**VHDL code to implement 16x1 Multiplexer using Structural modeling**

**T Raja Aadhithan**

**602162021**

Design Code :

--16 x 1 multiplexer

--or gate

library ieee;

use ieee.std\_logic\_1164.all;

entity or\_gate is port (a,b:in std\_logic; y:out std\_logic);end or\_gate;

architecture gate of or\_gate is begin

    y <= a or b;

end gate;

-- and gate

library ieee;

use ieee.std\_logic\_1164.all;

entity and\_gate is port (a,b:in std\_logic; y:out std\_logic);end and\_gate;

architecture gate of and\_gate is begin

    y <= a and b;

end gate;

-- not gate

library ieee;

use ieee.std\_logic\_1164.all;

entity not\_gate is port (a:in std\_logic; y:out std\_logic);end not\_gate;

architecture gate of not\_gate is begin

    y <= not a;

end gate;

-- 2x1 mux

library ieee;

use ieee.std\_logic\_1164.all;

entity mux2 is port (a,b,s:in std\_logic; y:out std\_logic);end mux2;

architecture mulp of mux2 is

    component or\_gate port (a,b:in std\_logic; y:out std\_logic);end component;

    component and\_gate port (a,b:in std\_logic; y:out std\_logic);end component;

    component not\_gate port (a:in std\_logic; y:out std\_logic);end component;

    signal sb,x,w:std\_logic;

    begin

        n1 : not\_gate port map(s,sb);

        a1 : and\_gate port map(a,sb,x);

        a2 : and\_gate port map(b,s,w);

        o1 : or\_gate port map(x,w,y);

end mulp;

--16x1 mux

library ieee;

use ieee.std\_logic\_1164.all;

entity mux is

    port ( i : in std\_logic\_vector(0 to 15);

             s : in std\_logic\_vector(0 to 3);

             y : out std\_logic);

end mux;

architecture mux16 of mux is

    component mux2 port (a,b,s:in std\_logic; y:out std\_logic);end component;

    signal w:std\_logic\_vector(0 to 13);

    begin

        m1 : mux2 port map(i(0),i(1),s(3),w(0));

        m2 : mux2 port map(i(2),i(3),s(3),w(1));

        m3 : mux2 port map(i(4),i(5),s(3),w(2));

        m4 : mux2 port map(i(6),i(7),s(3),w(3));

        m5 : mux2 port map(i(8),i(9),s(3),w(4));

        m6 : mux2 port map(i(10),i(11),s(3),w(5));

        m7 : mux2 port map(i(12),i(13),s(3),w(6));

        m8 : mux2 port map(i(14),i(15),s(3),w(7));

        m9 : mux2 port map(w(0),w(1),s(2),w(8));

        m10 : mux2 port map(w(2),w(3),s(2),w(9));

        m11 : mux2 port map(w(4),w(5),s(2),w(10));

        m12 : mux2 port map(w(6),w(7),s(2),w(11));

        m13 : mux2 port map(w(8),w(9),s(1),w(12));

        m14 : mux2 port map(w(10),w(11),s(1),w(13));

        m15 : mux2 port map(w(12),w(13),s(0),y);

end mux16;

Test benc

library ieee;

use ieee.std\_logic\_1164.all;

entity mux\_tb is end mux\_tb;

architecture behavior of mux\_tb is

    component mux

    port( i : IN  std\_logic\_vector(0 to 15);

 s : IN  std\_logic\_vector(0 to 3);

         y : OUT  std\_logic);

   end component;

   signal a : std\_logic\_vector(0 to 15);

   signal s : std\_logic\_vector(0 to 3);

   signal y : std\_logic;

    begin

   uut: mux port map ( i => a, s => s, y => y );

   stim\_proc: process

   begin

      a <= "0000000000000000";

        s <= "0000";

        a(0) <= '0';

       wait for 10 ns;

        a(0) <= '1';

        wait for 10 ns;

        s <= "0001";

        a(1) <= '0';

       wait for 10 ns;

        a(1) <= '1';

        wait for 10 ns;

        s <= "0010";

        a(2) <= '0';

       wait for 10 ns;

        a(2) <= '1';

        wait for 10 ns;

        s <= "0011";

        a(3) <= '0';

       wait for 10 ns;

        a(3) <= '1';

        wait for 10 ns;

        s <= "0100";

        a(4) <= '0';

       wait for 10 ns;

        a(4) <= '1';

        wait for 10 ns;

        s <= "0101";

        a(5) <= '0';

       wait for 10 ns;

        a(5) <= '1';

        wait for 10 ns;

s <= "0110";

        a(6) <= '0';

       wait for 10 ns;

        a(6) <= '1';

        wait for 10 ns;

s <= "0111";

        a(7) <= '0';

       wait for 10 ns;

        a(7) <= '1';

        wait for 10 ns;

        s <= "1000";

        a(8) <= '0';

       wait for 10 ns;

        a(8) <= '1';

        wait for 10 ns;

s <= "1001";

        a(9) <= '0';

       wait for 10 ns;

        a(9) <= '1';

        wait for 10 ns;

s <= "1010";

        a(10) <= '0';

       wait for 10 ns;

        a(10) <= '1';

        wait for 10 ns;

        s <= "1011";

        a(11) <= '0';

       wait for 10 ns;

        a(11) <= '1';

        wait for 10 ns;

        s <= "1100";

        a(12) <= '0';

       wait for 10 ns;

        a(12) <= '1';

        wait for 10 ns;

        s <= "1101";

        a(13) <= '0';

       wait for 10 ns;

        a(13) <= '1';

        wait for 10 ns;

        s <= "1110";

        a(14) <= '0';

       wait for 10 ns;

        a(14) <= '1';

        wait for 10 ns;

        s <= "1111";

        a(15) <= '0';

       wait for 10 ns;

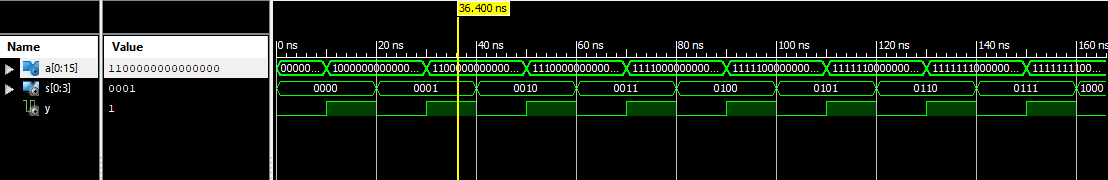
        a(15) <= '1';

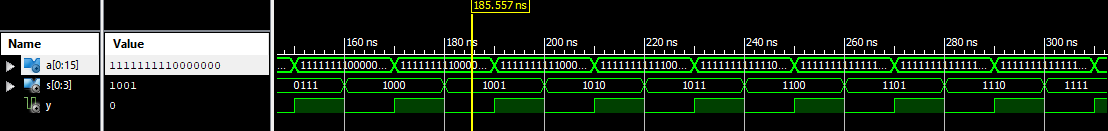
        wait for 10 ns;

   end process;

end;

Simulation result:





RTL Diagram:

