Name: Siddhartha Dhar

Stream: BTech CSE 4th Semester **University Roll:** T91/CSE/196010 **Class Roll:** 26

Sub: Computer Architecture Lab (VHDL programming)

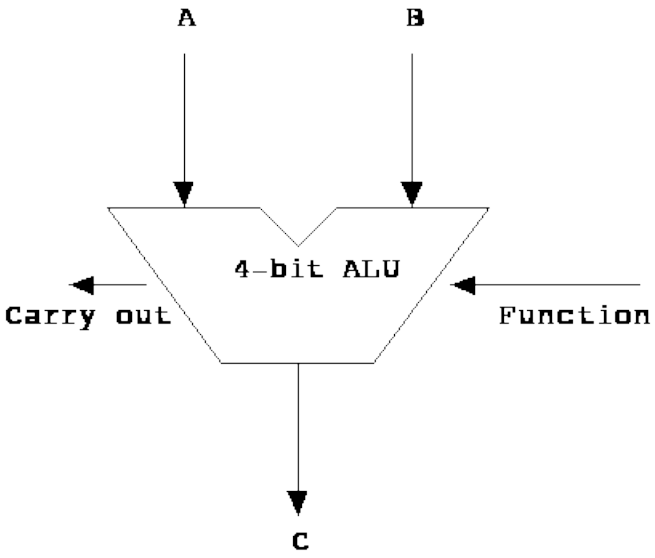
Problem Statement:

Write a VHDL Program to construct a 4bit ALU (Arithmetic and Logic Unit) using Structural Modeling

Environment Used : *Xilinx 14.7 ISE Design Suite*

Design :

Let see the Abstract ALU Diagram

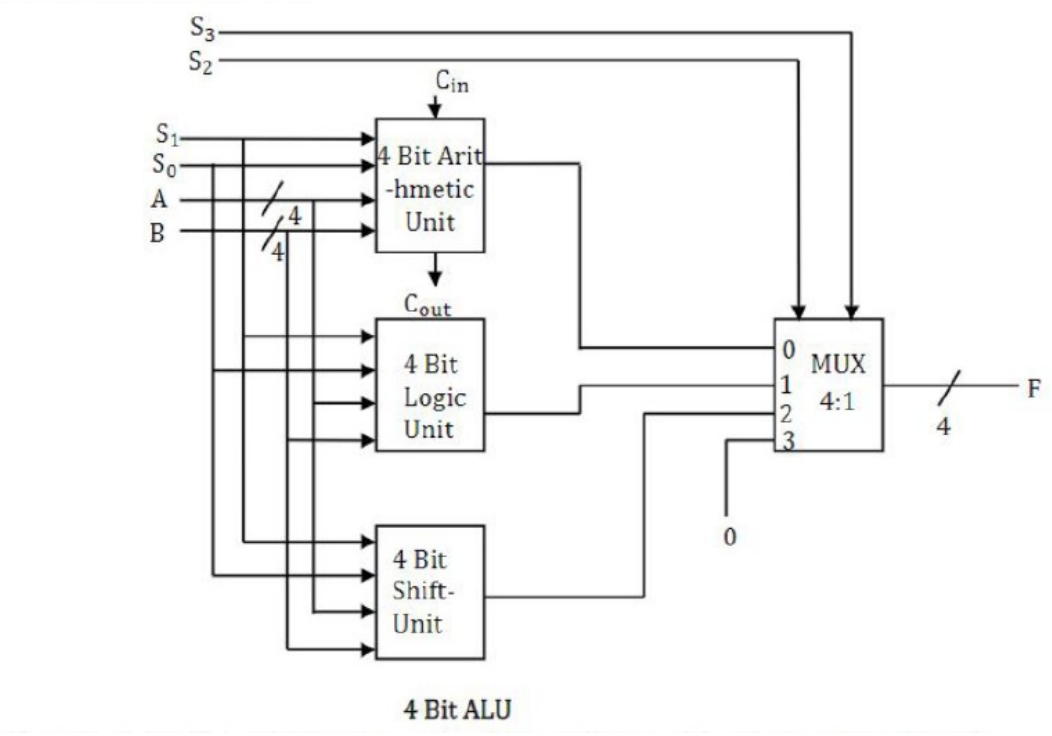


As it is a 4-bit ALU therefore the Input two operands **'A'** and **'B'** are 4 bits wide operands and the Output of the ALU is also 4 bits wide **'C'** Function is nothing but the Control Signal that instructs the ALU to perform certain Operation

