EXPERIMENT No.: - 9

Simulation of VHDL Models.

TITLE:

Implementation of 4:1 MUX using 2:1 MUX in VHDL using Structural Modelling.

OBJECTIVE:

To implement the Structural model of 4:1 MUX using 2:1 MUX using ISE Simulator.

THEORY:

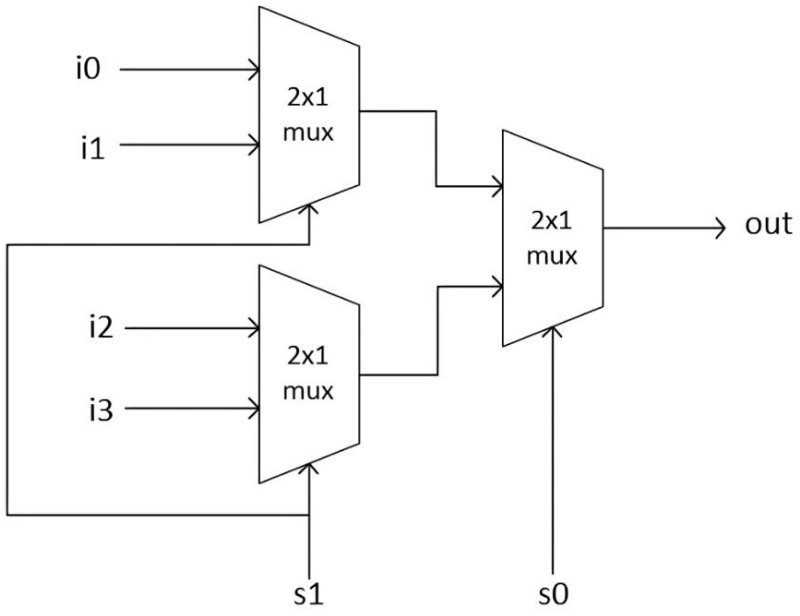
A multiplexer is a device that selects one output from multiple inputs. It is also known as a data selector. We refer to a multiplexer with the terms **MUX.**

Multiplexers are used in communication systems to increase the amount of data sent over a network within a certain amount of time and bandwidth. It allows us to squeeze multiple data lines into one data line. It switches between one of the many input lines and combines them one by one to the output. It decides which input line to switch using a control signal.

Physically, a multiplexer has **n** input pins, one output pin, and **m** control pins. $\mathbf{n = 2^m}$.

So, for constructing a 2:1 MUX there will be 2 input lines and 1 selection line which will result in one output line as a whole. Comparatively, to construct a 4:1 MUX we need 4 input lines with 2 selection lines which will result in one output line as a whole.

So, in order to construct the 4:1 MUX using 2:1 MUX we will need three 2:1 MUX as shown in the figure below –



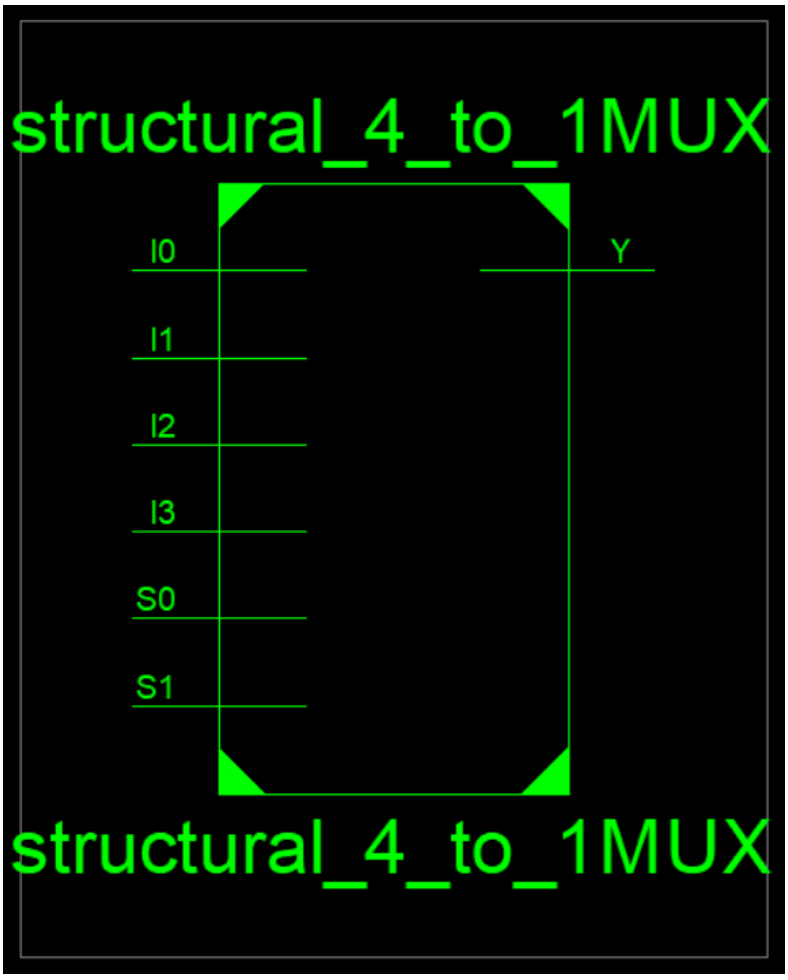
SOURCE CODE: -

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

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity structural_4_to_1MUX is
    Port ( I0 : in  STD_LOGIC;
           I1 : in  STD_LOGIC;
           I2 : in  STD_LOGIC;
           I3 : in  STD_LOGIC;
           S1 : in  STD_LOGIC;
           S0 : in  STD_LOGIC;
           Y : out  STD_LOGIC);
end structural_4_to_1MUX;

architecture STRUCTURAL of structural_4_to_1MUX is

COMPONENT mux_block is
        Port(inp1, inp2, s : in std_logic;
                    X   : out std_logic);
end COMPONENT;

SIGNAL o1, o2 : std_logic;
```

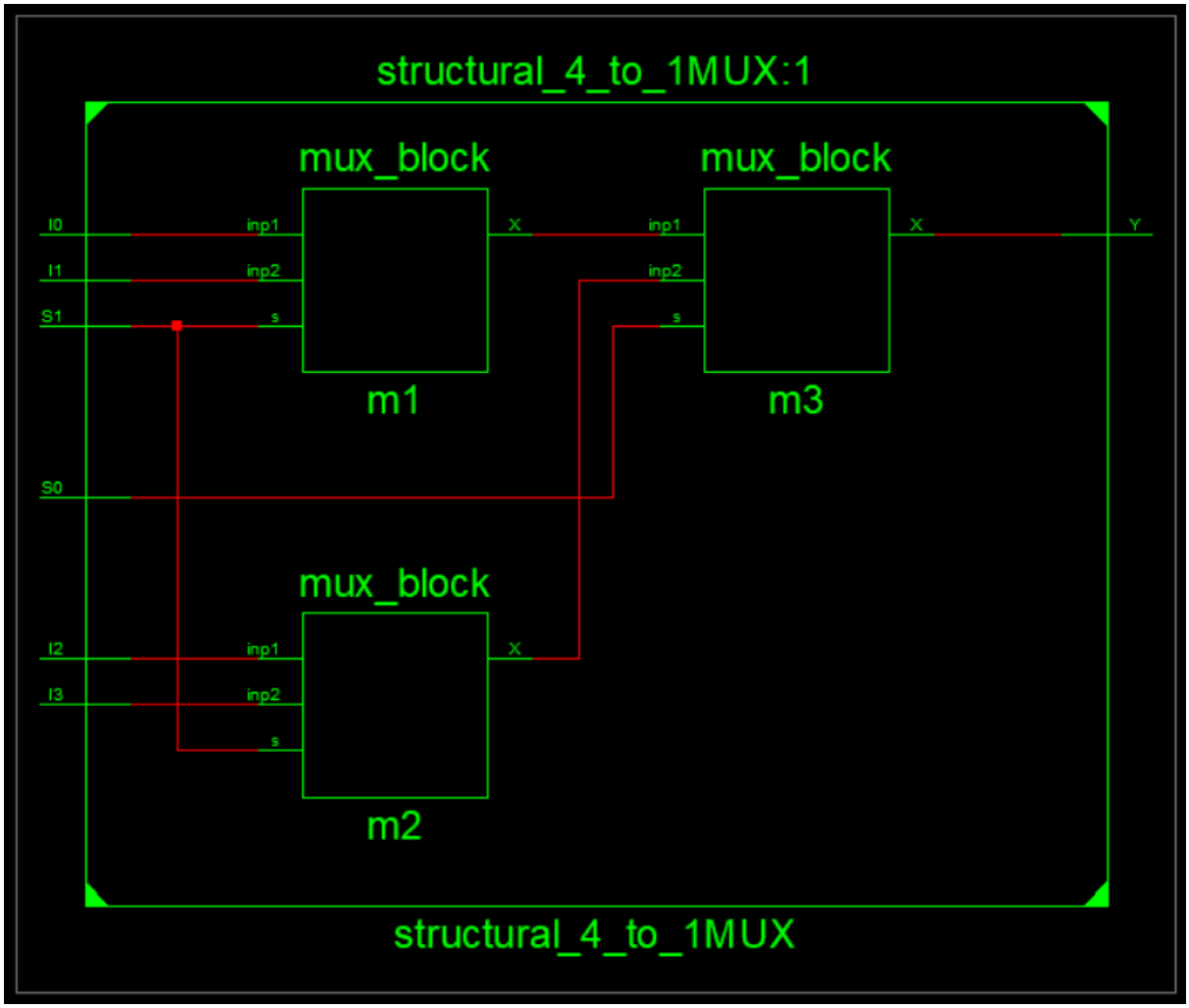
```
begin

m1: mux_block port map(I0,I1,S1,O1);
m2: mux_block port map(I2,I3,S1,O2);
m3: mux_block port map(O1,O2,S0,Y);

end STRUCTURAL;
```

OUTPUT: -

ENTITY DIAGRAM: -



DETAILED ENTITY DIAGRAM: -



RESULT:

Thus, through structural modelling we have implemented the 4:1 MUX using 2:1 MUX in VHDL and we have processed the overview of the basic structure of this entity as a whole using the RTL schematics. The blank square correctly represents the MUX's respective input, output Ports and their selection lines signals in Structural Modelling.

CONCLUSION:

In this experiment, we learnt about how to implement the 4:1 MUX using 2:1 MUX in VHDL as the structural ports design and placement of the respective MUX's input, output and selection lines is correctly represented using structural modelling.