S. B. JAIN INSTITUTE OF TECHNOLOGY, MANAGEMENT & RESEARCH,NAGPUR.

Practical No. 06

Aim: Design and implementation of Arithmetic Logic Unit (ALU) and verify it using test bench.

Name of Student	**: Aadesh R. Motghare**
Roll No.	**: 41(ET20065)**
Semester/Year	**: 6th Sem/3rd Year**
Academic Session	**: 2022-23**
Date of Performance	**:**
Date of Submission	**:**

AIM: Design and implementation of Arithmetic Logic Unit (ALU) and verify it using test bench.

OBJECTIVE:

- To verify the functionality of Arithmetic and Logical Unit.
- To implement and test the circuits which constitute the arithmetic logic circuit(ALU)
- To develop the logic of designing the digital calculator using Arithmetic andLogical Unit.

SOFTWARE: - Xilinx ISE9.1.

THEORY:-

The **ALU** is a digital circuit that provides arithmetic and logic operation It is the fundamental building block of the central processing unit of a computer. ALU performs operations such as addition, subtraction and multiplication of integers and bit-wise AND, OR, NOT, XOR and other Boolean operations.

A modern CPU has a very powerful ALU and it is complex in design. In addition to ALU modern CPU contains a control unit and a set of registers. Most of the operations are performed by one or more ALU's, which load data from the input register. Registers are a small amount of storage available to the CPU. These registers can be accessed very fast. The control unit tells ALU what operation to perform on the available data. After calculation/manipulation, the ALU stores the output in an output register.

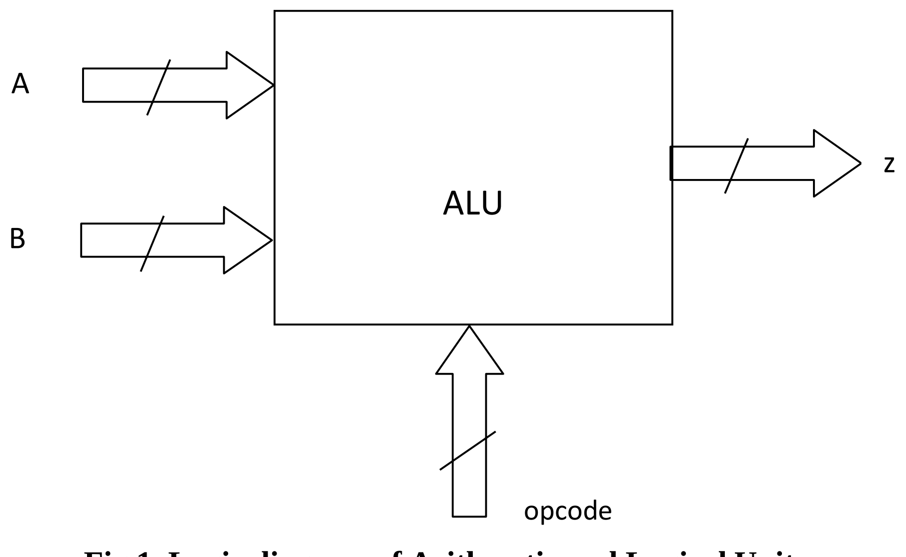


Fig 1. Logic diagram of Arithmetic and Logical Unit

UNIT	INSTRUCTION	OPCODE
LOGIC	A and B	000
	A OR C	001
	A NAND B	010
	A NOR B	011
ARITHEMATIC	A + B	100
	A-B	101
	A+1	110
	A-1	111

Fig.2 Example truth table for ALU with 3 bit opcode

VHDL CODE

STEPS FOR PROGRAM:-

Step 1.Library /Package Declaration : Involves declaration of all libraries and respective packages used in the design.

```
LIBRARY library_name;
USE library_name.package_name.all;
```

Step 2.Entity:

```
ENTITY entity_name is
        PORT(signal_name(s): mode signal_type;
                signal_name(s): mode signal_type;
                …);
end ENTITY entity_name;
```

Signals of the same mode and signal_type can be grouped on 1 line

MODE describes the direction data is transferred through port

- in – data flows into the port
- out – data flows out of port *only*
- buffer – data flows out of port *as well as read*

internally.