The Set of permissible Operation Considered in this project is listed below, Here S_3,S_2,S_1,S_0,C_{in} are the set of Control signals that defines the operation to be performed

S3	S2	S1	S0	Cin	Operation
0	0	0	0	0	(A+B) Addition
0	0	0	0	1	(A+B+1) Addition with Carry
0	0	0	1	0	(A+B') Subtract with Borrow
0	0	0	1	1	(A+B'+1) Subtraction
0	0	1	0	0	(A-1) Decrement
0	0	1	0	1	(A) Transfer
0	0	1	1	0	(A) Transfer
0	0	1	1	1	(A+1) Increment
0	1	0	0	-	A and B (bitwise)
0	1	0	1	-	A or B (bitwise)
0	1	1	0	-	A xor B (bitwise)
0	1	1	1	-	A' (Complement Bitwise)
1	0	0	0	-	Logical Left Shift A
1	0	0	1	-	Logical Right Shift A
1	0	1	0	-	Arithmetic Left Shift A
1	0	1	1	-	Arithmetic Right Shift A

Now lets see a more detailed diagram of the 4 bit ALU



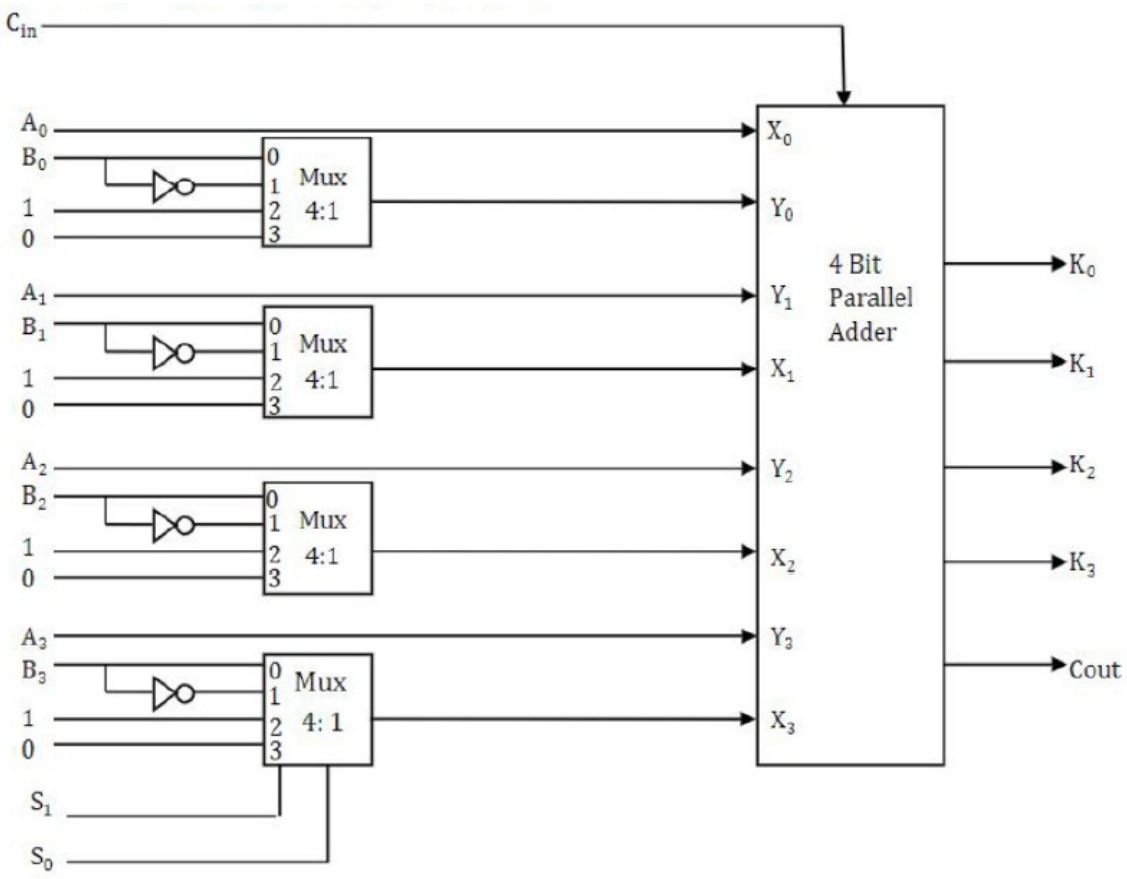
Here the Final Multiplexer is a bit special instead of single line input the multiplexer has 4-Bit Buses as Inputs and the Multiplexer outputs an 4 bit Bus

