**Multiplexer 4:1 using Structural model**

**1. code**

---------------------------------------------------------------------------------

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity multiplexer\_s is

Port ( a,b,c,d,s0,s1 : in STD\_LOGIC;

x : out STD\_LOGIC);

end multiplexer\_s;

architecture structural of multiplexer\_s is

COMPONENT AND\_GATE

PORT( a,b,c : IN STD\_LOGIC;

d : OUT STD\_LOGIC );

END COMPONENT;

COMPONENT NOT\_GATE

PORT( a : IN STD\_LOGIC;

b : OUT STD\_LOGIC );

END COMPONENT;

COMPONENT OR\_GATE

PORT( a,b,c,d : IN STD\_LOGIC;

e : OUT STD\_LOGIC );

END COMPONENT;

SIGNAL t1,t2,t3,t4 : STD\_LOGIC;

SIGNAL s0\_bar,s1\_bar : STD\_LOGIC;

BEGIN

U1 : NOT\_GATE PORT MAP (s0, s0\_bar);

U2 : NOT\_GATE PORT MAP (s1, s1\_bar);

U3 : AND\_GATE PORT MAP (a, s0\_bar, s1\_bar, t1);

U4 : AND\_GATE PORT MAP (b, s0, s1\_bar, t2);

U5 : AND\_GATE PORT MAP (c, s0\_bar, s1, t3);

U6 : AND\_GATE PORT MAP (d, s0, s1, t4);

U7 : OR\_GATE PORT MAP (t1, t2, t3, t4, x);

end Structural;

**Program Added as a source for AND\_GATE**

----------------------------------------------------------------------------------

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity AND\_GATE is

Port ( a,b,c : in STD\_LOGIC;

d : out STD\_LOGIC);

end AND\_GATE;

architecture Behavioral of AND\_GATE is

begin

d <= a AND b AND c;

end Behavioral;

**Program Added as a source for OR\_GATE**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity OR\_GATE is

Port ( a,b,c,d : in STD\_LOGIC;

e : out STD\_LOGIC);

end OR\_GATE;

architecture Behavioral of OR\_GATE is

begin

e <= a OR b OR c OR d;

end Behavioral;

**Program Added as a source for NOT\_GATE**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity NOT\_GATE is

Port ( a : in STD\_LOGIC;

b : out STD\_LOGIC);

end NOT\_GATE;

