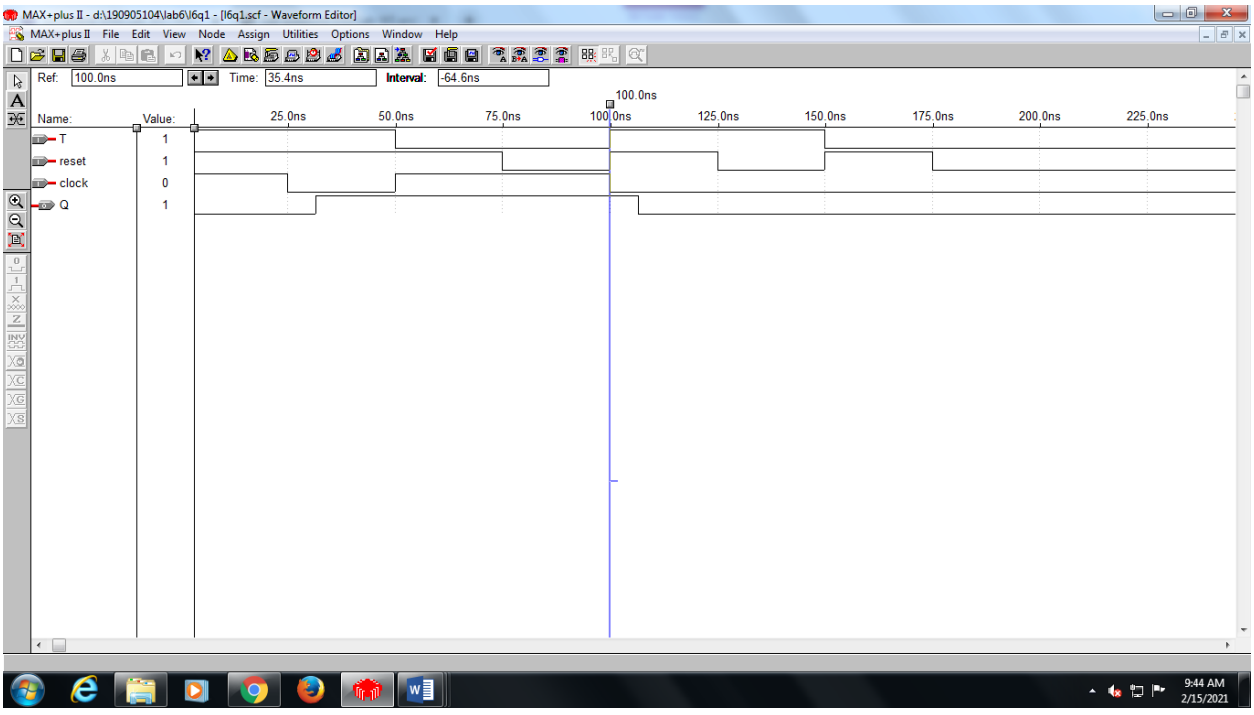
Week6

1)

// Write behavioral Verilog code for a negative edge triggered T FF with asynchronous active low reset.

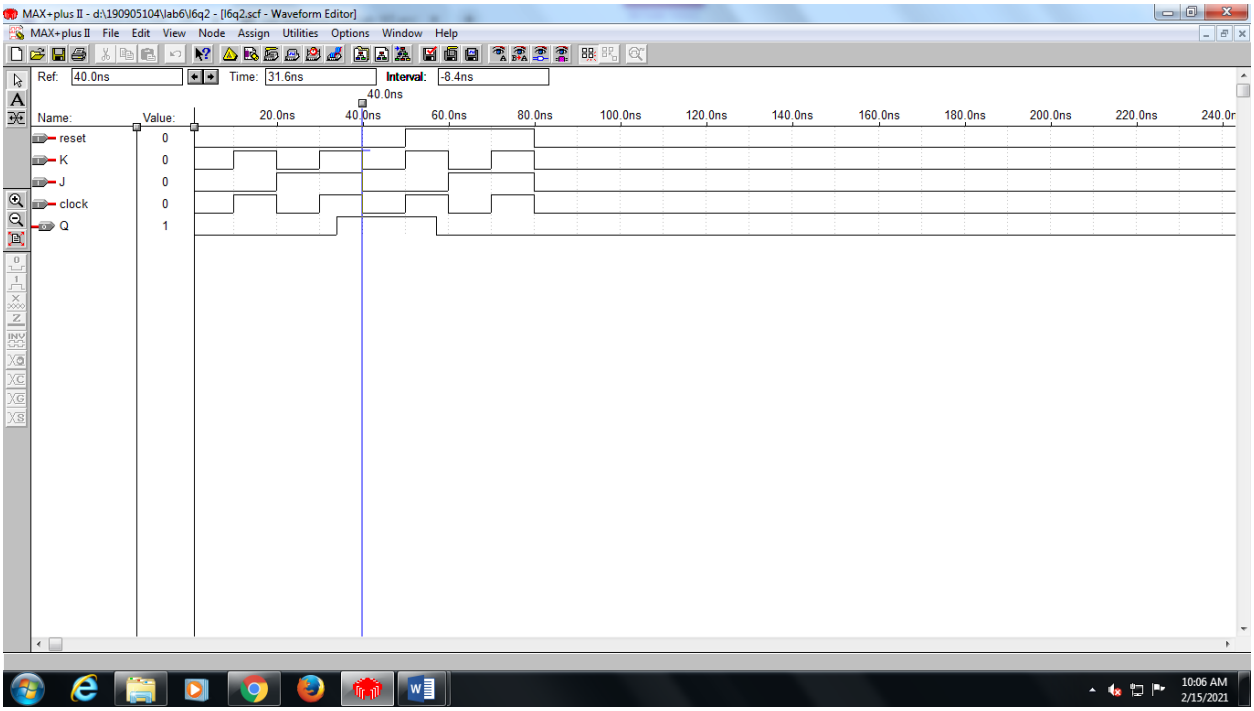
```
module l6q1(T, clock, reset, Q);
input T, clock, reset;
output Q;
reg Q;
always @(negedge clock)
begin
if(T==1)
Q <= ~Q;
if(reset==0)
Q <= 0;
end
endmodule
```