As there is 3 Individual Components :

1. **Arithmetic Unit**
2. **4-Bit Logic Unit-01**
3. **4-Bit Logic Unit-02**

Now let see the detailed Circuits of this Individual Units

1. Arithmetic Unit :



The Circuit required 4-bit parallel adder and four 4:1 multiplexers. There are two four bit inputs i.e. A & B. the inputs to the each multiplexer are B, B', 1, 0. According to the selection lines S_1, S_0, the input of multiplexer are transferred to the parallel adder input Y's, the final output of this arithmetic units is

$$K = A + Y + C_{in}$$

Where A is a bit number, Y is the 4 bit o/p of multiplexer and Cin is the carry input to the parallel adder, By above circuit it is possible to implement.

7 arithmetic micro operation

Case 1: when $S_1S_0 = 00$
If $C_{in} = 0$, then K = A + B i.e. Addition without Carry.
If $C_{in} = 1$, then K = A + B + 1 i.e. Addition with Carry.

Case 2: when $S_1S_0 = 01$
If $C_{in} = 0$, then K = A + B' i.e. Subtraction with Borrow.
If $C_{in} = 1$, then K = A + B' + 1 i.e. Subtraction A-B is performed.

Case 3: when $S_1S_0 = 10$
If $C_{in} = 0$, then K = A - 1 i.e. Decrement Operation.
If $C_{in} = 1$, then K = A i.e. Direct Transfer from A to K.

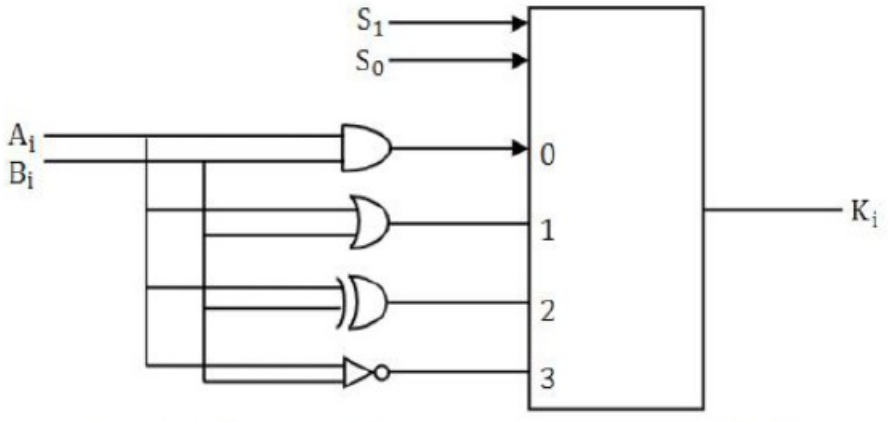
Case 4: when $S_1S_0 = 11$
If $C_{in} = 0$, then K = A i.e. Direct Transfer from A to K.
If $C_{in} = 1$, then K = A + 1 i.e. Increment Operation.

1. 4-Bit Logic Unit-01 :

Logic Unit is need to perform the logical micro operation such that OR, AND, XOR, Complement etc. on individual pair of bits stored in registers

$R_1 \leftarrow R_2$ (AND) R_3 denotes AND micro operation
$R_1 \leftarrow R_2$ (OR) R_3 denotes OR micro operation
$R_1 \leftarrow R_2$ (XOR) R_3 denotes XOR micro operation
$R_1 \leftarrow R_2$' denotes NOT or Complement micro operation

The diagram of the logic unit which can perform logic micro operation AND, OR, XOR, NOT is given below.



Here the multiplexer takes 4bit Buses as inputs that is the Input 0, 1, 2, 3 are the 4bit Bus Inputs, Similarly the Shown gates are also of 8 bit inputs and 4 bit outputs for every A_i and B_i

3. 4-Bit Logic Unit-02 :

The Logic Unit 2 performs Logical Left Shift , Logical Right Shift, Arithmetic Left Shift, and Arithmetic Right Shift.

Note that the we can add a 4th unit to the ALU here it is made reserved for future use.

