**LAB - 1**

Q1)

1. Verilog Code:

module q1a(a,b,c,f1);

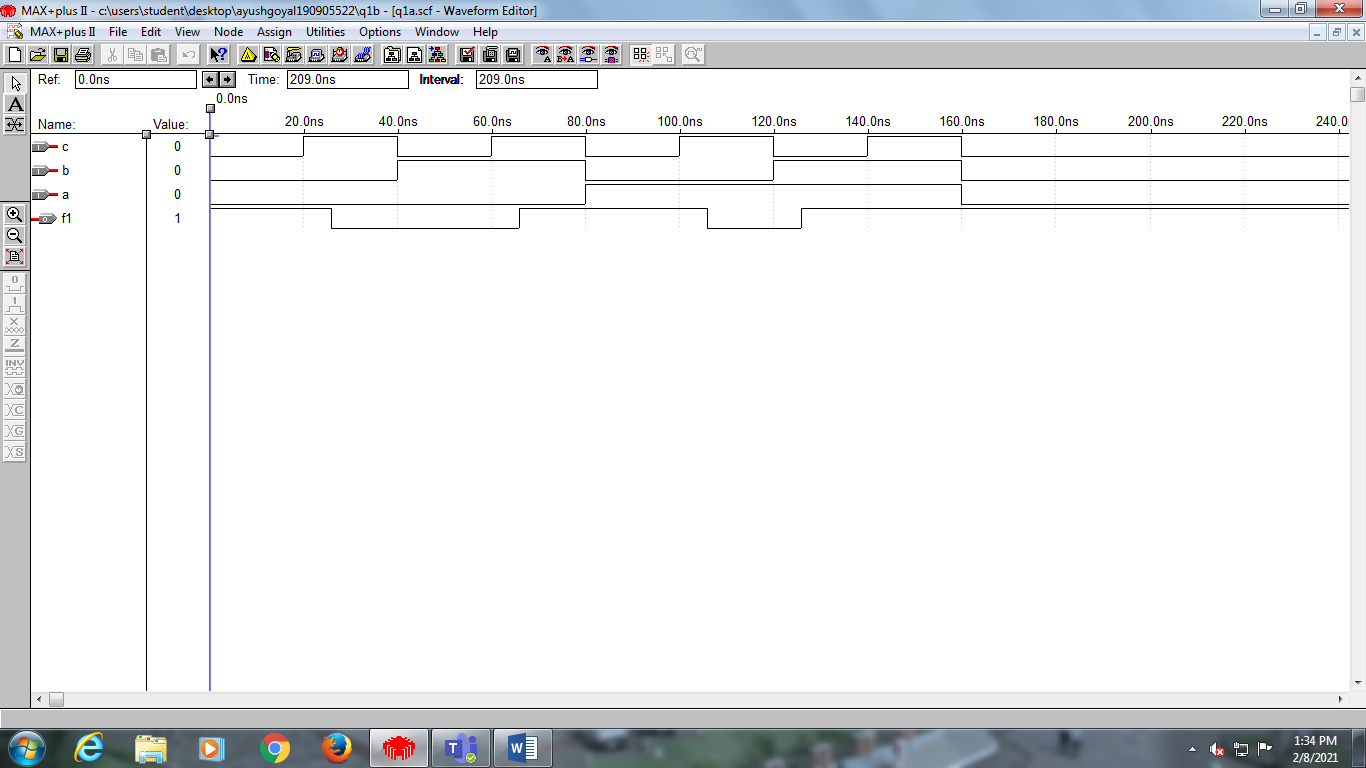
input a,b,c;

output f1;

assign f1 = a&(~c)|(b&c)|(~b)&(~c);

endmodule

Output:



1. Verilog Code:

module q1b(a,b,c,f2);

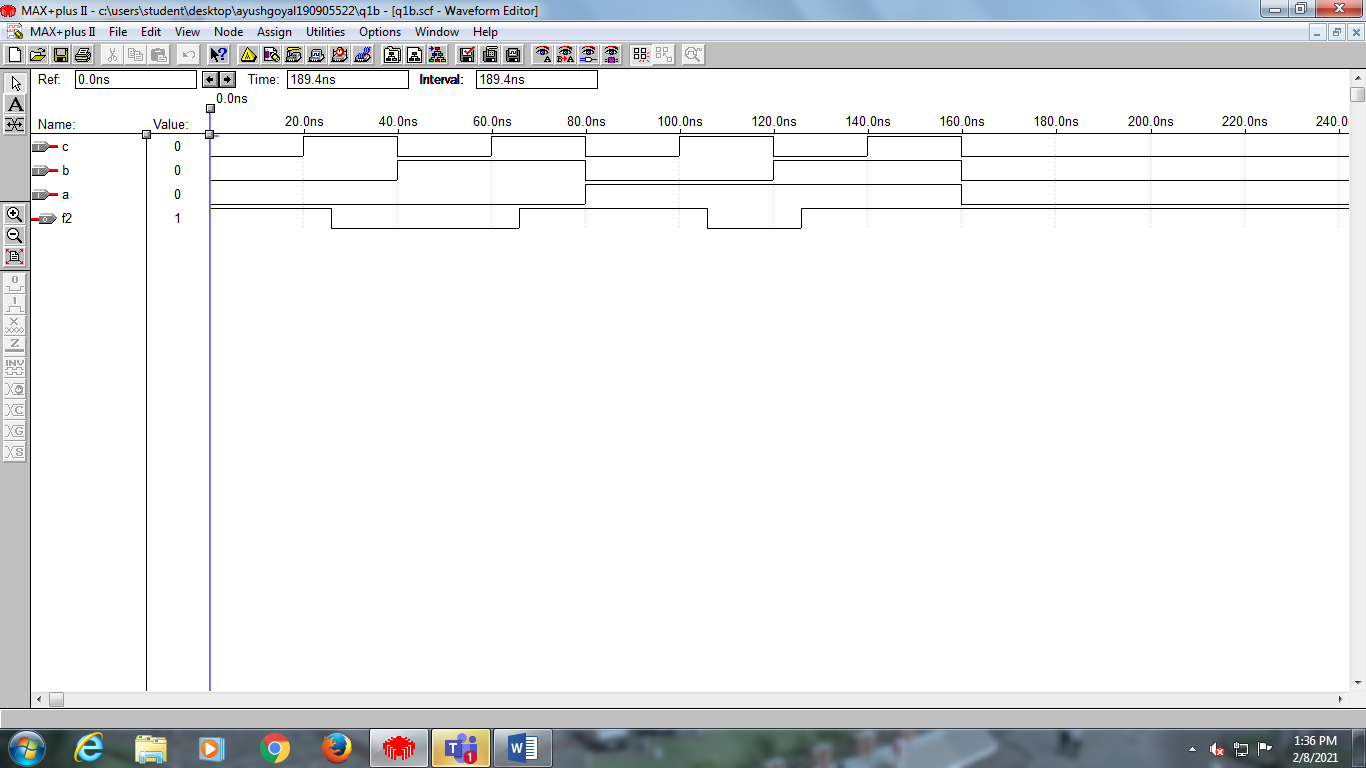
input a,b,c;

output f2;

assign f2 = (a|(~b)|c)&(a|b|(~c))&((~a)|b|(~c));

endmodule

Output:



Therefore we can see that f1 and f2 in question 1 are functionally equivalent.

2)

(A) Verilog Code:

module q2a(a,b,c,d,f);