- inout – bi-directional data flow into and out of port

SIGNAL_TYPE defines the data type for the signal(s)

- bit – single signals that can have logic values 0 and 1.
- bit_vector – bus signals(vector form of bit) that can have logic values 0 and 1.
- std_logic – part of std_logic_1164 package of IEEE library. Used to represent 2 value logical values i.e 0 and 1 as well as other values such as high impedance ,don't care and others as described below.
- std_logic_vector – bus signals (vector form of std_logic) but IEEE standard for simulation and synthesis note that all vectors must have a range specified example for a 4 bit bus: bit_vector (3 downto 0) or std_logic_vector (3 downto 0).

In order to use std_logic and std_logic_vector we must include the library and package usage declarations in the VHDL model before the entity statement as follows:

LIBRARY ieee;
USE ieee.std_logic_1164.all;

Values for std-logic:

U	un-initialized (undefined logic value)
X	forced unknown logic value
0	Logic low
1	Logic High
Z	high impedance (tri-state)
W	weak unknown
L	weak 0
H	weak 1
-	don't care value (for synthesis minimization)

Step 3: Architecture Declaration

```
architecture architecture name of entity_name

 architecture_declarative_part;
begin

Statements;

 end architecture_name;
```

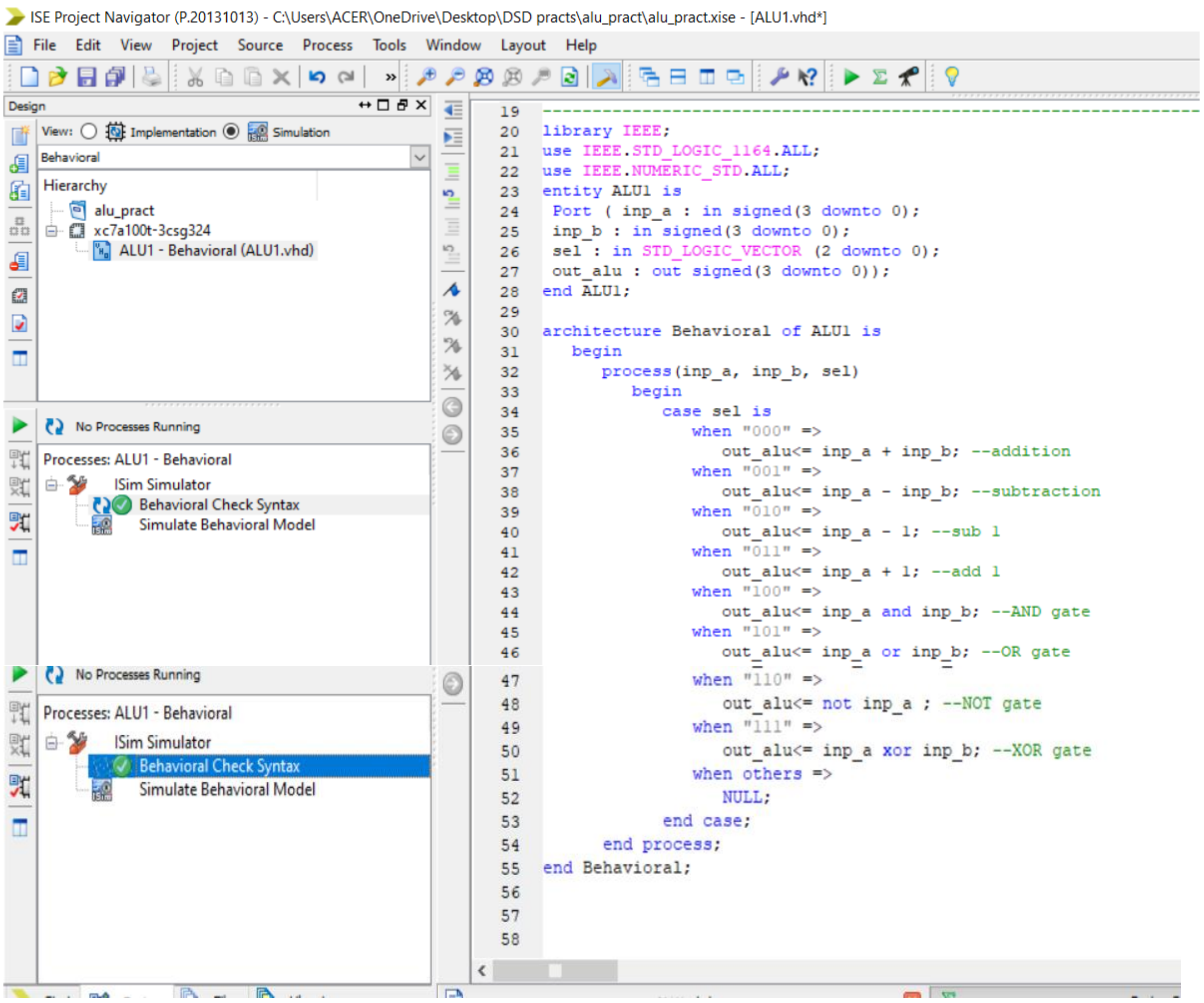
Here we should specify the entity name for which we are writing the architecture body. The architecture statements should be inside the begin and end keyword. Architecture declarative part may contain variables, constants, or component declaration.

Step 4:Simulate the VHDL code and remove the syntax

errors if any

Step 5: Write the testbench and verify the design

CODE:



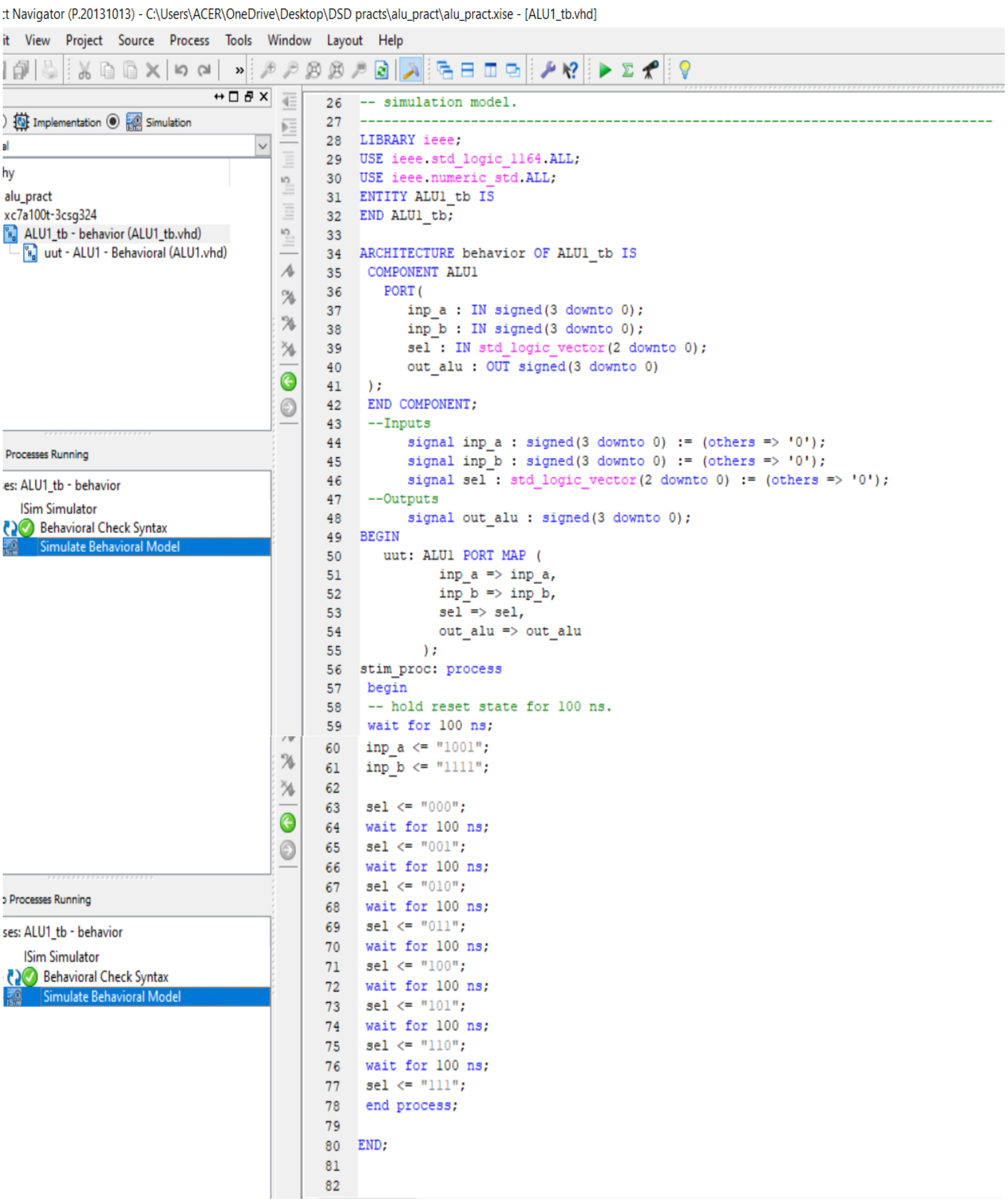
ISE Project Navigator (P.20131013) - C:\Users\ACER\OneDrive\Desktop\DSD practs\alu_pract\alu_pract.xise - [ALU1.vhd*]