VHDL Code for 4bit ALU :

```
----------------------------------------------------------------------------------
-- University:     University of Calcutta
-- Engineer:       Siddhartha Dhar
--
-- Module Name:    ALU_4bit - Behavioral

-- +-----------------------------------------------------+
-- | S3  S2  S1  S0  Cin |         Operation             |
-- +-----------------------------------------------------+
-- |  0   0   0   0   0  |      (A+B) Addition           |
-- +-----------------------------------------------------+
-- |  0   0   0   0   1  | (A+B+1) Addition with Carry   |
-- +-----------------------------------------------------+
-- |  0   0   0   1   0  | (A+B') Subtract with Borrow   |
-- +-----------------------------------------------------+
-- |  0   0   0   1   1  |    (A+B'+1) Subtraction       |
-- +-----------------------------------------------------+
-- |  0   0   1   0   0  |       (A-1) Decrement         |
-- +-----------------------------------------------------+
-- |  0   0   1   0   1  |         (A) Transfer          |
-- +-----------------------------------------------------+
-- |  0   0   1   1   0  |         (A) Transfer          |
-- +-----------------------------------------------------+
-- |  0   0   1   1   1  |       (A+1) Increment         |
-- +-----------------------------------------------------+
-- |  0   1   0   0   -  |       A and B (bitwise)       |
-- +-----------------------------------------------------+
-- |  0   1   0   1   -  |        A or B (bitwise)       |
-- +-----------------------------------------------------+
-- |  0   1   1   0   -  |       A xor B (bitwise)       |
-- +-----------------------------------------------------+
-- |  0   1   1   1   -  |    A' (Complement Bitwise)    |
-- +-----------------------------------------------------+
-- |  1   0   0   0   -  |    Logical Left Shift A       |
-- +-----------------------------------------------------+
-- |  1   0   0   1   -  |   Logical Right Shift A       |
-- +-----------------------------------------------------+
-- |  1   0   1   0   -  |   Arithmetic Left Shift A     |
-- +-----------------------------------------------------+
-- |  1   0   1   1   -  |   Arithmetic Right Shift A    |
-- +-----------------------------------------------------+
```

```
--------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-----------------------------------------------------------------Entity Definition
entity ALU_4bit is
    Port ( Cin : in STD_LOGIC;
                      A : in  STD_LOGIC_VECTOR (3 downto 0);
           B : in  STD_LOGIC_VECTOR (3 downto 0);
                      S : in  STD_LOGIC_VECTOR(3 downto 0);
           Cout : out  STD_LOGIC;
           Z : out  STD_LOGIC_VECTOR (3 downto 0));
end ALU_4bit;

-----------------------------------------------------------------Architecture Definition
architecture ALU_STRCTL of ALU_4bit is
-----------------------------------------------------------------Arithmetic Unit
component Arithmetic_Unit
    Port ( Cin : in STD_LOGIC;
                      A : in  STD_LOGIC_VECTOR (3 downto 0);
           B : in  STD_LOGIC_VECTOR (3 downto 0);
                      S : in  STD_LOGIC_VECTOR(1 downto 0);
           Cout : out  STD_LOGIC;
           Sum : out  STD_LOGIC_VECTOR (3 downto 0));
end component;
-----------------------------------------------------------------Logic_Unit_01
component Logic_Unit_01
    Port ( A : in  STD_LOGIC_VECTOR (3 downto 0);
           B : in  STD_LOGIC_VECTOR (3 downto 0);
           S : in  STD_LOGIC_VECTOR(1 downto 0);
           Y : out  STD_LOGIC_VECTOR (3 downto 0));
end component;
-----------------------------------------------------------------Logic_Unit_02
component Logic_Unit_02
    Port ( A : in  STD_LOGIC_VECTOR (3 downto 0);
           S : in  STD_LOGIC_VECTOR(1 downto 0);   -- Select Lines
           Y : out  STD_LOGIC_VECTOR (3 downto 0)); -- Output 4bit Bus
end component;
-----------------------------------------------------------------Bus_MUX_4X1
component BUS_MUX_4X1
    Port ( I0 : in  STD_LOGIC_VECTOR (3 downto 0);  -- First Input 4bit Bus
           I1 : in  STD_LOGIC_VECTOR (3 downto 0);  -- Second Input 4bit Bus
           I2 : in  STD_LOGIC_VECTOR (3 downto 0);  -- Third Input 4bit Bus
           I3 : in  STD_LOGIC_VECTOR (3 downto 0);  -- Fourth Input 4bit Bus
           S : in  STD_LOGIC_VECTOR(1 downto 0);   -- Select Lines
           Y : out  STD_LOGIC_VECTOR (3 downto 0)); -- Output 4bit Bus
end component;
Signal F0, F1, F2: STD_LOGIC_VECTOR(3 downto 0);
begin

      T0: Arithmetic_Unit port map(Cin=>Cin, A=>A, B=>B, S(1)=>S(1), S(0)=>S(0),
            Cout=>Cout, Sum=>F0); -- Arithmetic Unit Component
      T1: Logic_Unit_01 port map(A=>A, B=>B, S(1)=>S(1), S(0)=>S(0), Y=>F1);
      ----------------------------- Logical Unit 01
      T2: Logic_Unit_02 port map(A=>A, S(1)=>S(1), S(0)=>S(0), Y=>F2);
      ---------------------------------- Logical Unit 02
      T3: BUS_MUX_4X1 port map(I0=>F0, I1=>F1, I2=>F2, I3=>"0000", S(1)=>S(3), S(0)=>S(2),
            Y=>Z); ------- Combining the three Units using to an Multiplexer

end ALU_STRCTL;
```

