**PRACTICAL 05**

**B-81 Tanmay Pawan Bisen**

***Q) Design and implement the working structure of multiplexor and demultiplexor. Assume four cables are given input to multiplexor. The data of 8 bits should be outputted at the interval of 100ns in round robin fashion. The data will be given as input to demultiplexor and it will be redistributed to same number of channel in its output.***

**MULTIPLEXOR CODE:**

library ieee;

use ieee.std\_logic\_1164.all;

entity mux1 is

port(d0,d1,d2,d3: in std\_logic\_vector(7 downto 0);

s0: in std\_logic\_vector(1 downto 0);

x: out std\_logic\_vector(7 downto 0));

end mux1;

architecture bev of mux1 is

begin

process(d0,d1,d2,d3,s0)

begin

if(s0="00") then

x<=d0;

elsif(s0="01") then

x<=d1;

elsif(s0="10") then

x<=d2;

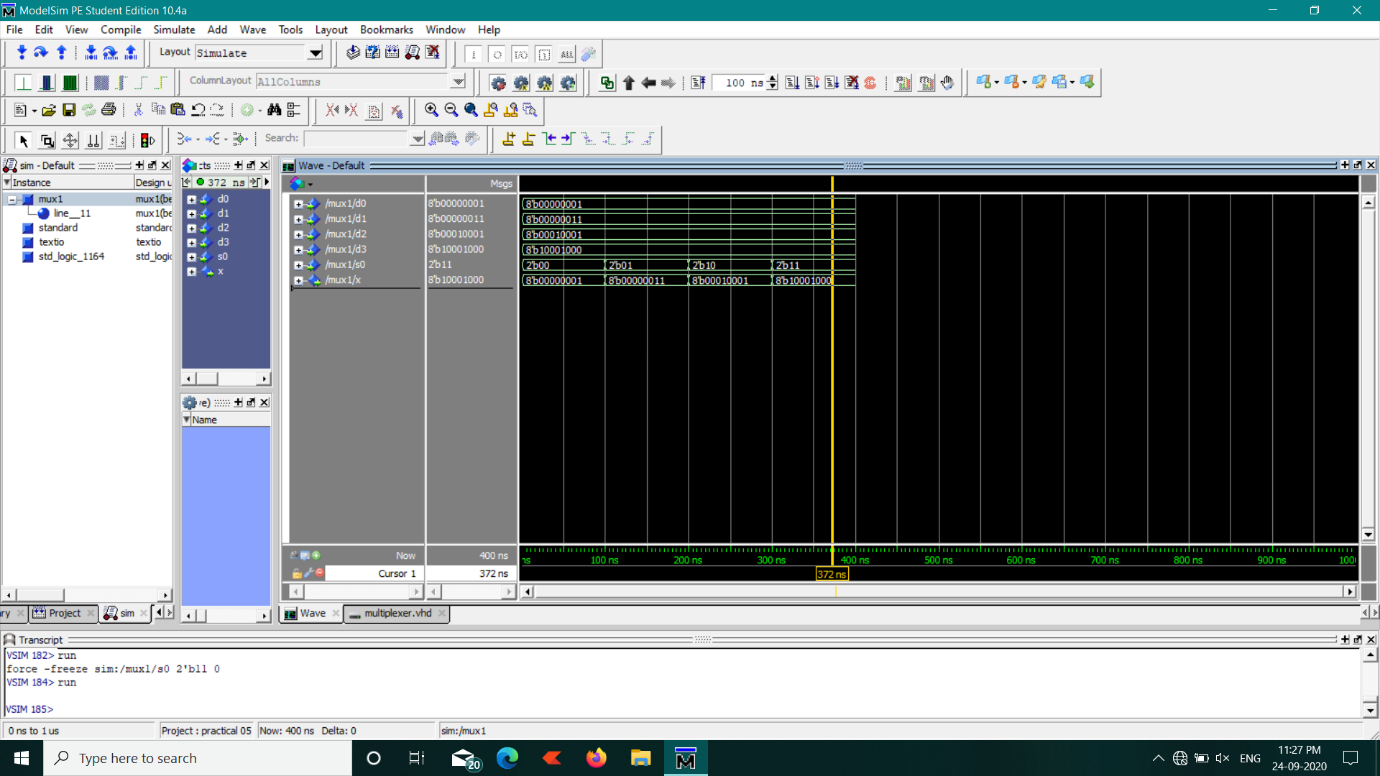
elsif(s0="11") then

x<=d3;

end if;

end process;

end bev;