File Edit View Project Source Process Tools Window Layout Help
Design
View: Implementation [✓ Simulation]
Behavioral
Hierarchy
alu_pract
xc7a100t-3csg324
ALU1 - Behavioral (ALU1.vhd)
No Processes Running
Processes: ALU1 - Behavioral
ISim Simulator
Behavioral Check Syntax
Simulate Behavioral Model

```
-------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
entity ALU1 is
 Port ( inp_a : in signed(3 downto 0);
 inp_b : in signed(3 downto 0);
 sel : in STD_LOGIC_VECTOR (2 downto 0);
 out_alu : out signed(3 downto 0));
end ALU1;

architecture Behavioral of ALU1 is
   begin
      process(inp_a, inp_b, sel)
         begin
            case sel is
               when "000" =>
                  out_alu<= inp_a + inp_b; --addition
               when "001" =>
                  out_alu<= inp_a - inp_b; --subtraction
               when "010" =>
                  out_alu<= inp_a - 1; --sub 1
               when "011" =>
                  out_alu<= inp_a + 1; --add 1
               when "100" =>
                  out_alu<= inp_a and inp_b; --AND gate
               when "101" =>
                  out_alu<= inp_a or inp_b; --OR gate
               when "110" =>
                  out_alu<= not inp_a ; --NOT gate
               when "111" =>
                  out_alu<= inp_a xor inp_b; --XOR gate
               when others =>
                  NULL;
            end case;
      end process;
end Behavioral;
```

TESTBENCH:


:t Navigator (P.20131013) - C:\Users\ACER\OneDrive\Desktop\DSD practs\alu_pract\alu_pract.xise - [ALU1_tb.vhd]
it View Project Source Process Tools Window Layout Help
Implementation
Simulation
hy
alu_pract
xc7a100t-3csg324
ALU1_tb - behavior (ALU1_tb.vhd)
uut - ALU1 - Behavioral (ALU1.vhd)
Processes Running
es: ALU1_tb - behavior
ISim Simulator
Behavioral Check Syntax
Simulate Behavioral Model