VHDL Code for Arithmetic unit :

```
----------------------------------------------------------------------------------
-- University:     University of Calcutta
-- Engineer:       Siddhartha Dhar
--
-- Module Name:    Arithmetic_Unit - Behavioral
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-------------------------------------------------------Entity Definition
entity Arithmetic_Unit is
    Port ( Cin : in STD_LOGIC;
                    A : in  STD_LOGIC_VECTOR (3 downto 0);
           B : in  STD_LOGIC_VECTOR (3 downto 0);
                    S : in  STD_LOGIC_VECTOR(1 downto 0);
           Cout : out  STD_LOGIC;
           Sum : out  STD_LOGIC_VECTOR (3 downto 0));
end Arithmetic_Unit;

-------------------------------------------------------Arithmetic Definition
architecture ARTH_UNIT_STRCTL of Arithmetic_Unit is
-------------------------------------------------------4 bit Adder Component
component Adder_4bit
    Port ( Cin : in STD_LOGIC;
                    A : in  STD_LOGIC_VECTOR (3 downto 0);
           B : in  STD_LOGIC_VECTOR (3 downto 0);
           Cout : out  STD_LOGIC;
           Sum : out  STD_LOGIC_VECTOR (3 downto 0));
end component;
-------------------------------------------------------4X1 Multiplexer Component
component MUX_4X1_DF
      port(I: in std_logic_vector(0 to 3);
        S: in std_logic_vector(0 to 1);
        Y: out std_logic);
end component;
signal Bb, f : STD_LOGIC_VECTOR(3 downto 0);
begin
      Bb <= not B;
      T0: MUX_4X1_DF port map(I(0)=>B(0), I(1)=>Bb(0), I(2)=>'1', I(3)=>'0', S(0)=>S(1),
            S(1)=>S(0), Y=>f(0));
      T1: MUX_4X1_DF port map(I(0)=>B(1), I(1)=>Bb(1), I(2)=>'1', I(3)=>'0', S(0)=>S(1),
            S(1)=>S(0), Y=>f(1));
      T2: MUX_4X1_DF port map(I(0)=>B(2), I(1)=>Bb(2), I(2)=>'1', I(3)=>'0', S(0)=>S(1),
            S(1)=>S(0), Y=>f(2));
      T3: MUX_4X1_DF port map(I(0)=>B(3), I(1)=>Bb(3), I(2)=>'1', I(3)=>'0', S(0)=>S(1),
            S(1)=>S(0), Y=>f(3));
      T4: Adder_4bit port map(Cin=>Cin, A=>A, B=>f, Cout=>Cout, Sum=>Sum);

end ARTH_UNIT_STRCTL;
```

VHDL Code for Logical Unit 01 :

```
----------------------------------------------------------------------------------
-- University:     University of Calcutta
-- Engineer:       Siddhartha Dhar
--
-- Module Name:    Logic_Unit_01 - Behavioral
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-------------------------------------------------Entity Definition
entity Logic_Unit_01 is
    Port ( A : in  STD_LOGIC_VECTOR (3 downto 0);  -- First Input 4bit Bus
           B : in  STD_LOGIC_VECTOR (3 downto 0);  -- Second Input 4bit Bus
           S : in  STD_LOGIC_VECTOR(1 downto 0);   -- Select Lines
           Y : out  STD_LOGIC_VECTOR (3 downto 0)); -- Output 4bit Bus
end Logic_Unit_01;

-------------------------------------------------Architecture Definition
architecture Behavioral of Logic_Unit_01 is
-----------------------------------------------4 bit Bus Multiplexer
component BUS_MUX_4X1
    Port ( I0 : in  STD_LOGIC_VECTOR (3 downto 0);  -- First Input 4bit Bus
           I1 : in  STD_LOGIC_VECTOR (3 downto 0);  -- Second Input 4bit Bus
           I2 : in  STD_LOGIC_VECTOR (3 downto 0);  -- Third Input 4bit Bus
           I3 : in  STD_LOGIC_VECTOR (3 downto 0);  -- Fourth Input 4bit Bus
           S : in  STD_LOGIC_VECTOR(1 downto 0);   -- Select Lines
           Y : out  STD_LOGIC_VECTOR (3 downto 0)); -- Output 4bit Bus
end component;
-------------------------------------------------------------Signal Decleration
signal Annd, Orr, Xoor, Noot: STD_LOGIC_VECTOR(3 downto 0);
begin
      Annd <= (A and B);
      Orr <= (A or B);
      Xoor <= (A xor B);
      Noot <= (not A);
      T0: BUS_MUX_4X1 port map(I0=>Annd, I1=>Orr, I2=>Xoor, I3=>Noot, S=>S, Y=>Y);
end Behavioral;
```

VHDL Code for Logical Unit 02 :

```
----------------------------------------------------------------------------------
-- University:     University of Calcutta
-- Engineer:       Siddhartha Dhar
--
-- Module Name:    Logic_Unit_02 - Behavioral
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-----------------------------------------------Entity Definition
entity Logic_Unit_02 is
    Port ( A : in  STD_LOGIC_VECTOR (3 downto 0);
           S : in  STD_LOGIC_VECTOR(1 downto 0);   -- Select Lines
           Y : out  STD_LOGIC_VECTOR (3 downto 0)); -- Output 4bit Bus
end Logic_Unit_02;

-----------------------------------------------Architecture Definition
architecture LU_02_STRCTL of Logic_Unit_02 is
-----------------------------------------------BUS Multiplexer 4bit width
component BUS_MUX_4X1
    Port ( I0 : in  STD_LOGIC_VECTOR (3 downto 0);  -- First Input 4bit Bus
           I1 : in  STD_LOGIC_VECTOR (3 downto 0);  -- Second Input 4bit Bus
           I2 : in  STD_LOGIC_VECTOR (3 downto 0);  -- Third Input 4bit Bus
           I3 : in  STD_LOGIC_VECTOR (3 downto 0);  -- Fourth Input 4bit Bus
           S : in  STD_LOGIC_VECTOR(1 downto 0);   -- Select Lines
           Y : out  STD_LOGIC_VECTOR (3 downto 0)); -- Output 4bit Bus
end component;
-----------------------------------------------Logical Left Shift 4bit
component Logical_Left_Shift_4bit
    Port ( A : in  STD_LOGIC_VECTOR (3 downto 0);
           Y : out  STD_LOGIC_VECTOR (3 downto 0));
end component;
-----------------------------------------------Logical Right Shift 4bit
component Logical_Right_Shift_4bit
    Port ( A : in  STD_LOGIC_VECTOR (3 downto 0);
           Y : out  STD_LOGIC_VECTOR (3 downto 0));
end component;
-----------------------------------------------Arithmetic RIght Shift
component Arithmetic_Right_Shift_4bit
    Port ( A : in  STD_LOGIC_VECTOR (3 downto 0);
           Y : out  STD_LOGIC_VECTOR (3 downto 0));
end component;
Signal F0, F1, F2, F3: STD_LOGIC_VECTOR(3 downto 0);
begin

      T0: Logical_Left_Shift_4bit port map(A=>A, Y=>F0); -- Logical Left Shift Component
      T1: Logical_Right_Shift_4bit port map(A=>A, Y=>F1);-- Logical Right Shift Component
      T2: Logical_Left_Shift_4bit port map(A=>A, Y=>F2);
      T3: Arithmetic_Right_Shift_4bit port map(A=>A, Y=>F3); -- Arithmetic Right Shift
                                                                Component
      T4: BUS_MUX_4X1 port map(I0=>F0, I1=>F1, I2=>F2, I3=>F3, S=>S, Y=>Y);

end LU_02_STRCTL;
```

