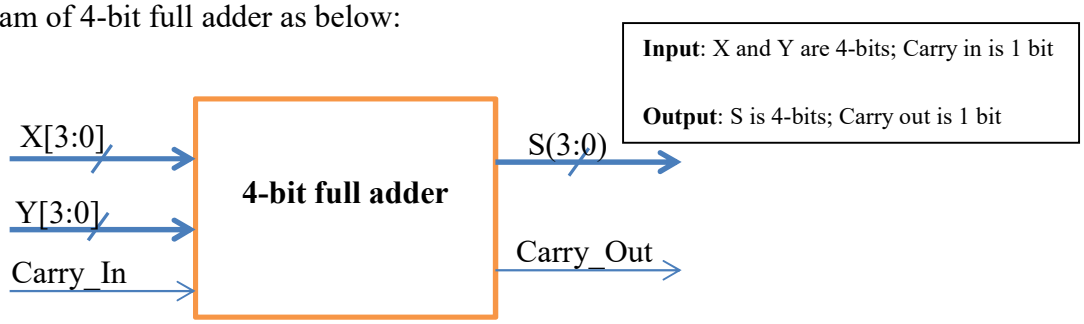
Lab #4: 4-Bit Full Adder (Structural Implementation Style)

1. Requirement

Design a 4-bit full adder circuit from 1-bit full adder circuits using structural implementation style. First, design a 1-bit full adder circuit (FA) in data flow implementation style, then use this component as building blocks in your 4-bit full adder.

Block diagram of 4-bit full adder as below:



You can use the following entity declaration for your design:

```
entity Full_Adder_4bits is
      port(
            X: in std_logic_vector(3 downto 0);
            Y: in std_logic_vector(3 downto 0);
            Carry_In: in std_logic;
            S: out std_logic_vector(3 downto 0);
            Carry_Out: out std_logic
      );
end Full_Adder_4bits;
```

2. Pre-lab

- Study and analyze the working of 1-bit full adder (what is 1-bit full adder, block diagram, its input/output, etc…)
- Write the truth table for 1-bit full adder and simplify expressions for *sum* and *carry out*
- Draw the circuit diagram of your 4-bit full adder in terms of 1-bit full adder

3. Lab

- Write the VHDL code for 1-bit full adder using ***data flow implementation style*** (using logic operations like: *XOR, AND, OR*, …)
- Write the VHDL code for 4-bit full adder from 1-bit full adder using ***structural implementation style*** (using "*port map*" and "*component*" statement)
- Simulate using Xilinx ISE simulator

4. Deliverables

- VHDL program for 1-bit full adder, 4-bit full adder
- VHDL test bench for 4-bit full adder, the testbench should cover all of the cases.
- Waveform for 4-bit full adder