Lab Session 5: ALU using *Behavioural Modelling*

Sravan K Suresh
Roll Number: 22B3936

September 8, 2023

Overview of the Report

This report contains:

- Approach to the Problem Statement
- Circuit Diagram of *ALU*
- RTL Waveform Simulation of the function
- RTL Gates Map of the function
- OUTPUT verifying all tests successful

Problem Statement

The aim of the assignment was to implement a MUX of the specifications as defined by the diagram below using *Behavioural modelling*. After that, we also had to perform Scanchain to ensure that the design was correct.-

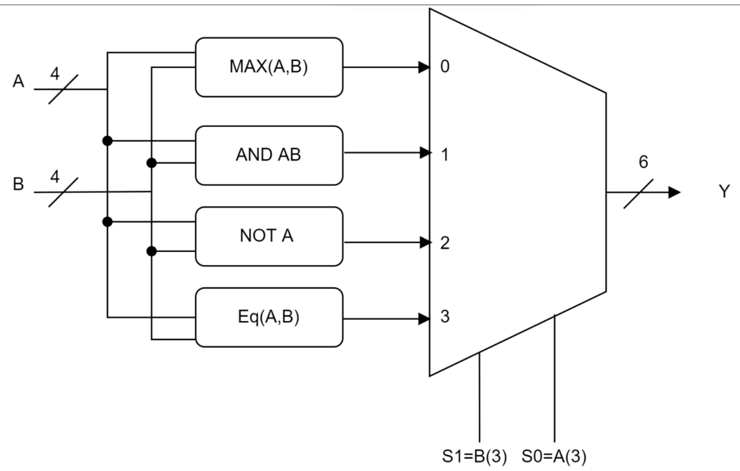


Figure 1: ALU with 4 functions

S1 S0	ALU Output
0 0	MAX(A,B): This block outputs larger number between A and B else outputs 0000.
0 1	AND A B: This block performs AND operation between A , B.
1 0	NOT A: This block performs NOT operation of A.
1 1	Eq(A,B): This block outputs the number whenever A=B else it should output 0000.

Procedure followed:

1. Made functions `MAX(A,B)`, `ANDing(A,B)`, `Compl(A)`, `isequal(A,B)`.
2. Made a process with static sensitivity list including A and B which calls any of the above functions based on the values of `B(3)` and `A(3)` when used as 'select inputs'.
3. Compiled and ran RTL simulation using the Testbench.
4. Dumped `.svf` file generated using Programmer onto the `Xen10` Board using `urjtag`.
5. Performed Scanchain of the Xen10 board to get a successful result for inputs.

1 Approach to the Problem Statement

Functions for behavioural modelling were implemented depending on the input of 'select lines' as follows-
$(S1, S0) := 00 \rightarrow MAX(A, B)$
$(S1, S0) := 01 \rightarrow ANDing(A, B)$
$(S1, S0) := 10 \rightarrow compl(A)$
$(S1, S0) := 11 \rightarrow isequal(A, B)$

1.1 MAX(A,B)

This function determined the larger of the two binary numbers fed to it as input after concatenating `"00"` onto the left of it and in case if the two numbers `A` and `B` were equal, it gave `std_logic_vector` return output `"000000"` when passed the values of `A,B`.

Code Documentation

```
function MAX(A: in std_logic_vector(3 downto 0) :=”0000”;
       B: in std_logic_vector(3 downto 0):=”0000”)
  return std_logic_vector is
  variable Eq: std_logic_vector(5 downto 0):=(others =>’0’)
   ;

  begin
    Eq := isequal(A,B);
      if Eq /= ”000000”
      then
        return ”000000”;
      else
        if A > B
        then
          return (”00” & A);
        else
          return (”00” & B);
        end if;
      end if;
  end function MAX;
```

1.2 ANDing(A, B)

This function computed the output of `AND` operation carried out on `A` and `B` (the two binary numbers fed to it as input) and returned `std_logic_vector` output after concatenating `"00"` onto the left of it when passed the values of `A,B`.

