**PRACTICAL 06**

**NAME**: T**anmay Pawan Bisen**, ROLL NO.: **B-81**

**1) AND behavioral code:**

library ieee;

use ieee.std\_logic\_1164.all;

entity and1 is

port(a4,b4:in std\_logic;

c4:out std\_logic);

end and1;

architecture bev of and1 is

begin

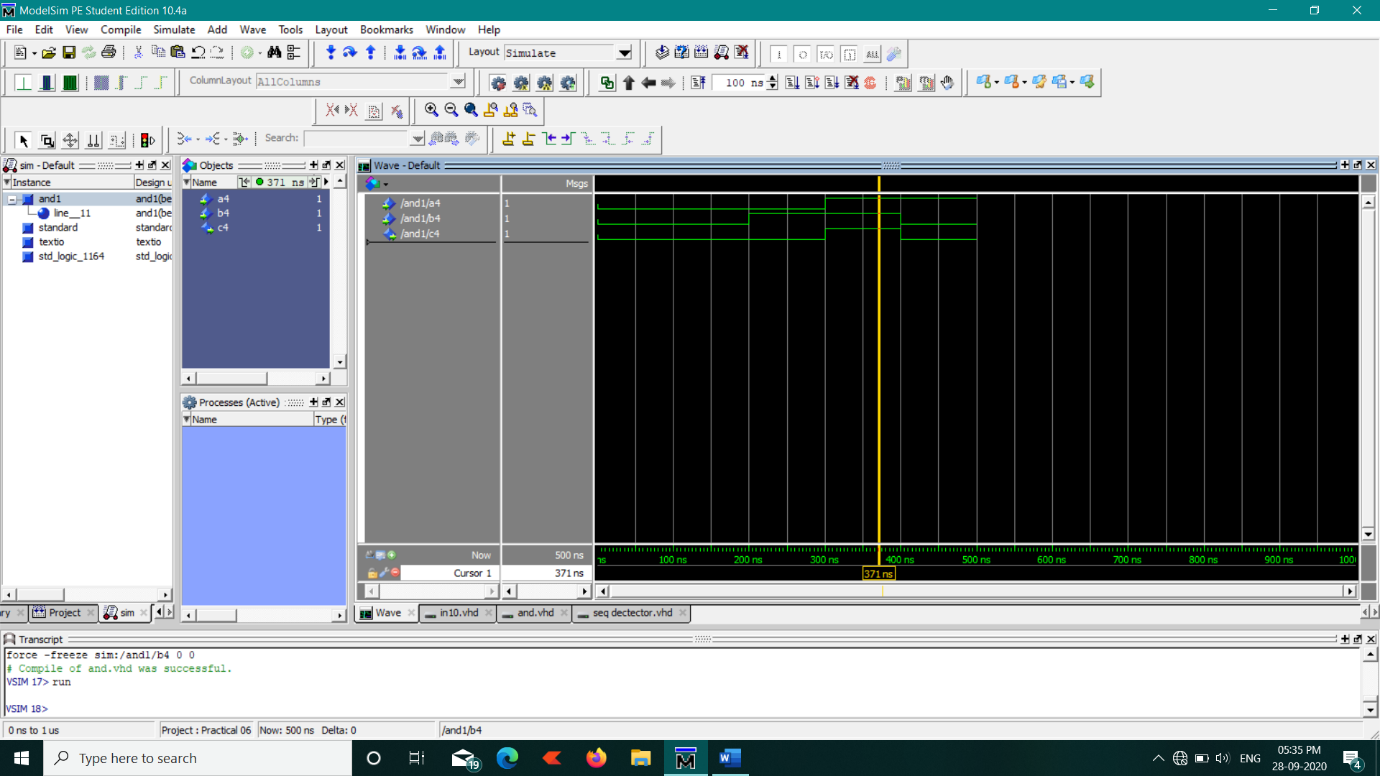
process(a4,b4)

begin

c4<=(a4 and b4);

end process;

end bev;



