EE214 Digital Circuits Laboratory
Wadhwani Electronics Laboratory
Electrical Engineering IIT Bombay

Problem set: 5 Date: September 12, 2024

Instructions:

1. Use Behavioral and Dataflow modelling for this experiment.
2. Perform RTL and Gate-level simulation using the provided testbench and tracefile.
3. Demonstrate the simulations to your TA.

Problem Statement

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

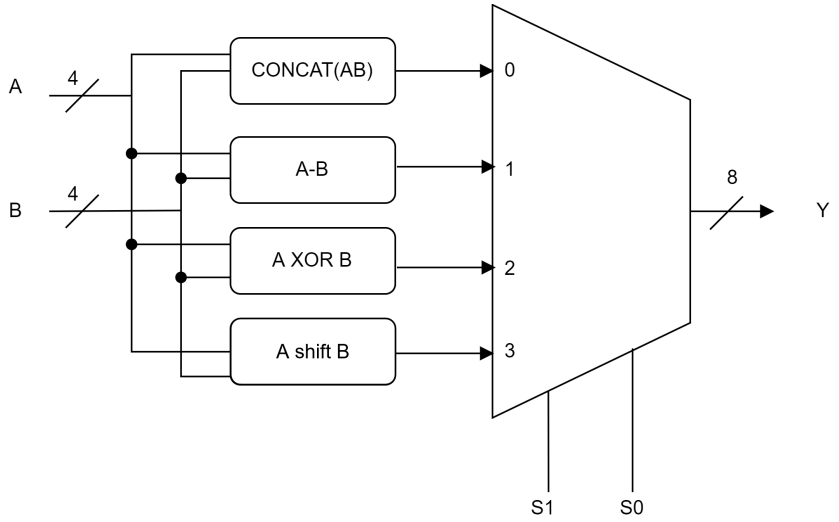

Figure 1: ALU with 4 functions

S1 S0	ALU Output
0 0	Concatenate two 4-bit inputs A and B
0 1	Performs A-B Operation
1 0	Performs A xor B Operation
1 1	Shift A by B bits [B3 - 0(left)/1(right)] [B2 - unused] [B1 B0 - no. of bits to be shifted]

- Simulate your design using the generic testbench to confirm the correctness of your description.
- Tracefile format $< S1\,S0\,A3\,A2\,A1\,A0\,B3\,B2\,B1\,B0 >< Y7\,Y6\,Y5\,Y4\,Y3\,Y2\,Y1\,Y0 >$ 11111111
- You can use the skeleton code given below:

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity alu_beh is
    port (
        A: in std_logic_vector(3 downto 0);
        B: in std_logic_vector(3 downto 0);
        sel: in std_logic_vector(1 downto 0);
        op: out std_logic_vector(7 downto 0)
    ) ;
end alu_beh;

architecture a1 of alu_beh is

    function sub(A: in std_logic_vector(3 downto 0); B: in std_logic_vector(3 downto 0))
        return std_logic_vector is
            -- declaring and initializing variables using aggregates
            variable diff : std_logic_vector(.....):=(others =>'0');
            variable carry: std_logic_vector(.....):=(others =>'0');
        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;

begin
alu : process( A, B, sel )
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
```