2)

// Write behavioral Verilog code for a positive edge-triggered JK FF with synchronous active high reset

```
module l6q2(J, K, clock, reset, Q);
input J, K, clock, reset;
output Q;
reg Q;

always @(posedge clock or posedge reset)
begin
case({J, K})
0: Q <= Q;
1: Q <= 0;
2: Q <= 1;
3: Q <= ~Q;
endcase
if(reset==1)
Q <= 0;
end
endmodule
```



3)

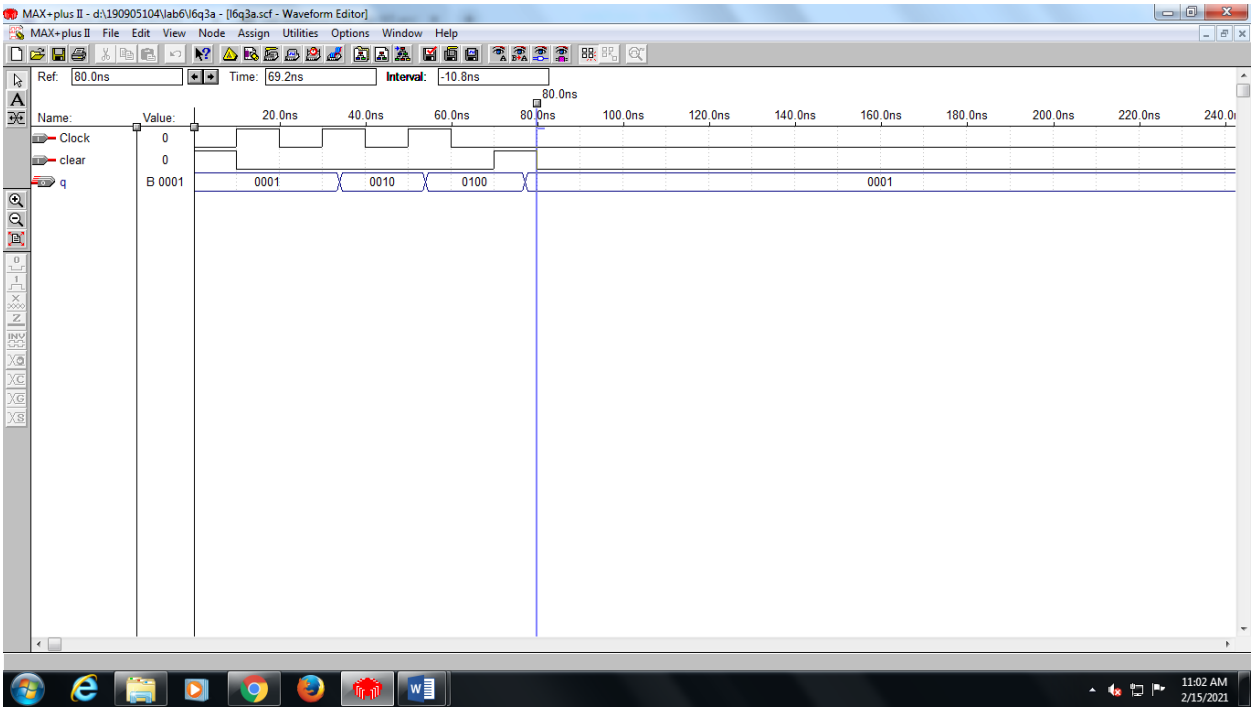
a)

```
// 4 bit ring counter

module l6q3a(q, Clock, clear);

input Clock, clear;

output [0:3]q;

reg[0:3]q;


always@(posedge Clock or posedge clear)

if(clear==1)

q<=4'b0001;

else

begin

q[3]<=q[0];

q[2]<=q[3];

q[1]<=q[2];
```

q[0]<=q[1];

end

endmodule



b)

// 5 bit Johnson counter.

module DFlipFlop(D, clock, reset, Q);

input D, clock, reset;

output Q;

reg Q;

always @(posedge clock)

begin

if(reset)

Q <= 0;

else if(D)

Q <= 1;

```
else
Q <= 0;
end
endmodule

module l6q3b(clock, reset, Q);
input clock, reset;
output [0:4]Q;
wire [0:4]Q;

DFlipFlop d0(~Q[4], clock, reset, Q[0]);
DFlipFlop d1(Q[0], clock, reset, Q[1]);
DFlipFlop d2(Q[1], clock, reset, Q[2]);
DFlipFlop d3(Q[2], clock, reset, Q[3]);
DFlipFlop d4(Q[3], clock, reset, Q[4]);
endmodule
```