Code Documentation

```
function ANDing(A: in std_logic_vector(3 downto 0) :="0000";
             B: in std_logic_vector(3 downto 0):="0000")
  return std_logic_vector is
  variable AandB: std_logic_vector(3 downto 0):=(others=>'0');
  begin

    for i in 0 to 3 loop
        AandB(i) := A(i) and B(i);
    end loop;
      return "00" & AandB;
  end function ANDing;
```

1.3 Compl(A)

This function returned a `std_logic_vector` output of size 6 representing the complement of the binary number A of size 4 fed to it as input, after concatenating `"00"` onto the left of it.

Code Documentation

```
function compl(A: in std_logic_vector(3 downto 0) :="0000";
            B: in std_logic_vector(3 downto 0):="0000")
  return std_logic_vector is
  variable Abar: std_logic_vector(3 downto 0):=(others =>'0');
  begin
    for i in 0 to 3 loop
        Abar(i) := not A(i);
    end loop;
      return "00" & Abar;
  end function compl;
```

1.4 isequal(A, B)

This function checked if the two inputs `A` and `B` of size 4 are equal or not.
If they were equal, then the number itself was returned as `std_logic_vector` output of size 6 after concatenating `"00"` onto the left of it.
Else, the function returned `"000000"` as `std_logic_vector` output of size 6.

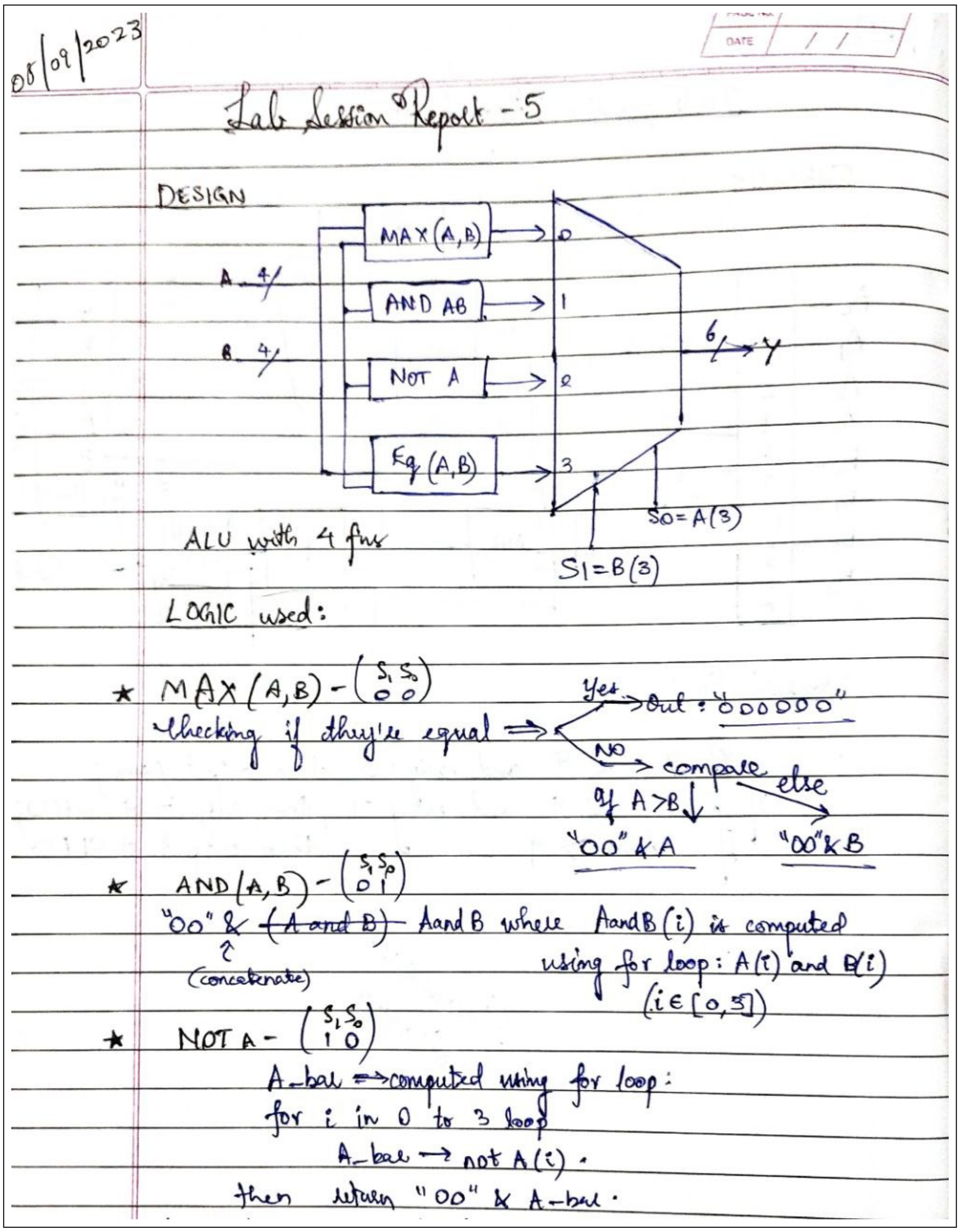
Code Documentation

```
function isequal(A: in std_logic_vector(3 downto 0) :="0000";
            B: in std_logic_vector(3 downto 0):="0000")
  return std_logic_vector is
  variable AxnorB: std_logic_vector(3 downto 0):=(others =>'0');
  begin

    if A = B
    then
      return "00" & A;
    else
      return "000000";
    end if;
  end function isequal;
```