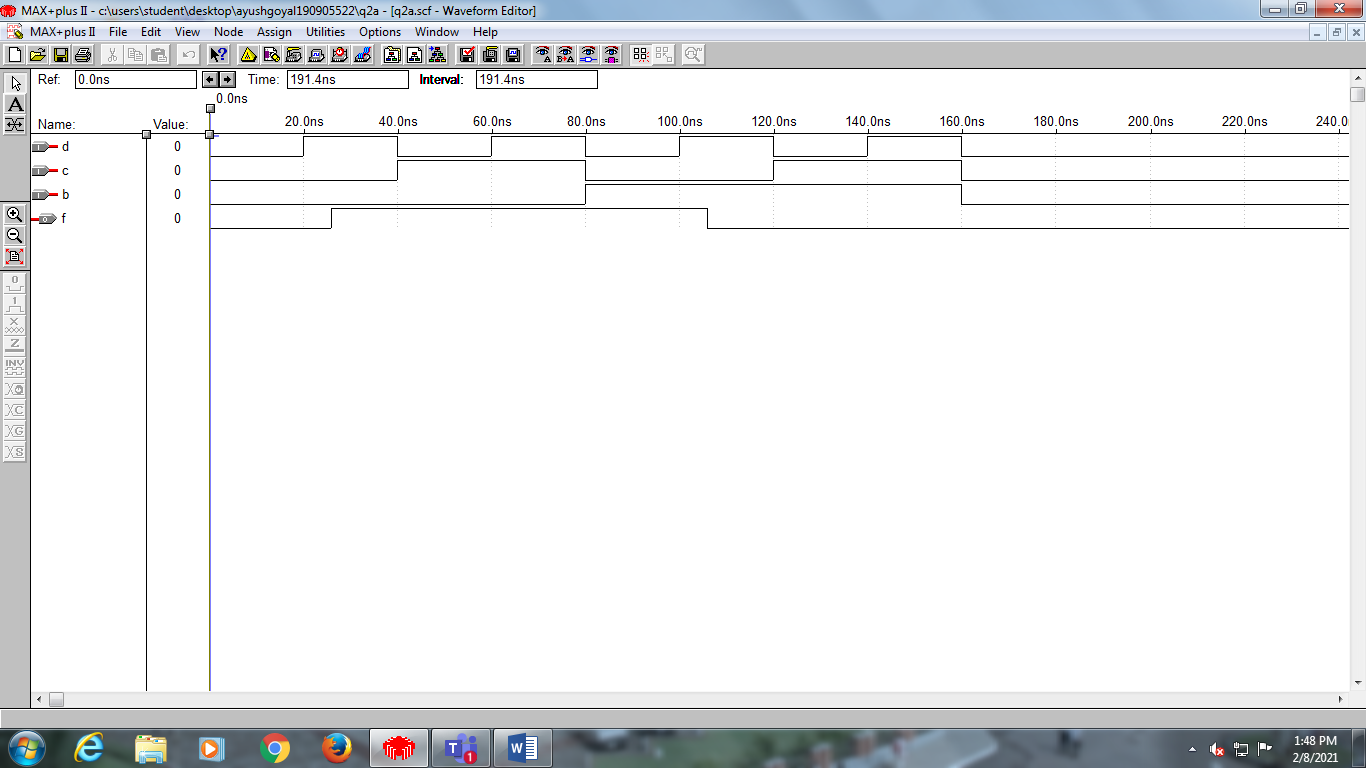
input a,b,c,d;

output f;

assign f = (b&(~c)&(~d))|((~b)&d)|((~b)&c);

endmodule

Output:



1. Verilog Code:

module q2b(a,b,c,d,f);

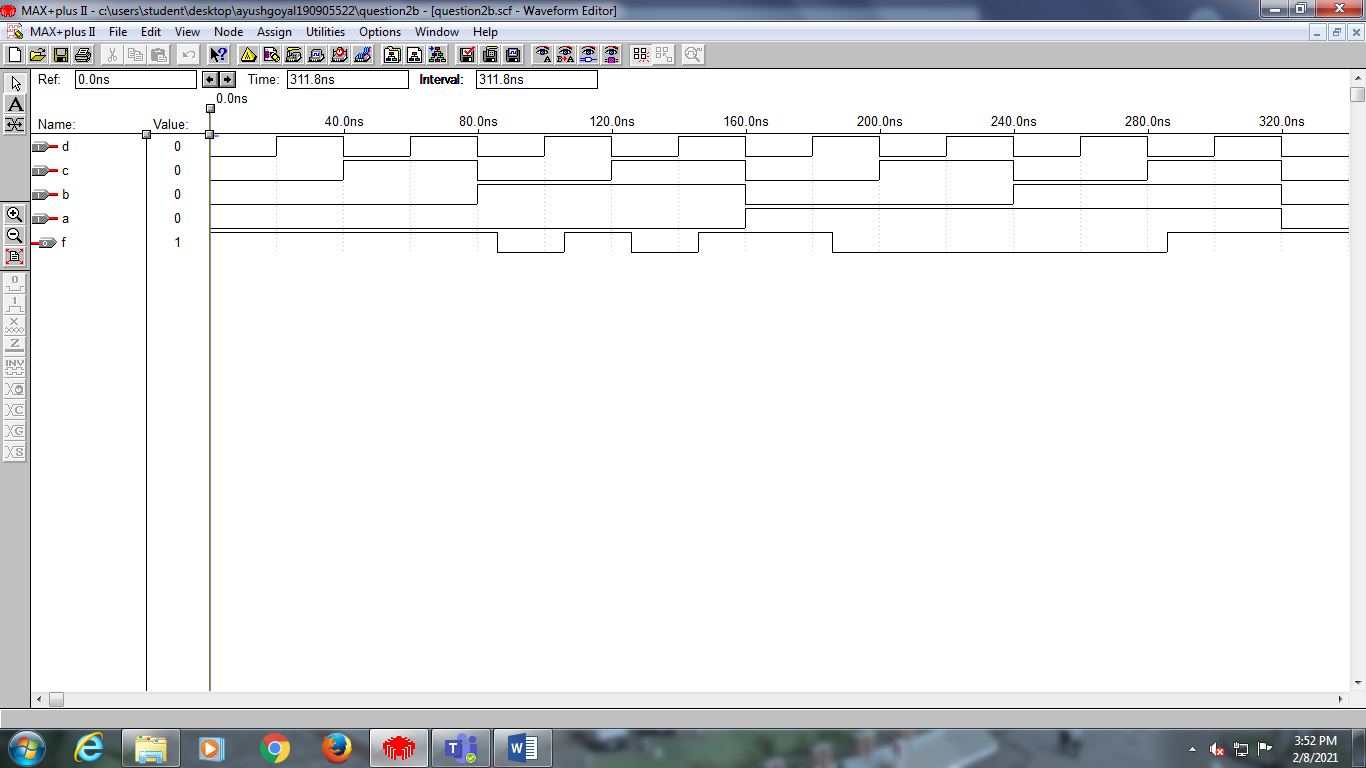
input a,b,c,d;

output f;

assign f = (a&b&c)|(~b&~c&~d)|(~a&d)|(~a&~b);

endmodule

Output:



3) Verilog Code:

module q3(a,b,c,d,f);

input a,b,c,d;

output f;

nand(p,a,a);

nand(q,c,c);

nand(r,p,q);

nand(s,p,d);

nand(t,q,b);

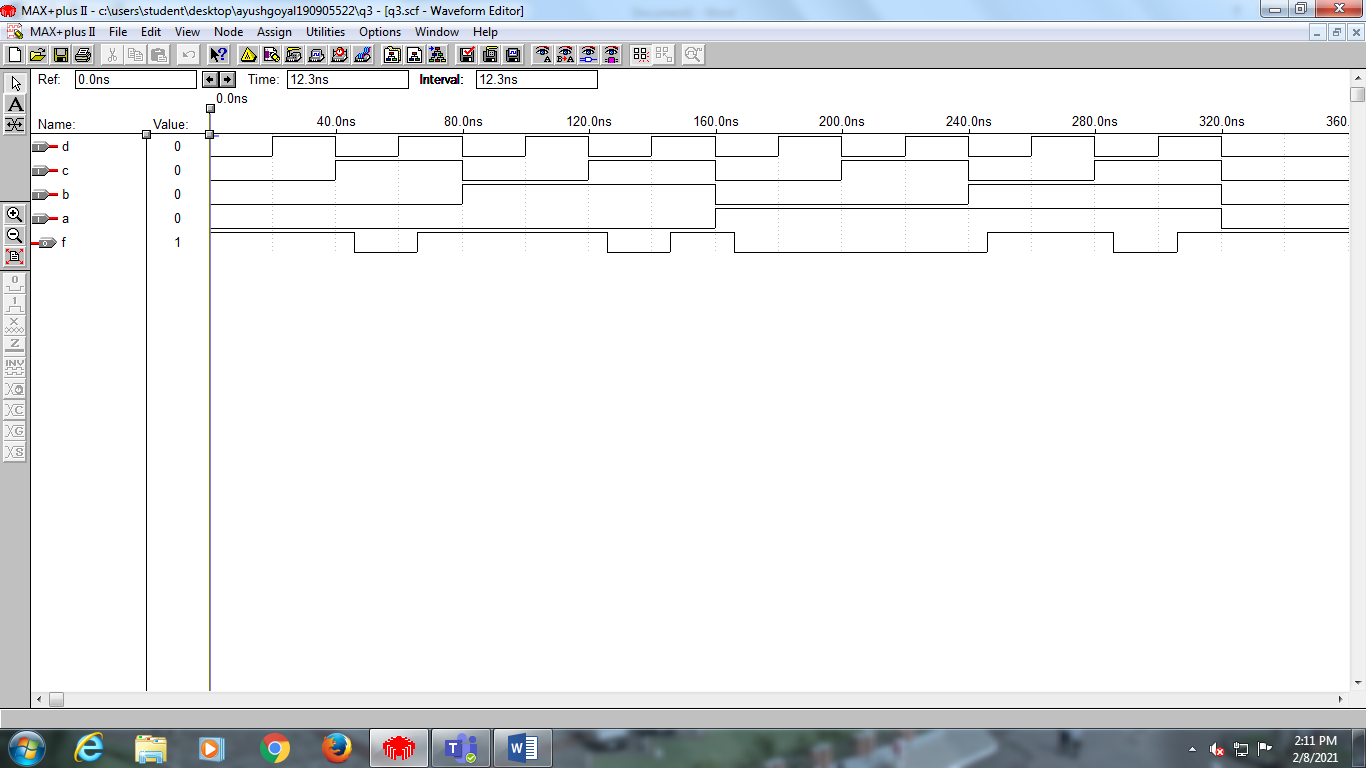
nand(u,d,b);

nand(f,r,s,t,u);

//*assign f = (~a&~c)|(~a&d)|(b&~c)|(b&d); is behavioural*

endmodule

Output:



Additional Question 1)

Verilog code:

module addn1(a,b,c,d,f);

input a,b,c,d;

output f;

nor(g1,a,a);

nor(g2,b,b);

nor(g3,c,c);

nor(g4,d,d);

nor(g5,g3,g4);

nor(g6,g2,d);

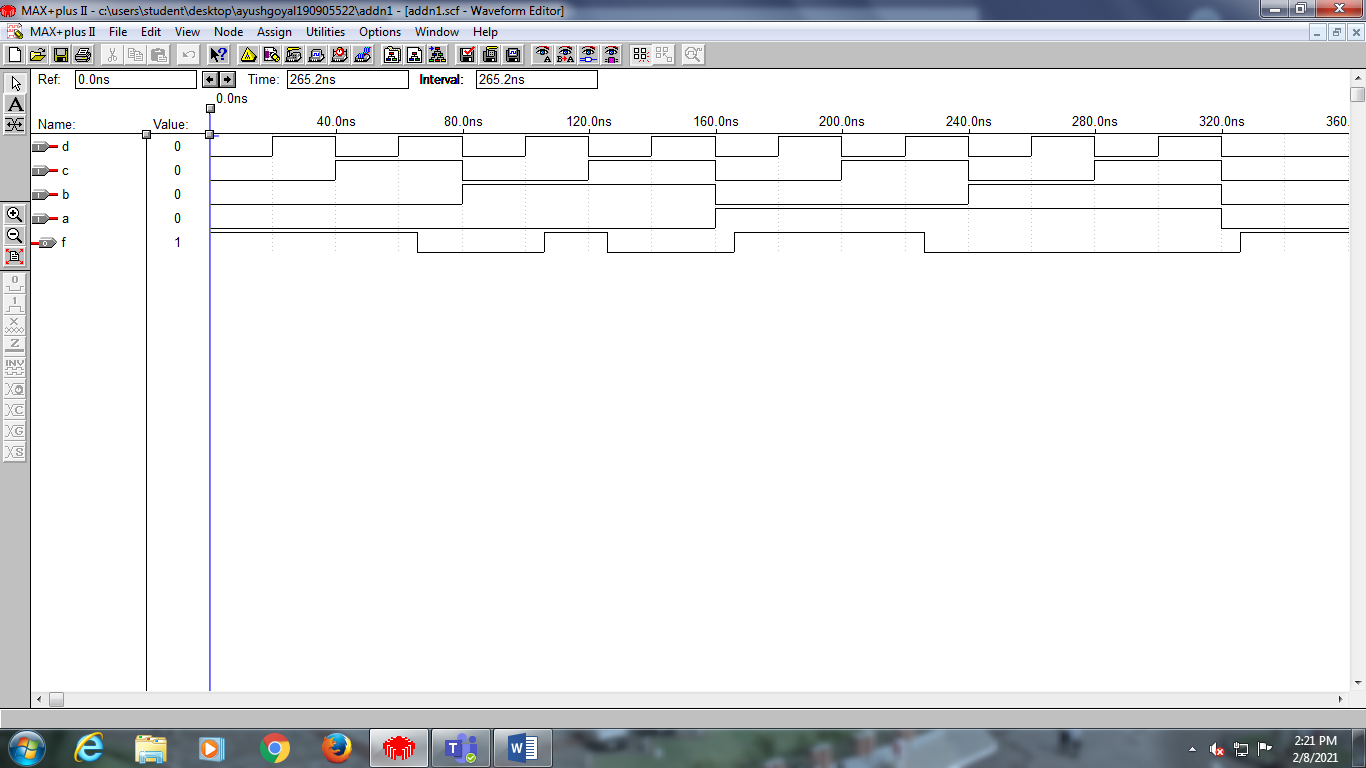
nor(g7,g1,g2);

nor(f,g5,g6,g7);

//assign f = (~b&~d)|(~a&~c&d)|(~b&~c); is behavioural

endmodule

Output:



END OF LAB 1(Week 1)