**DEMULTIPLEXOR:**

library ieee;

use ieee.std\_logic\_1164.all;

entity demux1 is

port(y: in std\_logic\_vector(7 downto 0);

s1: in std\_logic\_vector(1 downto 0);

d00,d11,d22,d33: out std\_logic\_vector(7 downto 0));

end demux1;

architecture bev of demux1 is

begin

process(y,s1)

begin

if(s1="00") then

d00<=y;

d11<="UUUUUUUU";

d22<="UUUUUUUU";

d33<="UUUUUUUU";

elsif(s1="01") then

d00<="UUUUUUUU";

d11<=y;

d22<="UUUUUUUU";

d33<="UUUUUUUU";

elsif(s1="10") then

d00<="UUUUUUUU";

d11<="UUUUUUUU";

d22<=y;

d33<="UUUUUUUU";

elsif(s1="11") then

d00<="UUUUUUUU";

d11<="UUUUUUUU";

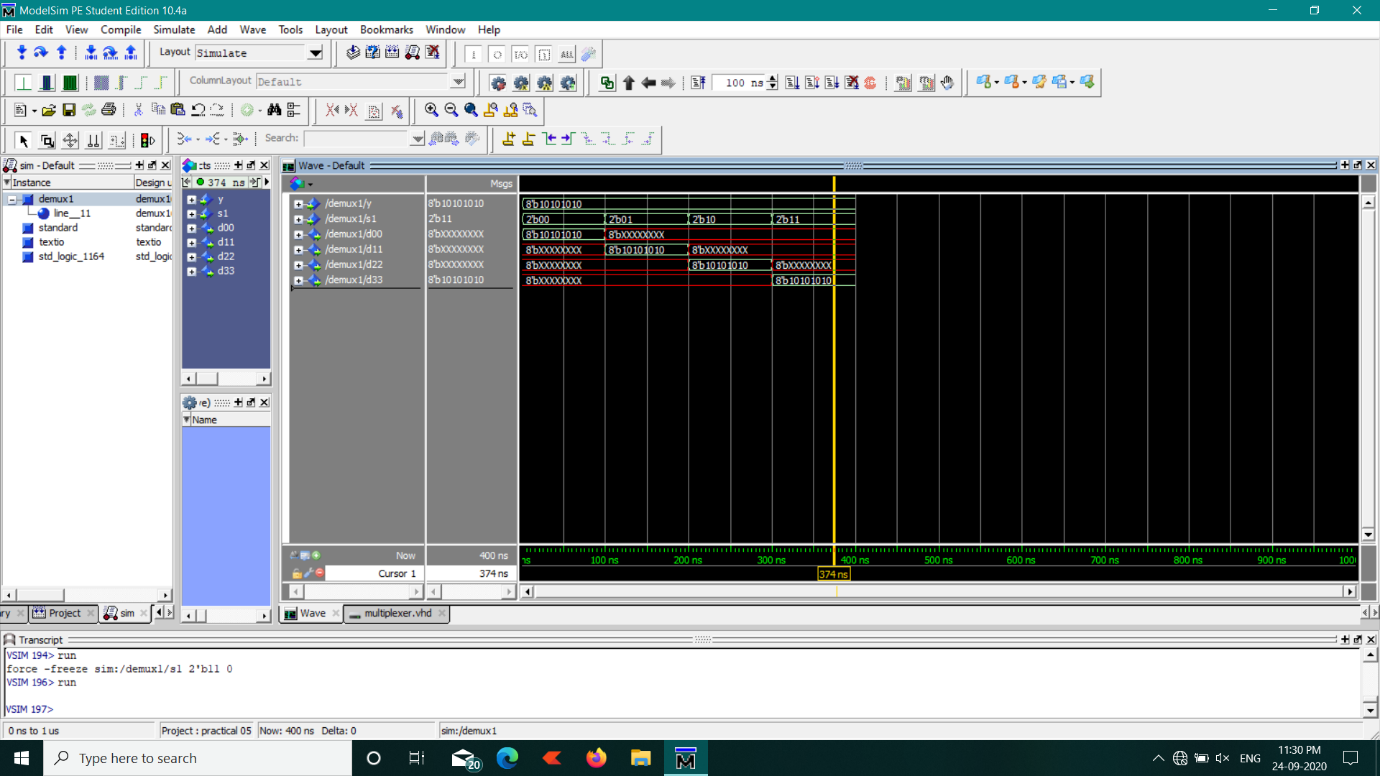
d22<="UUUUUUUU";

d33<=y;

end if;

end process;

end bev;