Observations

2 Pen-paper Design of the Circuit



DATE / /
MAX (A,B)
AND AB
NOT A
Eq (A,B)
A 4
B 4
0
1
2
3
6
Y
S0 = A(3)
S1 = B(3)

08/09/2023

Lab Session Report - 5

DESIGN

ALU with 4 fns

LOGIC used:

★ MAX (A,B) - $\begin{pmatrix} S_1 & S_0 \\ 0 & 0 \end{pmatrix}$

checking if they're equal ⇒ Yes → Out : "000000"; NO → compare: if A>B ↓ "00" & A; else → "00" & B

★ AND (A,B) - $\begin{pmatrix} S_1 & S_0 \\ 0 & 1 \end{pmatrix}$

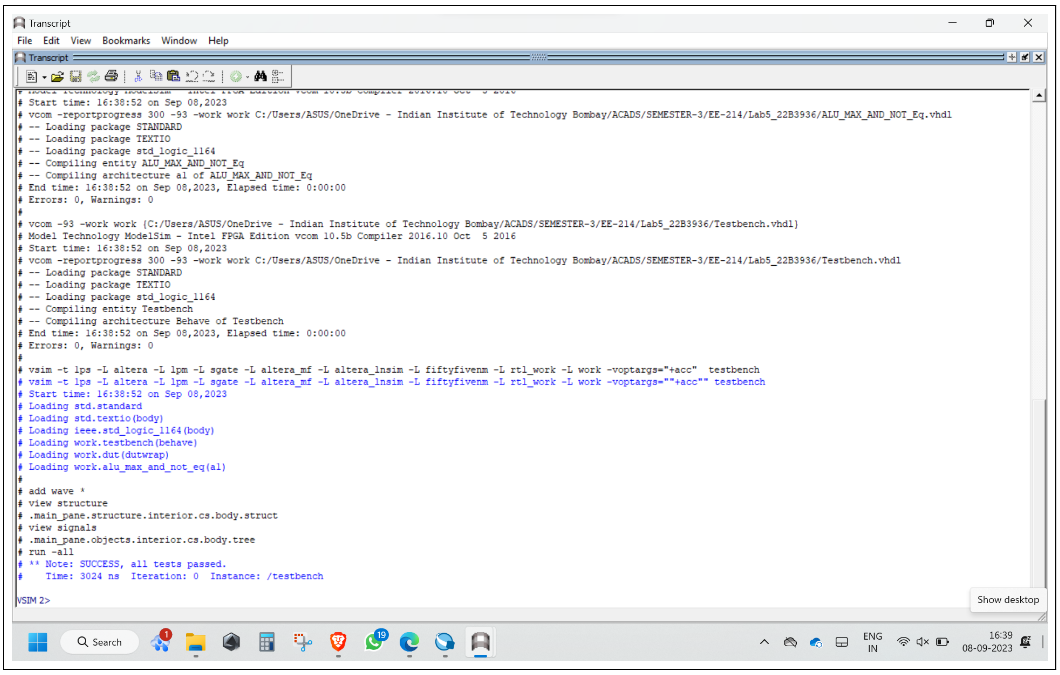
"00" & ~~(A and B)~~ AandB where AandB(i) is computed using for loop : A(i) and B(i) ($i \in [0,3]$)

