

EE214 - Digital Circuits Lab

ALU

Friday Batch

09/09/2022

Instructions:

1. Use **Behavioral-Dataflow** modelling for writing VHDL description
2. Perform RTL simulation using the provided testbench and tracefile.
3. Demonstrate the simulations to your TA
4. Perform **Scanchain** on the Xenon board and verify with your TA.
5. Submit the entire project files in .zip format in moodle.

Problem Statement: [20 Marks (5 Marks*4)]

1. Describe the given ALU in VHDL. This ALU circuit performs various functions based on select lines.

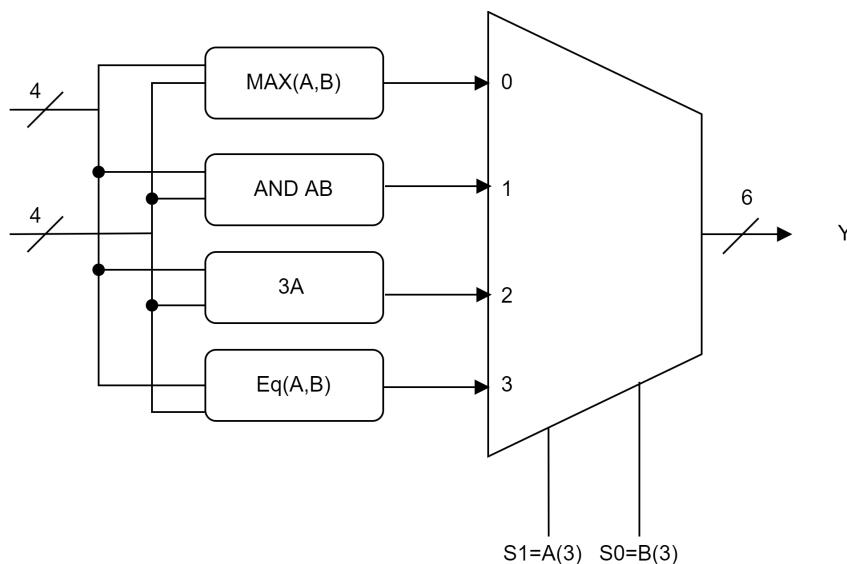


Figure 1: ALU with 4 functions

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 bitwise AND operation between A , B.
1 0	3*A: This block Produces output as 3*A
1 1	Eq(A,B): This block outputs the number whenever A=B else it should output 0000.

- In this problem MSB of inputs A and B are also working as selection lines. S0 is connected to MSB of input B [B(3)] and S1 is connected to MSB of input A [A(3)].
- Don't use multiply operation directly.
- Don't use numeric_std library.
- Simulate your design using the generic testbench to confirm the correctness of your description.
- Tracefile format < A3 A2 A1 A0 B3 B2 B1 B0 > < Y5 Y4 Y3 Y2 Y1 Y0 > 1 1 1 1 1 1
- Perform Scanchain on the Xenon board and verify with your TA. [5 Marks]

```

library ieee;
use ieee.std_logic_1164.all;

entity alu_beh is
    generic(
        operand_width : integer:=4);
    port (
        A: in std_logic_vector(operand_width-1 downto 0);
        B: in std_logic_vector(operand_width-1 downto 0);
        op: out std_logic_vector(5 downto 0)) ;
end alu_beh;

architecture a1 of alu_beh is
    function add(A: in std_logic_vector(operand_width-1 downto 0);
        B: in std_logic_vector(operand_width-1 downto 0))
        return std_logic_vector is
        -- Declare "sum" and "carry" variable
        -- you can use aggregate to initialize the variables as shown below
        -- variable variable_name : std_logic_vector(3 downto 0) := (others => '0');
    begin
        -- write logic for addition
        -- Hint: Use for loop
        return sum; --according to your logic you can change what you want to return
    end add;

begin
alu : process( A, B)
variable sel : std_logic_vector(1 downto 0);

--declare other variables
begin
    -- complete VHDL code for various outputs of ALU based on select lines
    sel := ---;
    case sel is
        when "00" =>
            -- Hint: use if/else statement
            --
            -- add function usage :
            -- signal_name <= add(A,B)
            -- variable_name := add(A,B)
            --
            -- concatenate operator usage:
            -- "0000"&A
    end process ; -- alu
end a1 ; -- a1

```

- Demo code snippet is given. Change the code accordingly.