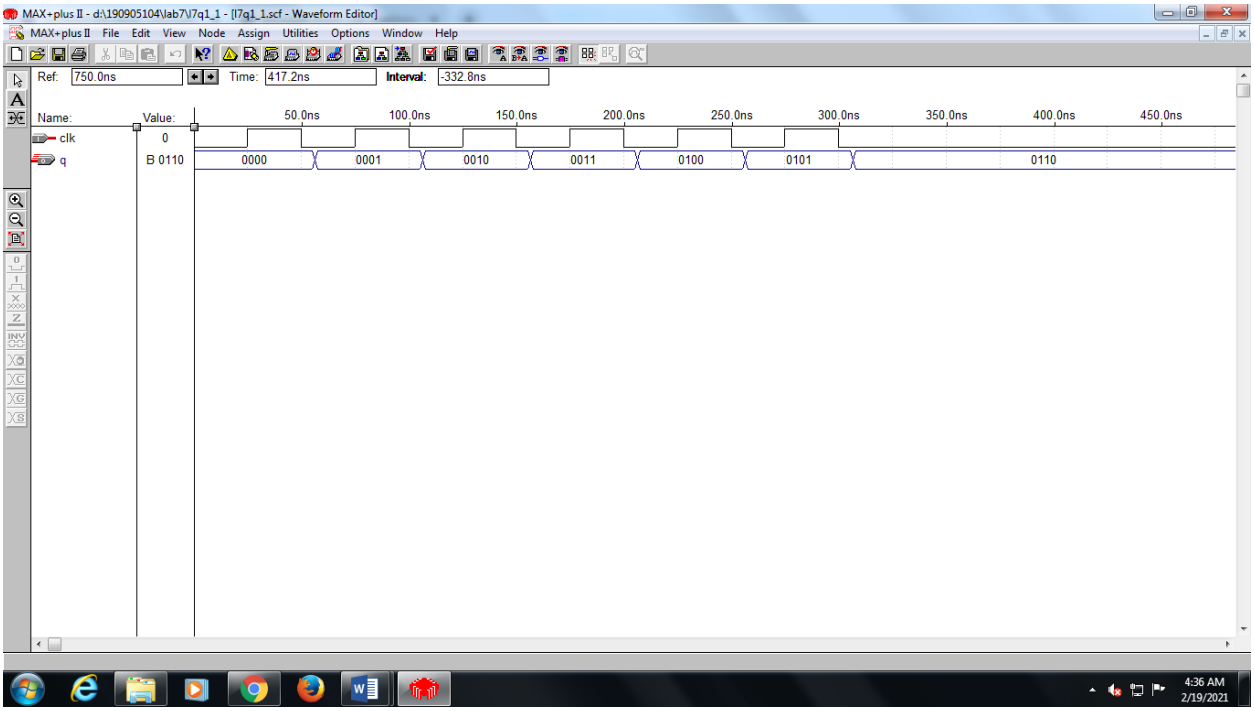
Week 7

1)

```
// 4 bit synchronous up counter
module tf(t,clk,q);
input t,clk;
output q;
reg q;
always @(negedge clk)
if(!t)
q<=q;
else
q<=~q;
endmodule

module l7q1_1(clk,q);
input clk;
output [3:0]q;

tf stage0(1,clk,q[0]);
tf stage1(1&&q[0],clk,q[1]);
tf stage2(1&&q[0]&&q[1],clk,q[2]);
tf stage3(1&&q[0]&&q[1]&&q[2],clk,q[3]);
endmodule
```



2)

// 3 bit synchronous up/down counter with a control input up/ⓜ$down$ ⓜⓜⓜⓜⓜⓜ. If up/$down$ ⓜⓜⓜⓜⓜⓜⓜ = 1, then the circuit should behave as an up counter. If up/$down$ ⓜⓜⓜⓜⓜⓜⓜ = 0, then the circuit should behave as a down counter.

```
module jkf(j,k,clk,y);
input j,k,clk;
output y;
reg y;
always @(posedge clk)
case({j,k})
2'b00:y<=y;
2'b01:y<=0;
2'b10:y<=1;
2'b11:y<=~y;
endcase
endmodule
```

```
module l7q2(clk,control,out);

input clk,control;

output [2:0]out;

wire a,b;

jkf stage0(1,1,clk,out[0]);

assign a = (control&&out[0])||(~control&&~out[0]);

jkf stage1(a,a,clk,out[1]);

assign b = (control&&out[0]&&out[1])||(~control&&~out[0]&&~out[1]);

jkf stage2(b,b,clk,out[2]);

endmodule
```