VHDL Code for 4bit Ripple Carry Adder :

```
----------------------------------------------------------------------------------
-- University:     University of Calcutta
-- Engineer:       Siddhartha Dhar
--
-- Module Name:    Adder_4bit - Behavioral
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-------------------------------------------Entity Definition
entity Adder_4bit is
    Port ( Cin : in STD_LOGIC;
                     A : in  STD_LOGIC_VECTOR (3 downto 0);
           B : in  STD_LOGIC_VECTOR (3 downto 0);
           Cout : out  STD_LOGIC;
           Sum : out  STD_LOGIC_VECTOR (3 downto 0));
end Adder_4bit;

-------------------------------------------Architecture Definition
architecture ADDR_STRCTL of Adder_4bit is
component Full_Adder
    Port ( A : in  STD_LOGIC;
           B : in  STD_LOGIC;
           C : in  STD_LOGIC;
           Carry : out  STD_LOGIC;
           Sum : out  STD_LOGIC);
end component;
signal Ctmp : STD_LOGIC_VECTOR(2 downto 0);
begin
       T0: Full_Adder port map(A=>A(0), B=>B(0), C=>Cin, Carry=>Ctmp(0), Sum=>Sum(0));
             ------ Full Adder for LSb
       T1: Full_Adder port map(A=>A(1), B=>B(1), C=>Ctmp(0), Carry=>Ctmp(1), Sum=>Sum(1));
             -- Full Adder for 2nd higher order bit
       T2: Full_Adder port map(A=>A(2), B=>B(2), C=>Ctmp(1), Carry=>Ctmp(2), Sum=>Sum(2));
             -- Full Adder for 3rd higher order bit
       T3: Full_Adder port map(A=>A(3), B=>B(3), C=>Ctmp(2), Carry=>Cout, Sum=>Sum(3));
             ------ Full Adder for MSb
end ADDR_STRCTL;
```

VHDL Code Full Adder :

```
----------------------------------------------------------------------------------
-- University:     University of Calcutta
-- Engineer:       Siddhartha Dhar
--
-- Module Name:    Full_Adder - Behavioral
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-------------------------------------------------Entity Definition
entity Full_Adder is
    Port ( A : in  STD_LOGIC;
           B : in  STD_LOGIC;
           C : in  STD_LOGIC;
           Carry : out  STD_LOGIC;
           Sum : out  STD_LOGIC);
end Full_Adder;

-------------------------------------------------Architecture Definition
architecture FULL_ADDER_DFLOW of Full_Adder is
signal t: STD_LOGIC_VECTOR(2 downto 0);
begin
      t(0) <= (A xor B);
      Sum <= (t(0) xor C);
      t(1) <= (t(0) and C);
      t(2) <= (A and B);
      Carry <= (t(1) or t(2));
end FULL_ADDER_DFLOW;
```

VHDL Code for Logical Left Shift :

```
----------------------------------------------------------------------------------
-- University:     University of Calcutta
-- Engineer:       Siddhartha Dhar
--
-- Module Name:    Logical_Left_Shift_4bit - Behavioral
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-----------------------------------------------------------Entity Definition
entity Logical_Left_Shift_4bit is
    Port ( A : in  STD_LOGIC_VECTOR (3 downto 0);
           Y : out  STD_LOGIC_VECTOR (3 downto 0));
end Logical_Left_Shift_4bit;
-----------------------------------------------------------Architecture Defintion
architecture LLS_4_DFLOW of Logical_Left_Shift_4bit is
Signal f: STD_LOGIC_VECTOR(3 downto 0);
begin
      f(0)<='0'; -- O inserted from left
      f(1)<=A(0);
      f(2)<=A(1);
      f(3)<=A(2);
      Y <= f;
end LLS_4_DFLOW;
```

VHDL Code for Logical Right Shift :

```
----------------------------------------------------------------------------------
-- University:     University of Calcutta
-- Engineer:       Siddhartha Dhar
--
-- Module Name:    Logical_Left_Shift_4bit - Behavioral
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-------------------------------------------------Entity Definition
entity Logical_Right_Shift_4bit is
    Port ( A : in  STD_LOGIC_VECTOR (3 downto 0);
           Y : out  STD_LOGIC_VECTOR (3 downto 0));
end Logical_Right_Shift_4bit;

-----------------------------------------------------------Architecture Defintion
architecture LRS_4_DFLOW of Logical_Right_Shift_4bit is
Signal f: STD_LOGIC_VECTOR(3 downto 0);
begin
       f(0)<=A(1);
       f(1)<=A(2);
       f(2)<=A(3);
       f(3)<='0'; -- 0 inserted from the Right side
       Y <= f;
end LRS_4_DFLOW;
```