**2) OR behavioral code:**

library ieee;

use ieee.std\_logic\_1164.all;

entity or1 is

port(a2,b2:in std\_logic;

c2:out std\_logic);

end or1;

architecture bev of or1 is

begin

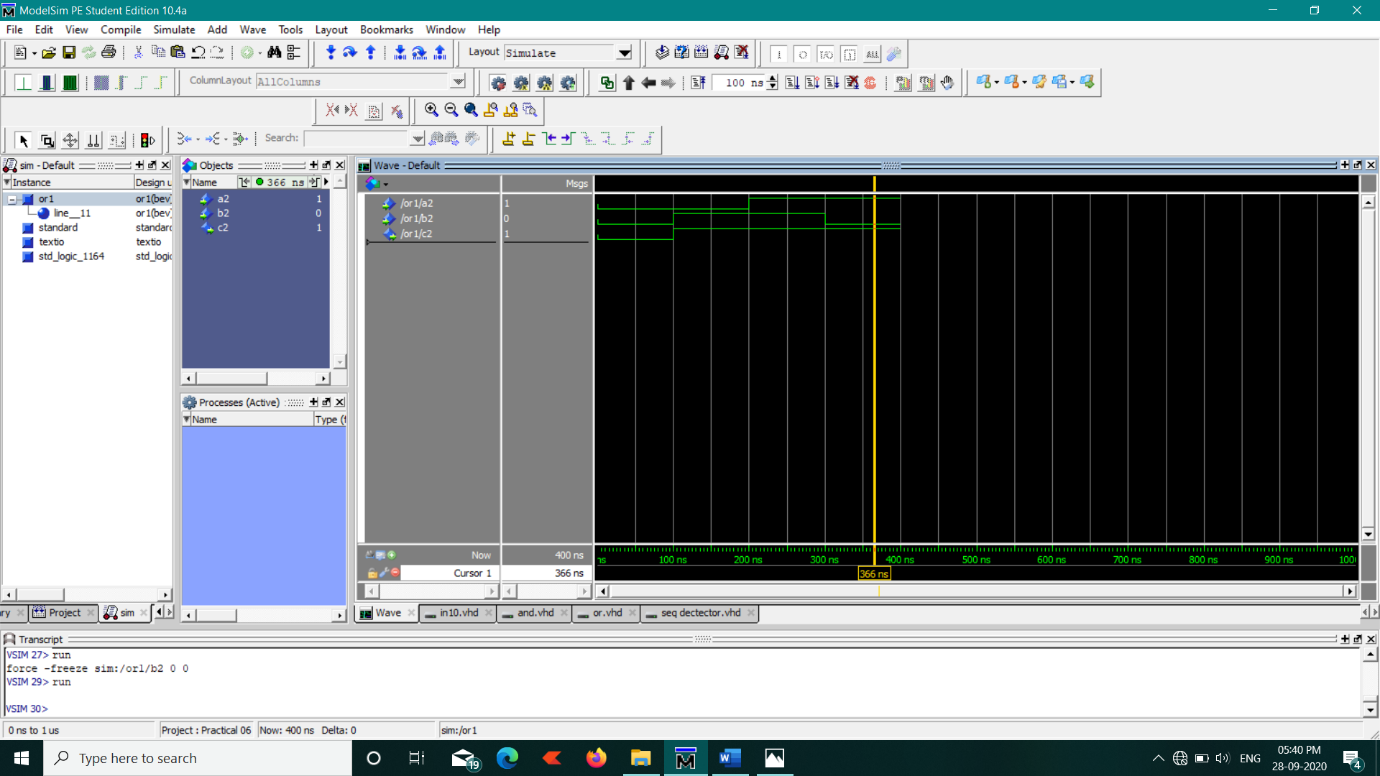
process(a2,b2)

begin

c2<=(a2 or b2);

end process;

end bev;



**3) NOT BEHAVIOURAL CODE:**

library ieee;

use ieee.std\_logic\_1164.all;

entity not1 is

port(a3:in std\_logic;

c3:out std\_logic);

end not1;

architecture bev of not1 is

begin

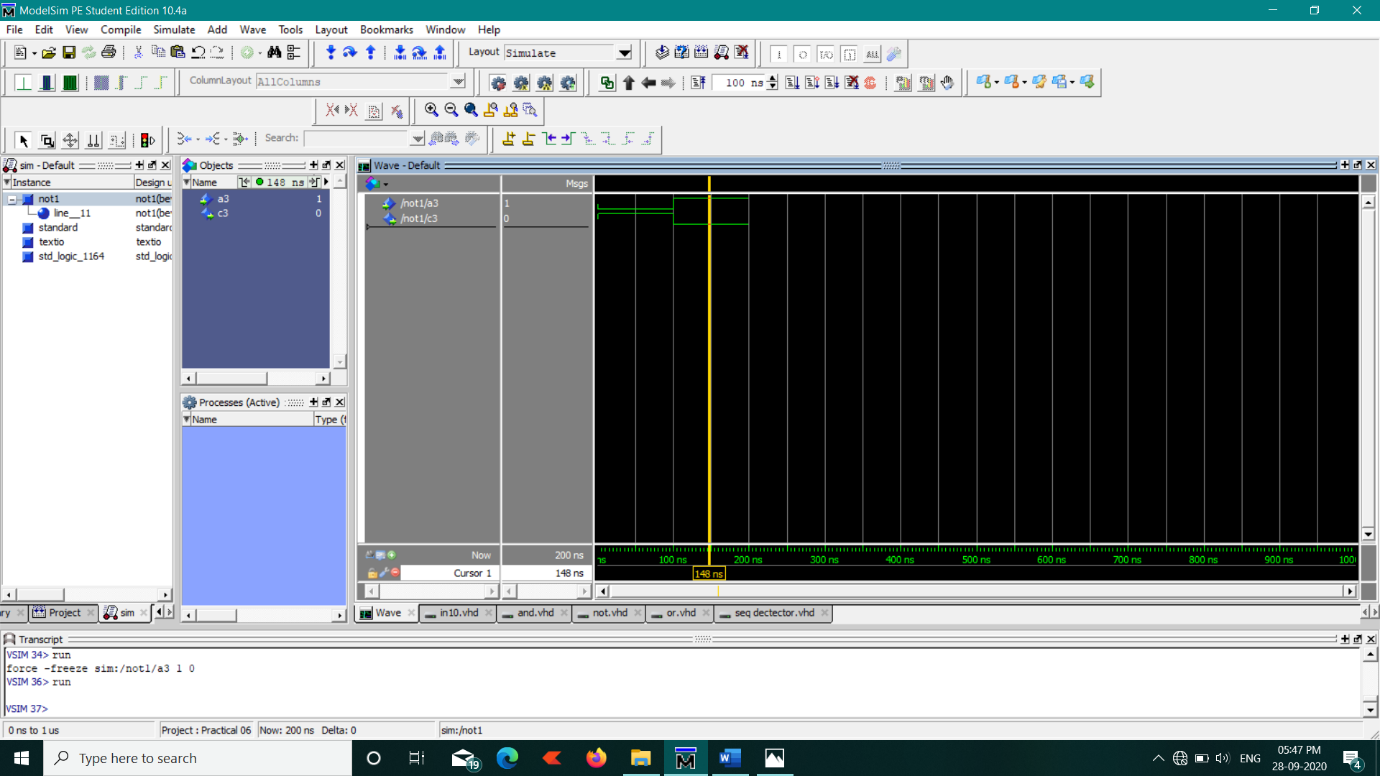
process(a3)

begin

c3<=(not a3);

end process;

end bev;



**4) JK FLIP FLOP CODE:**

library ieee;

use ieee.std\_logic\_1164.all;

entity JKFF is

port(

clk,j,k,ps:in std\_logic;

ns:out std\_logic

);

end JKFF;

architecture bev of JKFF is

begin

process(clk,j,k,ps)

begin

if(clk='1' and j='0' and k='0') then

ns<=ps;

elsif(clk='1' and j='0' and k='1') then

ns<='0';

elsif(clk='1' and j='1' and k='0') then

ns<='1';

elsif(clk='1' and j='1' and k='1') then

ns<=not ps;

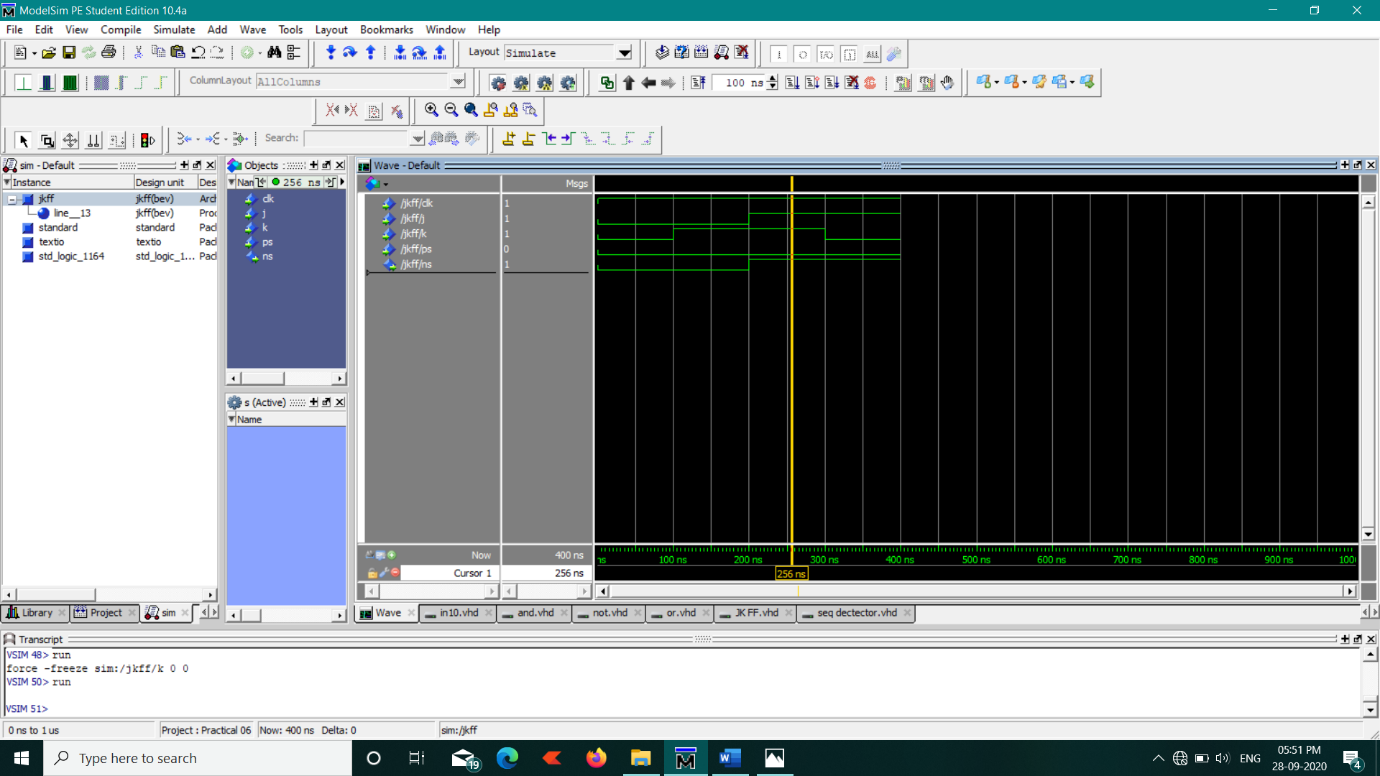
elsif(clk='0') then

ns<=ps;

end if;

end process;

end bev;



**5) SEQUENCE DETECTOR CODE:**

library ieee;

use ieee.std\_logic\_1164.all;

entity seqdetect is

port(clk1,x,psa,psb:in std\_logic;

nsa,nsb,z:out std\_logic);

end seqdetect;

architecture bev of seqdetect is

component JKFF

port(

clk,j,k,ps:in std\_logic;

ns:out std\_logic

);

end component;

component and1

port(a4,b4:in std\_logic;

c4:out std\_logic);

end component;

component or1

port(a2,b2:in std\_logic;

c2:out std\_logic);

end component;

component not1

port(a3:in std\_logic;

c3:out std\_logic);

end component;

signal fout,notofx,j1andout,or1out,or2out:std\_logic;

begin

g0:not1 port map(x,notofx);

g1:and1 port map(x,psb,j1andout);

g2:or1 port map(psb,notofx,or1out);

g3:JKFF port map(clk1,j1andout,or1out,psa,nsa);

g4:or1 port map(notofx,psa,or2out);

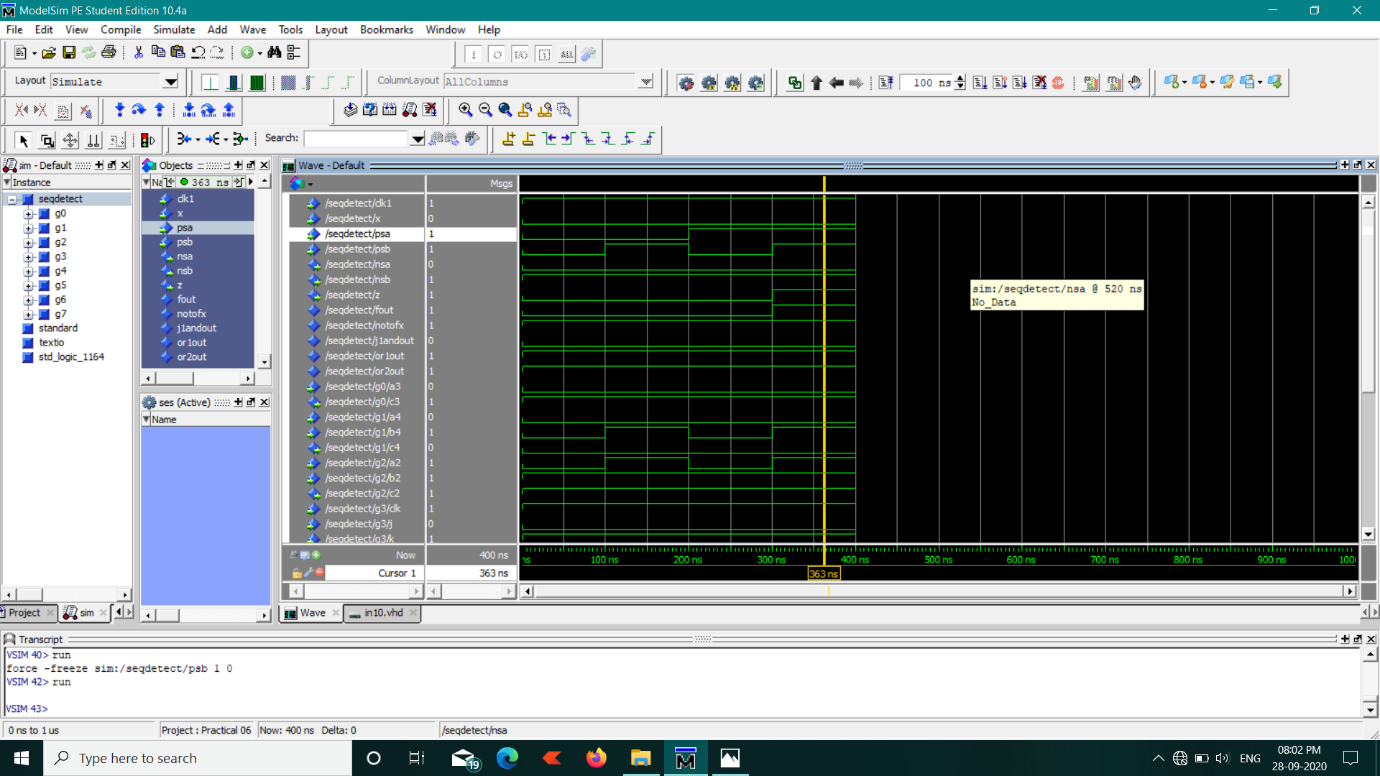
g5:JKFF port map(clk1,or2out,x,psb,nsb);

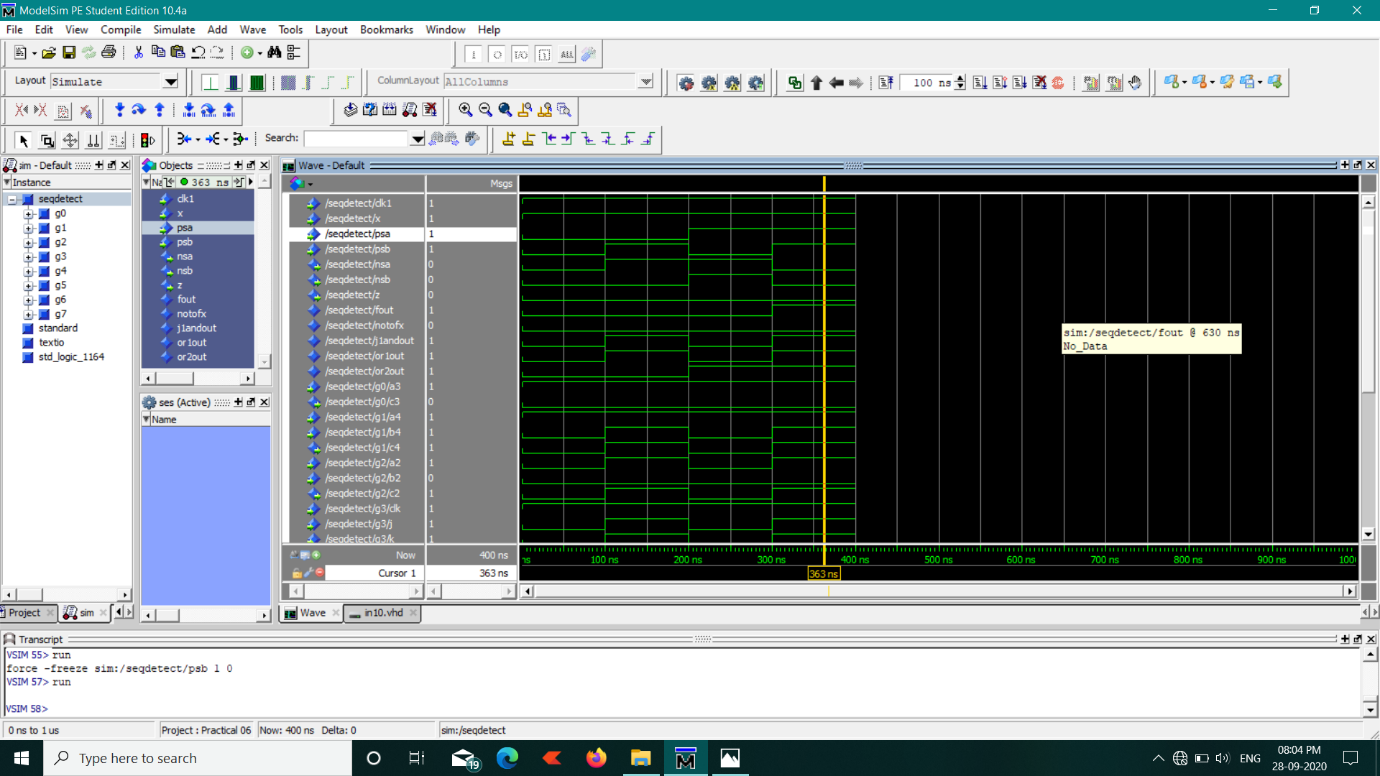
g6:and1 port map(psa,psb,fout);

g7:and1 port map(fout,notofx,z);

end bev;

**OUTPUT:**





**6) CODE FOR INPUT 10 BIT STREAM GIVEN BY USER:**

library ieee;

use ieee.std\_logic\_1164.all;

entity in10 is

port( clk2:in std\_logic;

h:in std\_logic\_vector(0 to 9);

j:out std\_logic\_vector(0 to 9));

end in10;

architecture bev of in10 is

component seqdetect

port(clk1,x,psa,psb:in std\_logic;

nsa,nsb,z:out std\_logic);

end component;

signal st,w0,w1,w2,w3,w4,w5,w6,w7,w8,w9,w10,w11,w12,w13,w14,w15,w16,w17,w18,w19:std\_logic;

begin

st<='0';

s0:seqdetect port map(clk2,h(0),st,st,w0,w1,j(0));

s1:seqdetect port map(clk2,h(1),w0,w1,w2,w3,j(1));

s2:seqdetect port map(clk2,h(2),w2,w3,w4,w5,j(2));

s3:seqdetect port map(clk2,h(3),w4,w5,w6,w7,j(3));

s4:seqdetect port map(clk2,h(4),w6,w7,w8,w9,j(4));

s5:seqdetect port map(clk2,h(5),w8,w9,w10,w11,j(5));

s6:seqdetect port map(clk2,h(6),w10,w11,w12,w13,j(6));

