ECE-223, Solutions for Assignment #5

---

1. Write a dataflow VHDL code for the full-adder.

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY fulladd IS
    PORT (      Cin, x, y       : IN    STD_LOGIC ;
                s, Cout         : OUT   STD_LOGIC ) ;
    END fulladd;

ARCHITECTURE dataflow of fulladd IS
 BEGIN
  S <= Cin XOR x XOR y;
  Cout <=  (Cin AND x) OR (Cin AND y) OR (x AND y);

END dataflow ;
```

---

2. Write a behavioural VHDL code for a 2-to-4 binary decoder.

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY dec2to4 IS
    PORT (      w       : IN    STD_LOGIC_VECTOR(1 DOWNTO 0) ;
                En      : IN    STD_LOGIC ;
                y       : OUT   STD_LOGIC_VECTOR(0 TO 3) ) ;
END dec2to4 ;

ARCHITECTURE Behavior OF dec2to4 IS
    SIGNAL Enw : STD_LOGIC_VECTOR(2 DOWNTO 0) ;
BEGIN
    Enw <= En & w ;
    WITH Enw SELECT
        y <=    "1000" WHEN "100",
                       "0100" WHEN "101",
                       "0010" WHEN "110",
                       "0001" WHEN "111",
                       "0000" WHEN OTHERS ;
END Behavior ;
```

3. Write a behavioural VHDL code for a 2-to1 multiplexer using if-then-else statement.

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY mux2to1 IS
        PORT (          w0, w1, s       : IN    STD_LOGIC ;
                        f                       : OUT  STD_LOGIC ) ;
END mux2to1 ;

ARCHITECTURE Behavior OF mux2to1 IS
BEGIN
        PROCESS ( w0, w1, s )
        BEGIN
                IF s = '0' THEN
                        f <= w0 ;
                ELSE
                        f <= w1 ;
                END IF ;
        END PROCESS ;
END Behavior ;
```

---

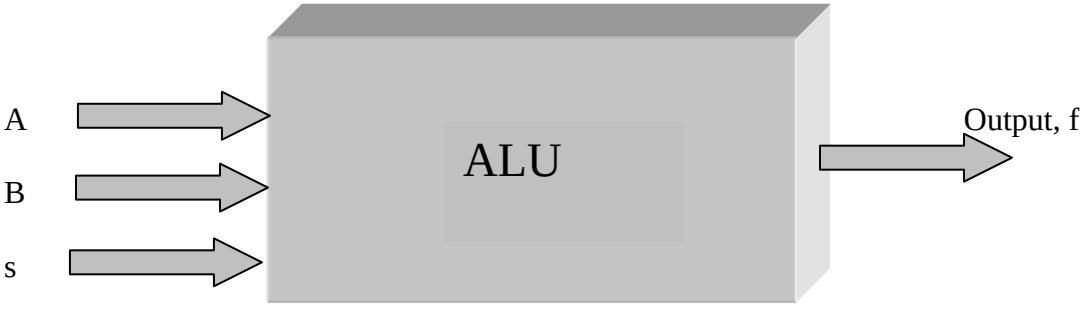
4. Write a behavioural VHDL code for a 2-to1 multiplexer case statement.

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY mux2to1 IS
    PORT (          w0, w1, s       : IN    STD_LOGIC ;
                    f                       : OUT  STD_LOGIC ) ;
END mux2to1 ;
ARCHITECTURE Behavior OF mux2to1 IS
BEGIN
    PROCESS ( w0, w1, s )
    BEGIN
            CASE s IS
                    WHEN '0' =>
                            f <= w0 ;
                    WHEN OTHERS =>
                            f <= w1 ;
            END CASE ;
    END PROCESS ;
END Behavior ;
```

5. Write a behavioural VHDL code for a 3-bit ALU. Functionality of the ALU is shown in Table-1 also A and B are data inputs.

Operation	Control Input	Output
Clear	000	0000
B-A	001	B-A
A-B	010	A-B
ADD	011	A+B
XOR	100	A XOR B
OR	101	A OR B
AND	110	A AND B
Preset	111	1111




```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
USE ieee.std_logic_unsigned.all ;

ENTITY alu IS
        PORT (      s          : IN     STD_LOGIC_VECTOR(2 DOWNTO 0) ;
                    A, B  : IN     STD_LOGIC_VECTOR(3 DOWNTO 0) ;
                    F          : OUT  STD_LOGIC_VECTOR(3 DOWNTO 0) ) ;
END alu ;

ARCHITECTURE Behavior OF alu IS
BEGIN
        PROCESS ( s, A, B )
        BEGIN
                CASE s IS
                        WHEN "000" =>
                                F <= "0000" ;
```

```
                WHEN "001" =>
                        F <= B - A ;
                WHEN "010" =>
                        F <= A - B ;
                WHEN "011" =>
                        F <= A + B ;
                WHEN "100" =>
                        F <= A XOR B ;
                WHEN "101" =>
                        F <= A OR B ;
                WHEN "110" =>
                        F <= A AND B ;
                WHEN OTHERS =>
                        F <= "1111" ;
        END CASE ;
    END PROCESS ;
END Behavior ;
```