EE214 - Digital Circuits Lab

ALU

Thrusday Batch

08/09/2022

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

1. Describe the given ALU using VHDL. This ALU circuit performs various functions based on select lines. [20 Marks (5 Marks*4)]

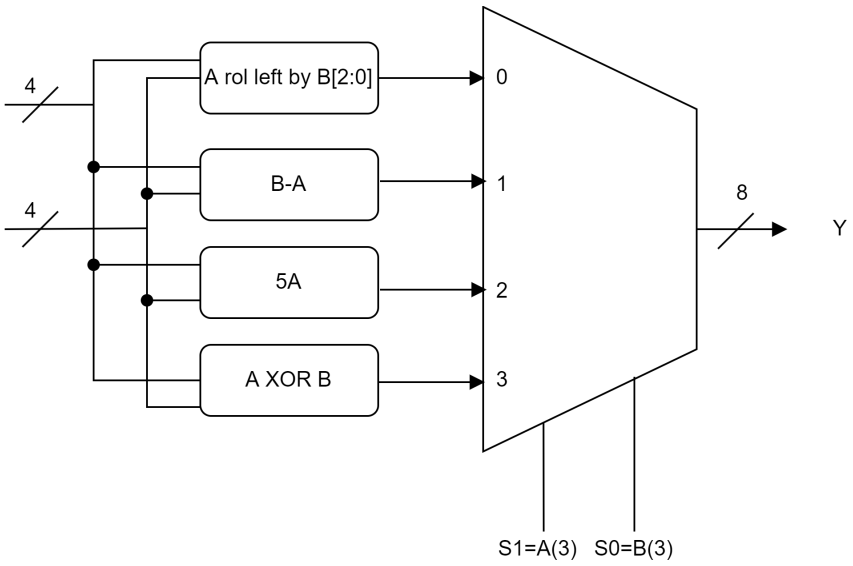

Figure 1: ALU with 4 functions

S1 S0	ALU Output
0 0	Rotate left A by B[2:0] number of bits
0 1	Performs B-A Operation
1 0	Produces output as 5*A
1 1	Performs bitwise A xor B Operation

- 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.
- Simulate your design using the generic testbench to confirm the correctness of your description.
- Tracefile format $< A3\,A2\,A1\,A0\,B3\,B2\,B1\,B0 > < Y7\,Y6\,Y5\,Y4\,Y3\,Y2\,Y1\,Y0 >$ 11111111
- Perform Scanchain on the Xenon board and verify with your TA. [5 Marks]

```vhdl
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((operand_width*2)-1 downto 0)) ;
end alu_beh;

architecture a1 of alu_beh is

    function sub(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
            -- declaring and initializing variables using aggregates
            variable diff : std_logic_vector(operand_width*2-1 downto 0):= (others=>'0');
            variable carry : std_logic:= '1';
        begin
            -- Hint: Use for loop to calculate value of "diff" and "carry" variable
            -- Use aggregates to assign values to multiple bits

            return diff;
    end sub;


    function rolf(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
            variable shift : std_logic_vector((operand_width*2)-1 downto 0):= (others=>'0');
        begin
            shift(operand_width-1 downto 0):= A;

            -- Hint: use for loop to calculate value of shift variable
            -- shift(____ downto _____) & shift(____ downto ______)
            -- to calculate exponent, you can use double asterisk. ex: 2**i

        return shift;
    end rolf;

begin
alu : process( A, B)
begin
    -- complete VHDL code for various outputs of ALU based on select lines
   -- Hint: use if/else statement
   --
   -- sub function usage :
   --   signal_name <= sub(A,B)
   --   variable_name := sub(A,B)
   --
   -- concatenate operator usage:
   --    "0000"&A
end process ; --alu
end a1 ; -- a1
```