VHDL Code for Arithmetic Right Shift :

```
----------------------------------------------------------------------------------
-- University:     University of Calcutta
-- Engineer:       Siddhartha Dhar
--
-- Module Name:    Arithmetic_Left_Shift_4bit - Behavioral
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-------------------------------------------------------------------Entity Definition
entity Arithmetic_Right_Shift_4bit is
    Port ( A : in  STD_LOGIC_VECTOR (3 downto 0);
           Y : out  STD_LOGIC_VECTOR (3 downto 0));
end Arithmetic_Right_Shift_4bit;

-----------------------------------------------------------------Architecture Definition
architecture ARTH_R_SHFT_DFLOW of Arithmetic_Right_Shift_4bit is
signal f: STD_LOGIC_VECTOR(3 downto 0);
begin
       f(3) <= A(3);
       f(2) <= A(3);
       f(1) <= A(2);
       f(0) <= A(1);
       Y <= f;
end ARTH_R_SHFT_DFLOW;
```

VHDL 4X1 Bus MUX :

```
----------------------------------------------------------------------------------
-- University:     University of Calcutta
-- Engineer:       Siddhartha Dhar
--
-- Module Name:    BUS_MUX_4X1 - Behavioral
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-------------------------------------------------------Entity Definition
entity BUS_MUX_4X1 is
    Port ( I0 : in  STD_LOGIC_VECTOR (3 downto 0);  -- First Input 4bit Bus
           I1 : in  STD_LOGIC_VECTOR (3 downto 0);  -- Second Input 4bit Bus
           I2 : in  STD_LOGIC_VECTOR (3 downto 0);  -- Third Input 4bit Bus
           I3 : in  STD_LOGIC_VECTOR (3 downto 0);  -- Fourth Input 4bit Bus
           S : in  STD_LOGIC_VECTOR(1 downto 0);   -- Select Lines
           Y : out  STD_LOGIC_VECTOR (3 downto 0)); -- Output 4bit Bus
end BUS_MUX_4X1;

-------------------------------------------------------Architecture Definition
architecture Behavioral of BUS_MUX_4X1 is
begin

process(S, I0, I1, I2, I3)
begin
             if(S="00")then
                    Y <= I0;
             elsif(S="01")then
                    Y <= I1;
             elsif(S="10")then
                    Y <= I2;
             else
                    Y <= I3;
             end if;
end process;

end Behavioral;
```

VHDL Test Bench for 4bit ALU :

```
--------------------------------------------------------------------------------
-- University:   University of Calcutta
-- Engineer:     Siddhartha Dhar
--
-- Project Name:  ALU_4bit_STRCTL
--------------------------------------------------------------------------------
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

----------------------------------------Test Bench Entity
ENTITY ALU_4bit_TB IS
END ALU_4bit_TB;
----------------------------------------Architecture Definition
ARCHITECTURE behavior OF ALU_4bit_TB IS

    -- Component Declaration for the Unit Under Test (UUT)
    COMPONENT ALU_4bit
    PORT(
         Cin : IN  std_logic;
         A : IN  std_logic_vector(3 downto 0);
         B : IN  std_logic_vector(3 downto 0);
         S : IN  std_logic_vector(3 downto 0);
         Cout : OUT  std_logic;
         Z : OUT  std_logic_vector(3 downto 0)
        );
    END COMPONENT;

   --Inputs
   signal Cin : std_logic := '0';
   signal A : std_logic_vector(3 downto 0) := (others => '0');
   signal B : std_logic_vector(3 downto 0) := (others => '0');
   signal S : std_logic_vector(3 downto 0) := (others => '0');

      --Outputs
   signal Cout : std_logic;
   signal Z : std_logic_vector(3 downto 0);

BEGIN
       -- Instantiate the Unit Under Test (UUT)
   uut: ALU_4bit PORT MAP (
          Cin => Cin,
          A => A,
          B => B,
          S => S,
          Cout => Cout,
          Z => Z
        );
```

```
   -- Stimulus process
   stim_proc: process
   begin
      -- hold reset state for 100 ns.
      wait for 10 ns;
      -- insert stimulus here
             Cin<='0';
             A <= "1010";
             B <= "0100";
             S <= "0000"; -- A+B
             wait for 10 ns;
             Cin <= '1';
             S <= "0011"; -- A+1
             wait for 10 ns;
             S <= "0100";
             wait for 10 ns;
             S <= "1000";
             wait for 10 ns;
             S <= "1011";
             wait for 10 ns;
   end process;

END;
```