**MUX-DEMUX:**

library ieee;

use ieee.std\_logic\_1164.all;

entity muxdemux1 is

end muxdemux1;

architecture structural of muxdemux1 is

component mux1

port(d0,d1,d2,d3: in std\_logic\_vector(7 downto 0);

s0: in std\_logic\_vector(1 downto 0);

x: out std\_logic\_vector(7 downto 0));

end component;

component demux1

port(y: in std\_logic\_vector(7 downto 0);

s1: in std\_logic\_vector(1 downto 0);

d00,d11,d22,d33: out std\_logic\_vector(7 downto 0));

end component;

signal i0,i1,i2,i3,r4,j0,j1,j2,j3,s4 : std\_logic\_vector(7 downto 0);

signal k0,k1 : std\_logic\_vector(1 downto 0);

begin

g0: mux1 port map(i0,i1,i2,i3,k0,r4);

g1: demux1 port map(s4,k1,j0,j1,j2,j3);

process

begin

wait for 100 ns;

k0<="00";

s4<=r4;

wait for 100 ns;

k0<="01";

k1<="00";

s4<=r4;

wait for 100 ns;

k0<="10";

k1<="01";

s4<=r4;

wait for 100 ns;

k0<="11";

k1<="10";

s4<=r4;

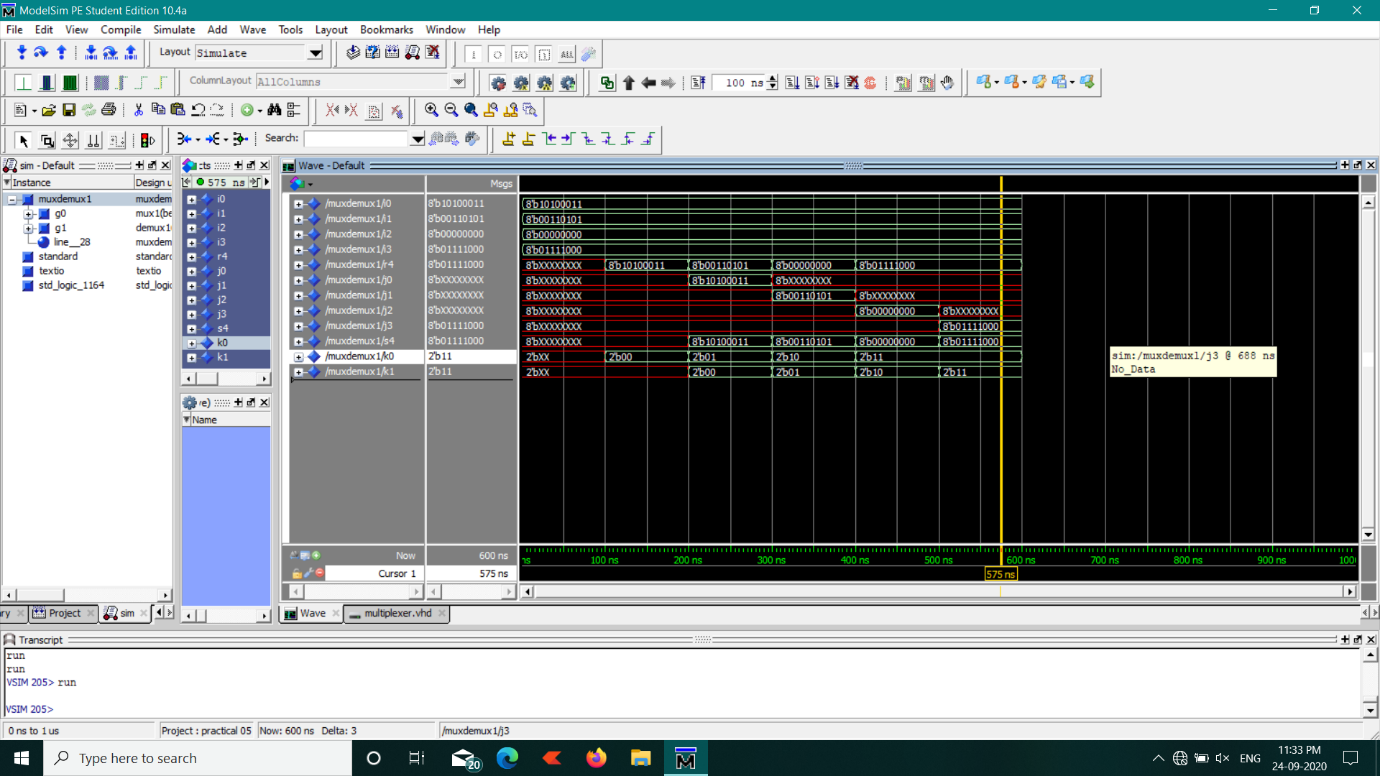
wait for 100 ns;

k1<="11";

s4<=r4;

end process;

end structural;



***--------------------------------------------------------------XXXXXXXX---------------------------------------------------------***