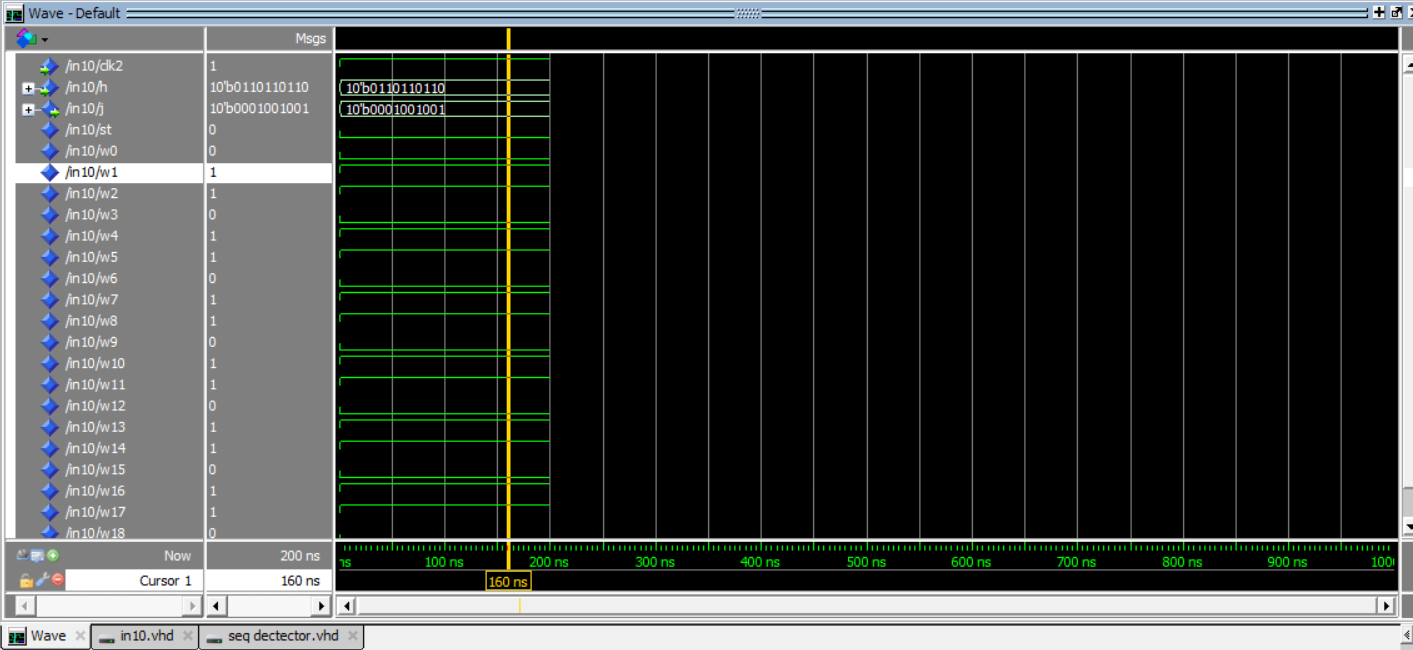
s7:seqdetect port map(clk2,h(7),w12,w13,w14,w15,j(7));

s8:seqdetect port map(clk2,h(8),w14,w15,w16,w17,j(8));

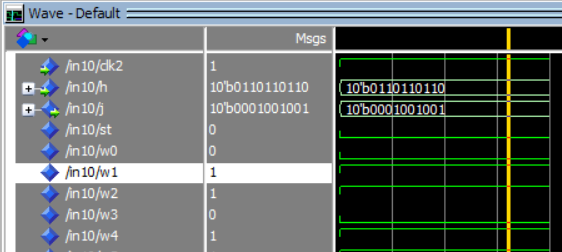
s9:seqdetect port map(clk2,h(9),w16,w17,w18,w19,j(9));

end bev;

**FINAL OUTPUT:**



**Magnified view:**



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_XXXXXX\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**