

# Experiment 5

## Implementing 4X1 Multiplexer

### 4x1 Multiplexer

#### *Design code Structural :*

```
C:\Users\student\Desktop\DD\multi\multi.srcs\sources_1\new\multi.vhd

22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24
25 -- Uncomment the following library declaration if using
26 -- arithmetic functions with Signed or Unsigned values
27 --use IEEE.NUMERIC_STD.ALL;
28
29 -- Uncomment the following library declaration if instantiating
30 -- any Xilinx leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33
34 entity multi is
35 port(
36 a1:in std_logic;
37 a2:in std_logic;
38 a3:in std_logic;
39 a4:in std_logic;
40 s1:in std_logic;
41 s2:in std_logic;
42 y:out std_logic;
43 end multi;
44
45 architecture Behavioral of multi is
46
47 begin
48 process(a1,a2,a3,a4,s1,s2) is
49 begin
50 y<=(a1 and (not s1)and (not s2) ) or(a2 and(not s1)and (s2)) or (a3 and(s1) and (not s2)) or (a4 and(s1)and(s2));
51 end process;
52
53 end Behavioral;
54
```

#### *Design Code behavioural :*

```
Project Summary | Schematic | multi.vhd * | testbenchmulti.vhd | Schematic (2) | Schematic (3) | ? |
C:\Users\student\Desktop\DD\multi\multi.srcs\sources_1\new\multi.vhd

13
14 entity multi is
15 port(
16 a1:in std_logic;
17 a2:in std_logic;
18 a3:in std_logic;
19 a4:in std_logic;
20 s1:in std_logic;
21 s2:in std_logic;
22 y:out std_logic;
23 end multi;
24
25 architecture Behavioral of multi is
26
27 begin
28 process(a1,a2,a3,a4,s1,s2) is
29 begin
30 if s1 = '0' and s2 = '0' then
31 y<=a1;
32 elsif s1='0'and s2='1' then
33 y<=a2;
34 elsif s1='1'and s2='0' then
35 y<=a3;
36 else
37 y<=a4;
38 end if;
39 end process;
40
```

```

39 component multi is
40 port(
41   a1:in std_logic;
42   a2:in std_logic;
43   a3:in std_logic;
44   a4:in std_logic;
45   s1:in std_logic;
46   s2:in std_logic;
47   y:out std_logic);
48 end component;
49 signal a1_in,a2_in,a3_in,a4_in,y_out,s1_in,s2_in:std_logic;
50 begin
51   DUT: multi port map(a1_in,a2_in,a3_in,a4_in,s1_in,s2_in,y_out);
52 process
53 begin
54   a1_in<='1';
55   a2_in<='0';
56   a3_in<='0';
57   a4_in<='1';
58   s1_in<='0';
59   s2_in<='0';
60   wait for 50 ns;
61   a1_in<='1';
62   a2_in<='0';
63   a3_in<='0';
64   a4_in<='1';
65   s1_in<='0';
66   s2_in<='1';
67   wait for 50 ns;
68   wait:

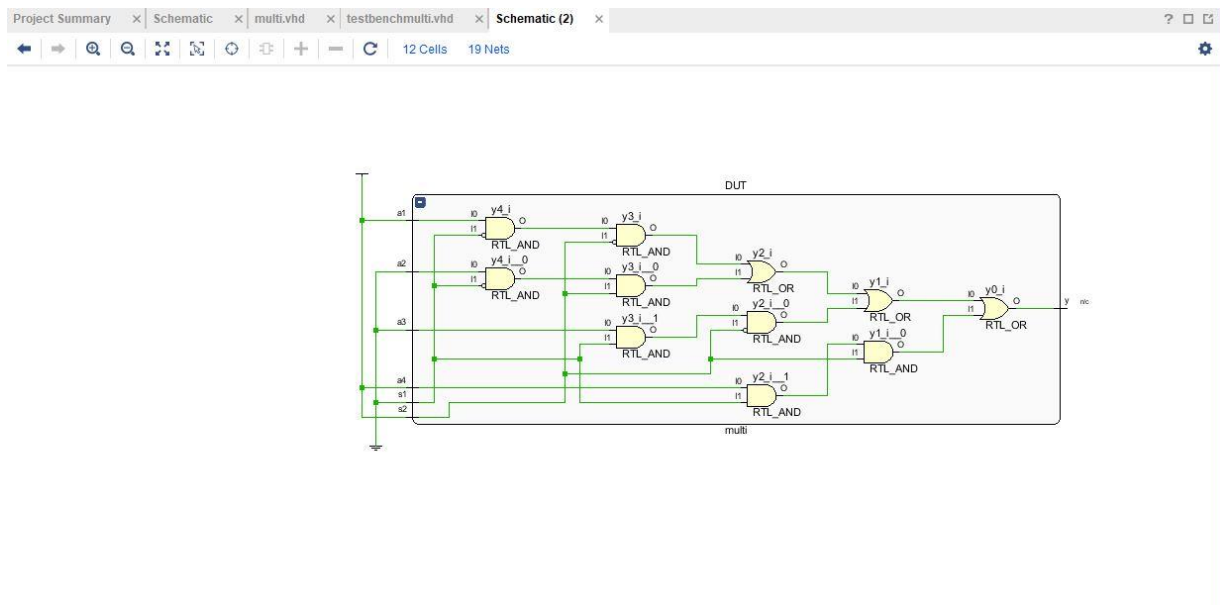
```

The screenshot shows the Vivado IDE interface. On the left, a table lists the current values of signals in the testbench:

Name	Value
a1_in	1
a2_in	0
a3_in	0
a4_in	1
y_out	0
s1_in	0
s2_in	1

On the right, a timing diagram is displayed for a duration of 1,000.000 ns. The diagram shows the digital waveforms for the signals listed in the table. The signals a1\_in, a2\_in, a3\_in, a4\_in, s1\_in, and s2\_in are all constant at their respective values throughout the simulation. The signal y\_out is constant at 0.

*Schematic:*



**Result:-** 4X1 Multiplexer has been implemented successfully and their test results are shown above.