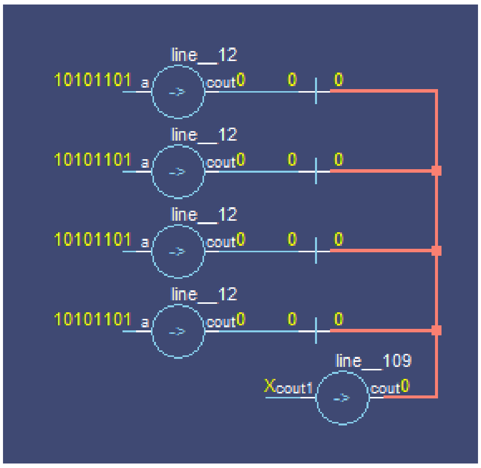


Fig. 14: Internal Carry Flow

## 7. CONCLUSION

In this project, Arithmetic Logic Unit was successfully designed and implemented using Very High Speed Hardware Descriptive Language and *ModelSim-Altera 6.4a (Quartus II 9.0) Starter Edition''*. All the primary operations of Arithmetic Logic Unit are fabulously done alongside some extra features that provide status of output. Graphical Splitting and Truth Table formation provided a synthesizable system design by separating Arithmetic, Logic, Shifter and Rotator blocks which were integrated in later stages. Design methods involved follows a TOP-DOWN approach in which software design leads the physical and hardware construction. Software section has been realized using Behavioral Model of VHDL. The programming is done for 8 bit lanes of 2 inputs each but system can accommodate up to 6 input channels. The 4 bit select input holds responsibility for shaping the output of specified operation. Further enhancements can be made on this system by adding more number of inputs with increased number of bit size. Digital Signal Processing (DSP) is being credited with lots of applications from VHDL designs. Advancement in applications of ALU can mutually benefit the two fields

## 9. REFERENCES

[1] Volnei A. Pedroni, *Circuit Design with VHDL*, MIT Press Cambridge, Massachusetts London, England, 2004

[2] Mark Zwolin´ ski, *Digital System Design with VHDL*, Pearson Education Limited, Edinburgh Gate, England, Second edition published 2004

[3] M. Morris Mano, *Digital Logic and Computer Design*, Prentice, Hall of India, 2000

[4] Shrivastava Purnima, Tiwari Mukesh, Singh Jaikaran and Rathore Sanjay, *"VHDL Environment for Floating point Arithmetic Logic Unit - ALU Design and Simulation"* Research Journal of Engineering, Sciences Vol. 1(2), 1-6, August (2012), ISSN 2278 – 9472

[5] Anushka Pakrashi, Arindam Bose, Kausik Bhattacharya, Monigingir Pal and Tanaya Bose, *"Design and Implementation of a 32-Bit ALU on Xilinx FPGA Using VHDL"* Future Institute Of EngineeringAnd Management, 2012