↑ (concatenate)

★ NOT A - $\begin{pmatrix} S_1 & S_0 \\ 1 & 0 \end{pmatrix}$

```
A_bar ⇒ computed using for loop:
for i in 0 to 3 loop
        A_bar → not A(i).
then return "00" & A_bar.
```

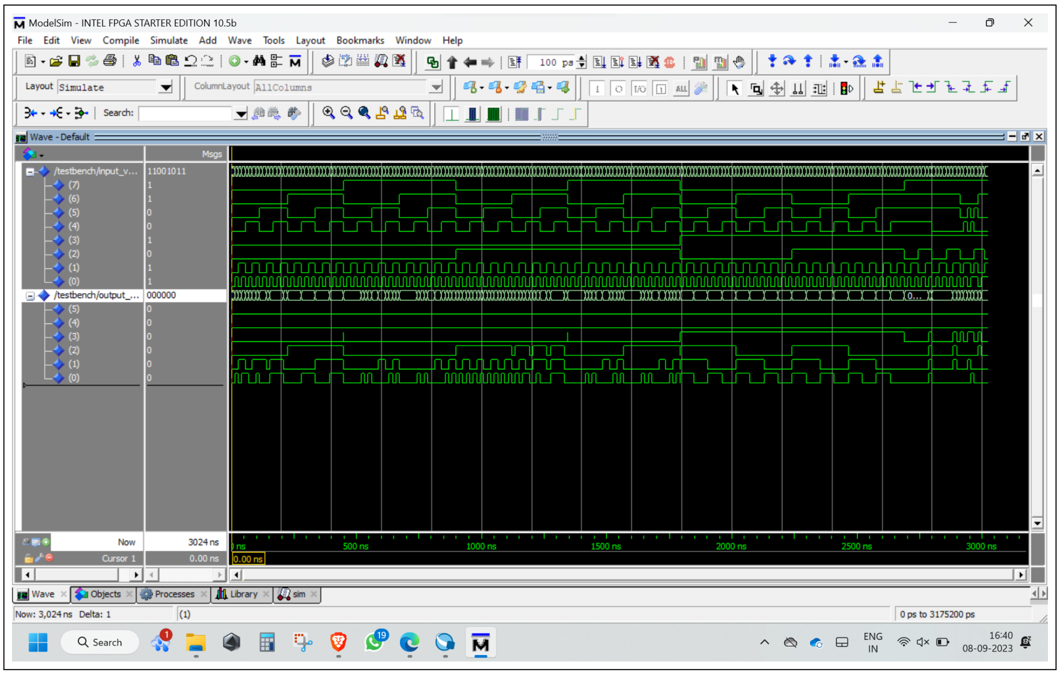
3 Transcript



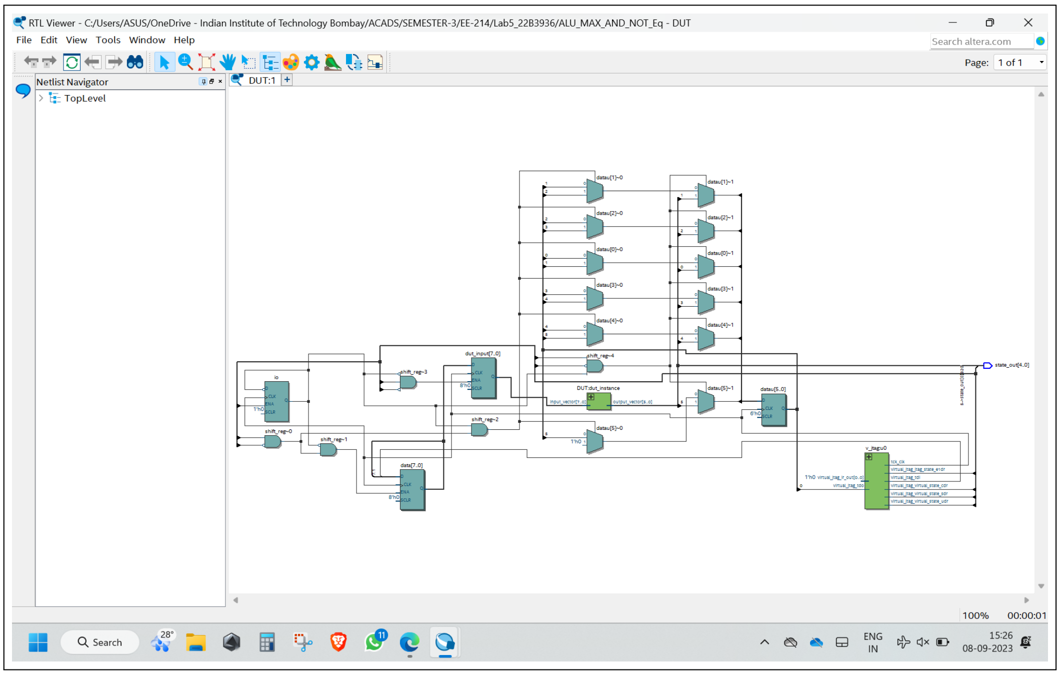
```
# Model Technology ModelSim - Intel FPGA Edition vcom 10.5b Compiler 2016.10 Oct  5 2016
# Start time: 16:38:52 on Sep 08,2023
# vcom -reportprogress 300 -93 -work work C:/Users/ASUS/OneDrive - Indian Institute of Technology Bombay/ACADS/SEMESTER-3/EE-214/Lab5_22B3936/ALU_MAX_AND_NOT_Eq.vhdl
# -- Loading package STANDARD
# -- Loading package TEXTIO
# -- Loading package std_logic_1164
# -- Compiling entity ALU_MAX_AND_NOT_Eq
# -- Compiling architecture a1 of ALU_MAX_AND_NOT_Eq
# End time: 16:38:52 on Sep 08,2023, Elapsed time: 0:00:00
# Errors: 0, Warnings: 0
#
# vcom -93 -work work {C:/Users/ASUS/OneDrive - Indian Institute of Technology Bombay/ACADS/SEMESTER-3/EE-214/Lab5_22B3936/Testbench.vhdl}
# Model Technology ModelSim - Intel FPGA Edition vcom 10.5b Compiler 2016.10 Oct  5 2016
# Start time: 16:38:52 on Sep 08,2023
# vcom -reportprogress 