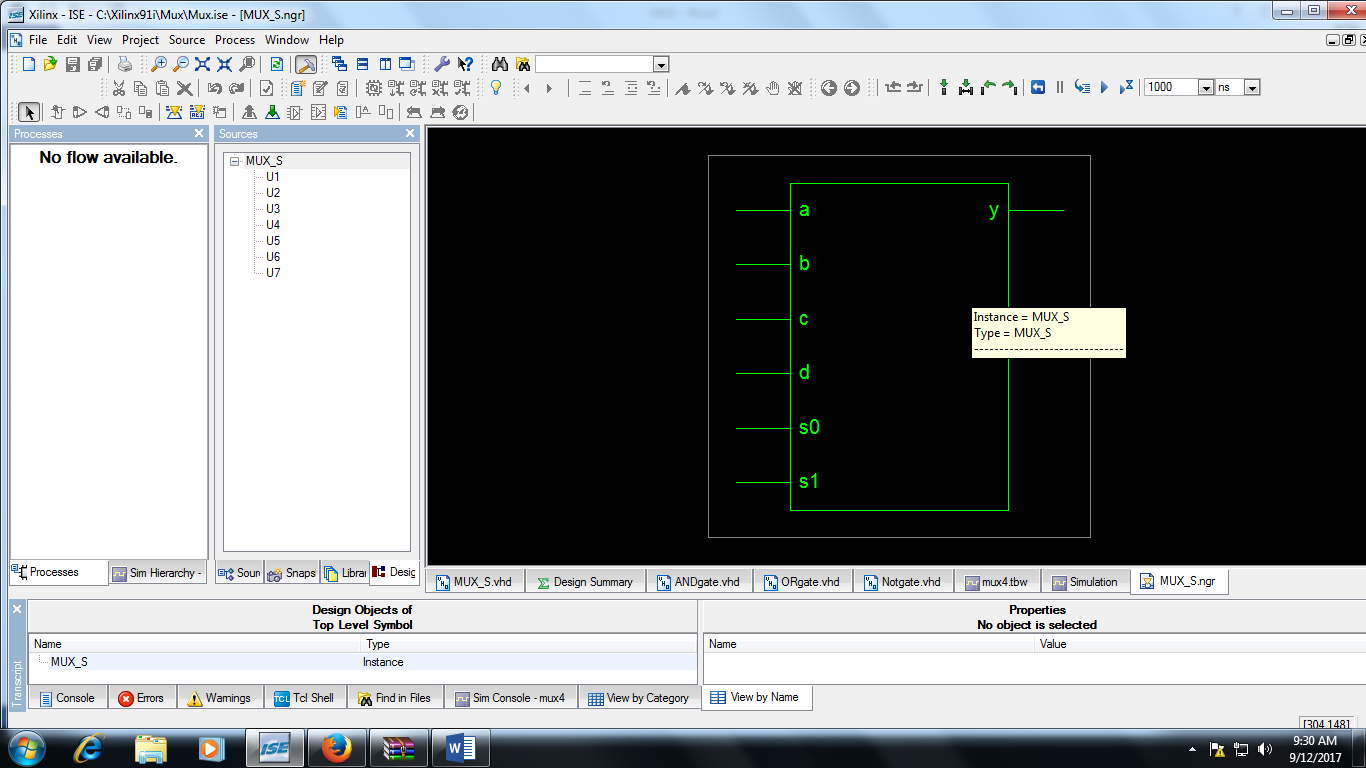
architecture Behavioral of NOT\_GATE is

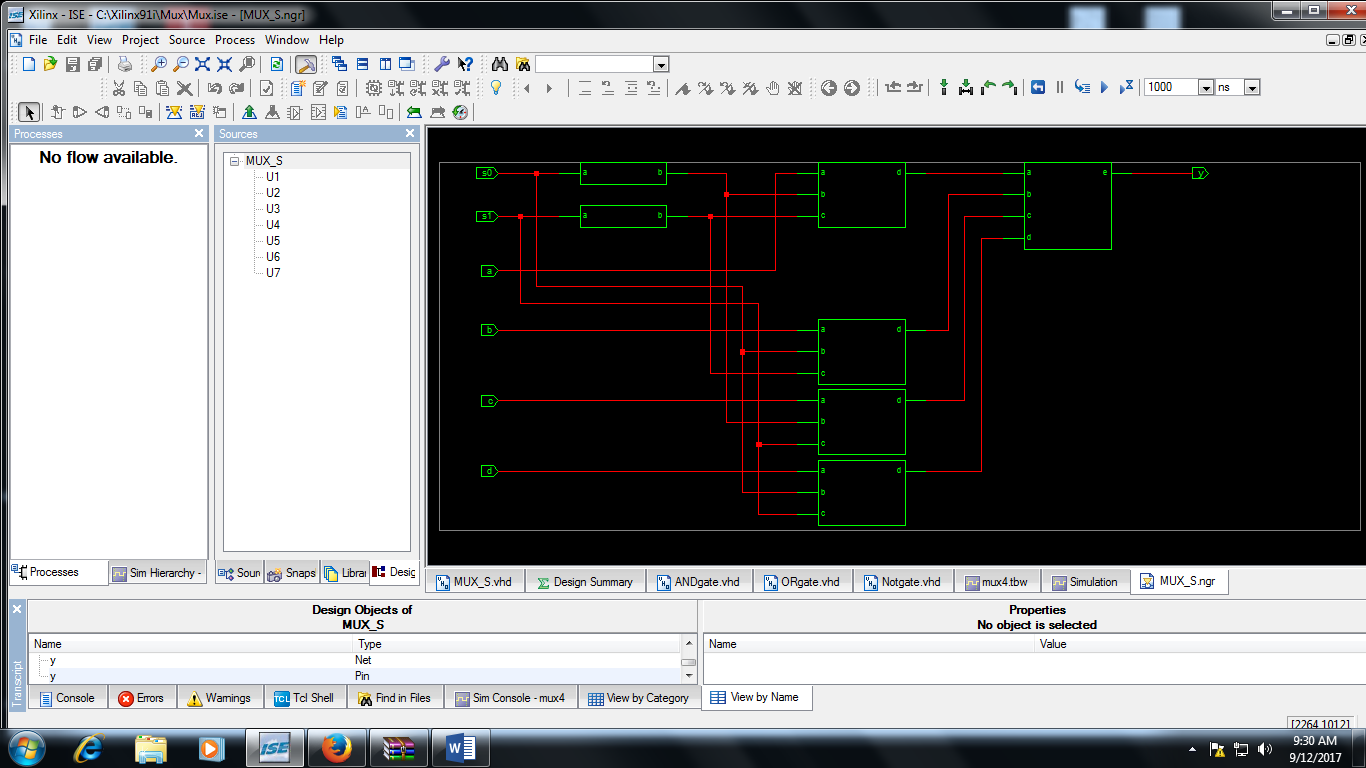
begin

b <= not a ;

end Behavioral;

**2. RTL Schematic**





**3. Simulation Output**