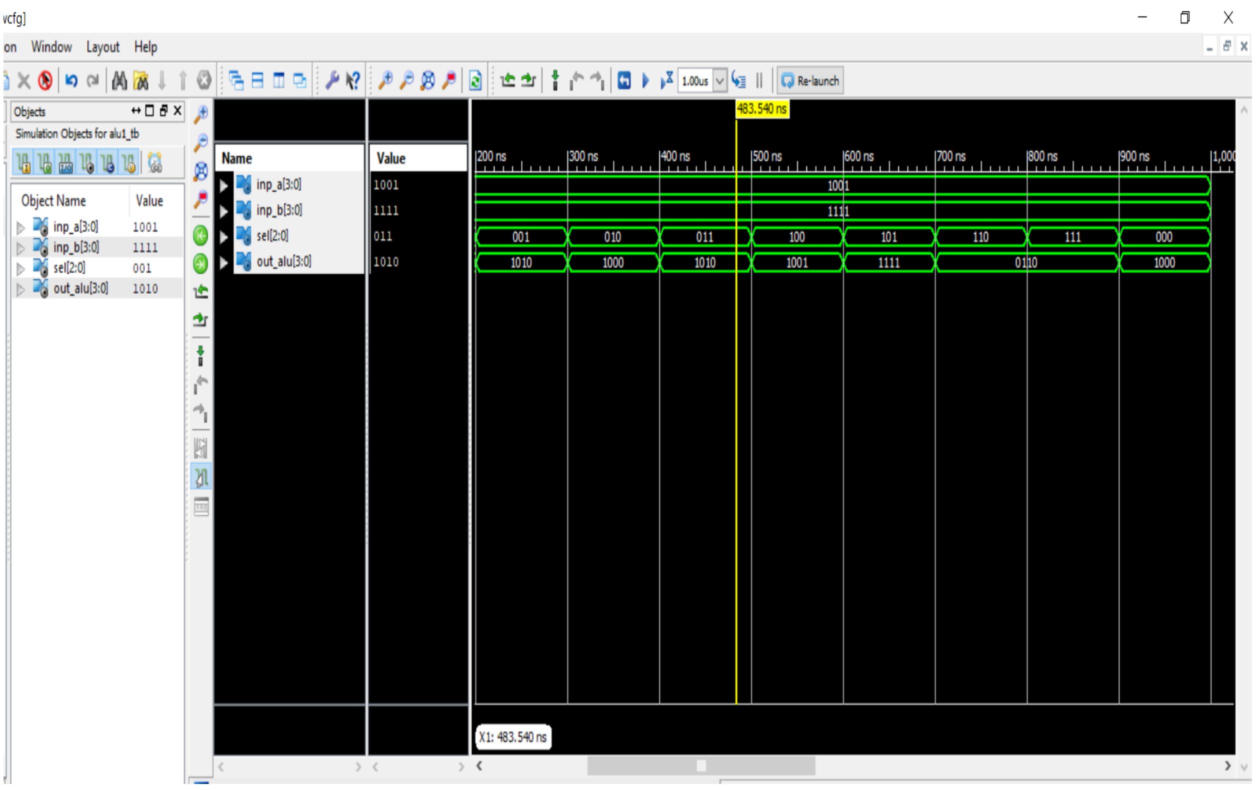
```
-- simulation model.
--------------------------------------------------------------------------------
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;
ENTITY ALU1_tb IS
END ALU1_tb;

ARCHITECTURE behavior OF ALU1_tb IS
 COMPONENT ALU1
    PORT(
       inp_a : IN signed(3 downto 0);
       inp_b : IN signed(3 downto 0);
       sel : IN std_logic_vector(2 downto 0);
       out_alu : OUT signed(3 downto 0)
 );
 END COMPONENT;
 --Inputs
       signal inp_a : signed(3 downto 0) := (others => '0');
       signal inp_b : signed(3 downto 0) := (others => '0');
       signal sel : std_logic_vector(2 downto 0) := (others => '0');
 --Outputs
       signal out_alu : signed(3 downto 0);
BEGIN
   uut: ALU1 PORT MAP (
           inp_a => inp_a,
           inp_b => inp_b,
           sel => sel,
           out_alu => out_alu
         );
stim_proc: process
 begin
 -- hold reset state for 100 ns.
 wait for 100 ns;
 inp_a <= "1001";
 inp_b <= "1111";

 sel <= "000";
 wait for 100 ns;
 sel <= "001";
 wait for 100 ns;
 sel <= "010";
 wait for 100 ns;
 sel <= "011";
 wait for 100 ns;
 sel <= "100";
 wait for 100 ns;
 sel <= "101";
 wait for 100 ns;
 sel <= "110";
 wait for 100 ns;
 sel <= "111";
 end process;

END;
```

WAVEFORM:-



RESULT:-

The VHDL code for Arithmetic Logic Unit (ALU) is executed and desired output is obtained.

CONCLUSION:

Here, I successfully design the Arithmatic Logic Unit (ALU) and the Test Bench for the same to check the output.

DISCUSSION & VIVA VOCE

1) Explain the working of 8 Bit Arithmetic and Logical Unit.

2) How to design Arithmetic and Logical Unit in VHDL

3) What are the applications of Arithmetic and Logical Unit.

REFERENCE:

- VHDL Primer–J Bhasker –Pearson Education
- NPTEL Video Lecture Link https://www.youtube.com/watch?v=RkAE4zE4uSE.