300 -93 -work work C:/Users/ASUS/OneDrive - Indian Institute of Technology Bombay/ACADS/SEMESTER-3/EE-214/Lab5_22B3936/Testbench.vhdl
# -- Loading package STANDARD
# -- Loading package TEXTIO
# -- Loading package std_logic_1164
# -- Compiling entity Testbench
# -- Compiling architecture Behave of Testbench
# End time: 16:38:52 on Sep 08,2023, Elapsed time: 0:00:00
# Errors: 0, Warnings: 0
#
# vsim -t 1ps -L altera -L lpm -L sgate -L altera_mf -L altera_lnsim -L fiftyfivenm -L rtl_work -L work -voptargs="+acc"  testbench
# vsim -t 1ps -L altera -L lpm -L sgate -L altera_mf -L altera_lnsim -L fiftyfivenm -L rtl_work -L work -voptargs=""+acc"" testbench
# Start time: 16:38:52 on Sep 08,2023
# Loading std.standard
# Loading std.textio(body)
# Loading ieee.std_logic_1164(body)
# Loading work.testbench(behave)
# Loading work.dut(dutwrap)
# Loading work.alu_max_and_not_eq(a1)
#
# add wave *
# view structure
# .main_pane.structure.interior.cs.body.struct
# view signals
# .main_pane.objects.interior.cs.body.tree
# run -all
# ** Note: SUCCESS, all tests passed.
#    Time: 3024 ns  Iteration: 0  Instance: /testbench

VSIM 2>
```

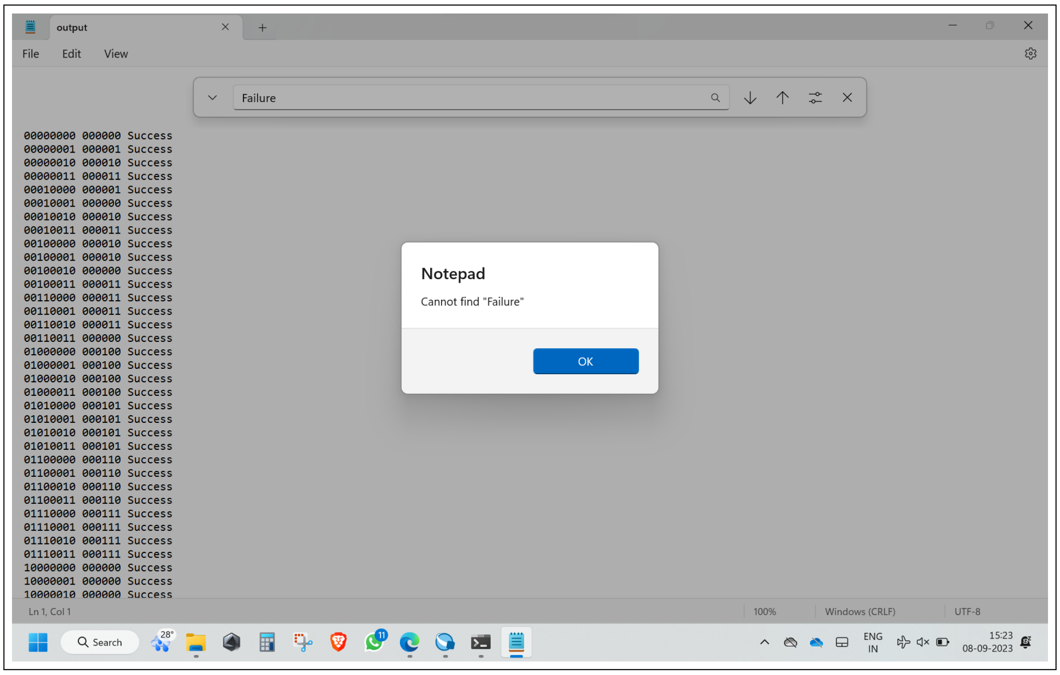
4 RTL Waveform Simulation



5 RTL Gates Map



6 All Success OUTPUT



```
00000000 000000 Success
00000001 000001 Success
00000010 000010 Success
00000011 000011 Success
00010000 000001 Success
00010001 000000 Success
00010010 000010 Success
00010011 000011 Success
00100000 000010 Success
00100001 000010 Success
00100010 000000 Success
00100011 000011 Success
00110000 000011 Success
00110001 000011 Success
00110010 000011 Success
00110011 000000 Success
01000000 000100 Success
01000001 000100 Success
01000010 000100 Success
01000011 000100 Success
01010000 000101 Success
01010001 000101 Success
01010010 000101 Success
01010011 000101 Success
01100000 000110 Success
01100001 000110 Success
01100010 000110 Success
01100011 000110 Success
01110000 000111 Success
01110001 000111 Success
01110010 000111 Success
01110011 000111 Success
10000000 000000 Success
10000001 000000 Success
10000010 000000 Success
```

Notepad

Cannot find "Failure"

OK

References

[1] EE214 Github Page: https://ee214.github.io/