

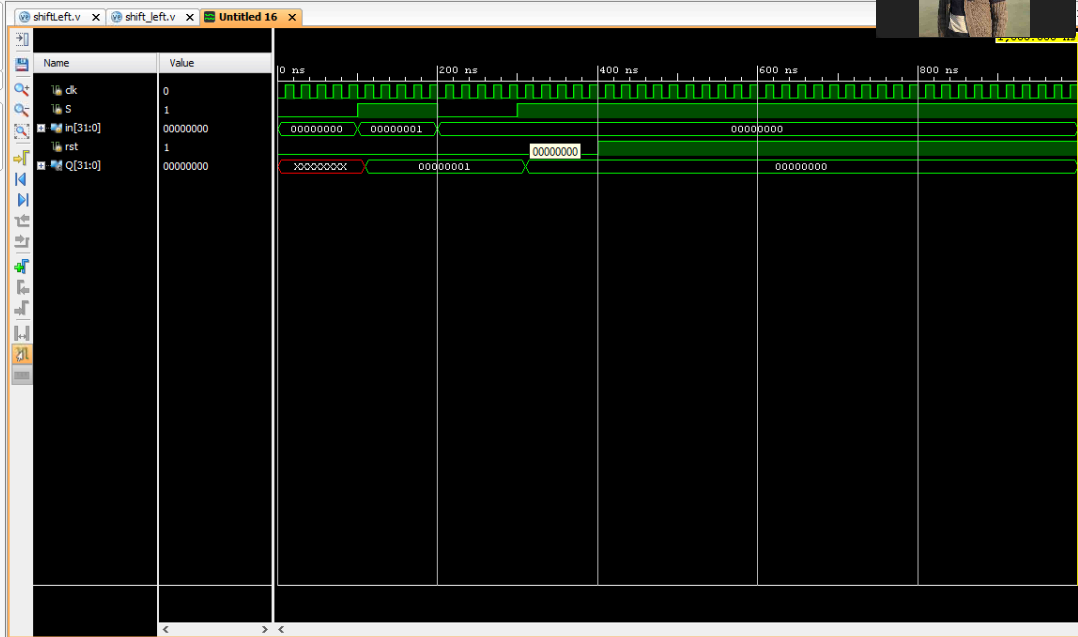
**lab report 3**

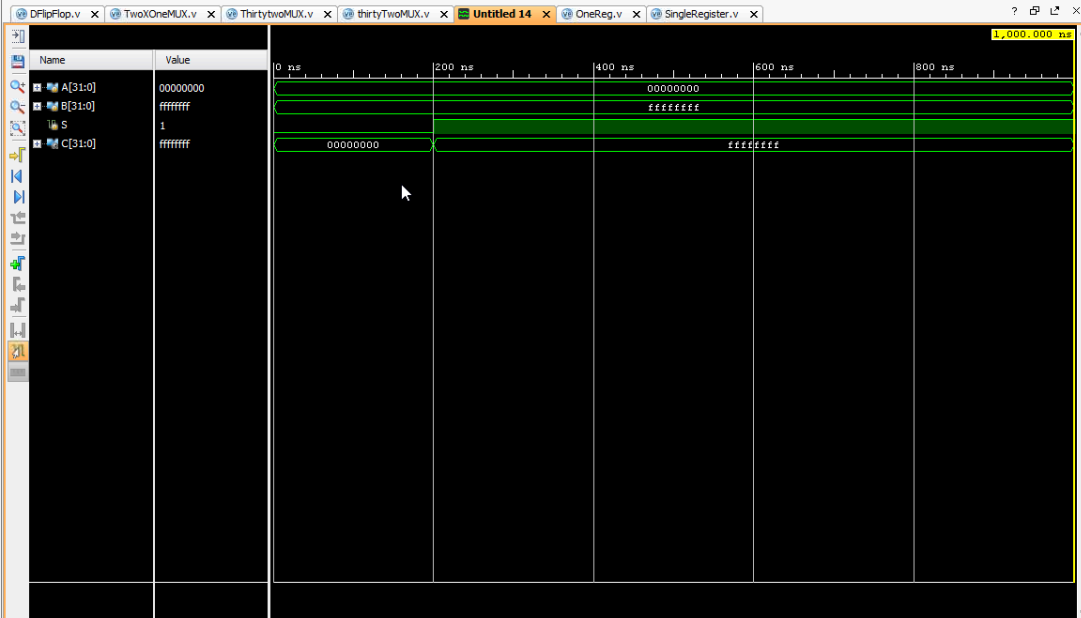
Andrew Nady

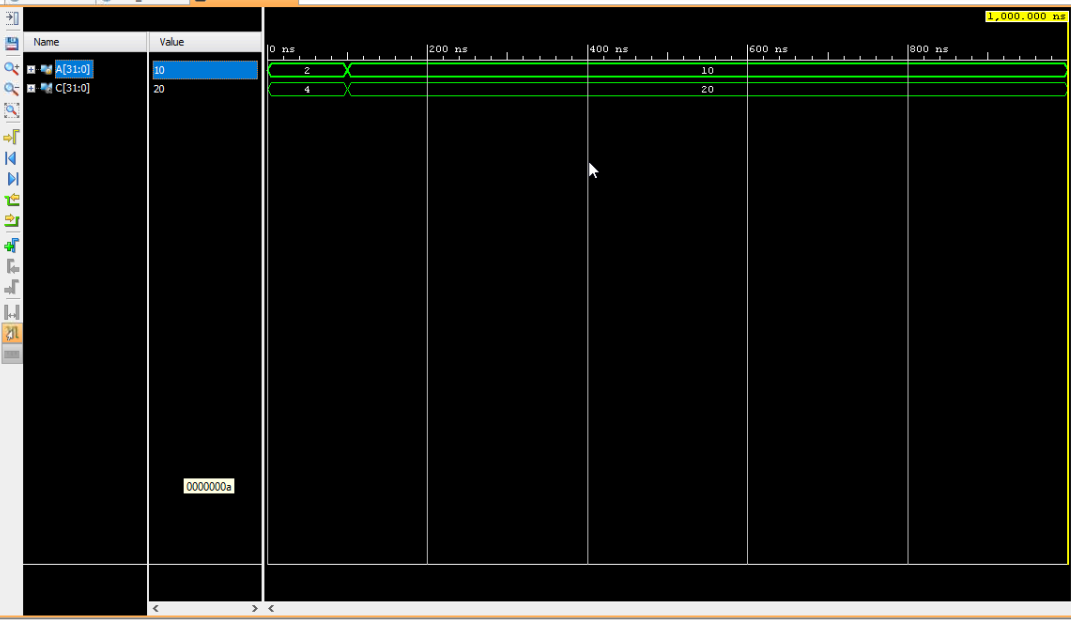
900184042

Results:

Exp1:



Ex2

Ex3:

Ex4:



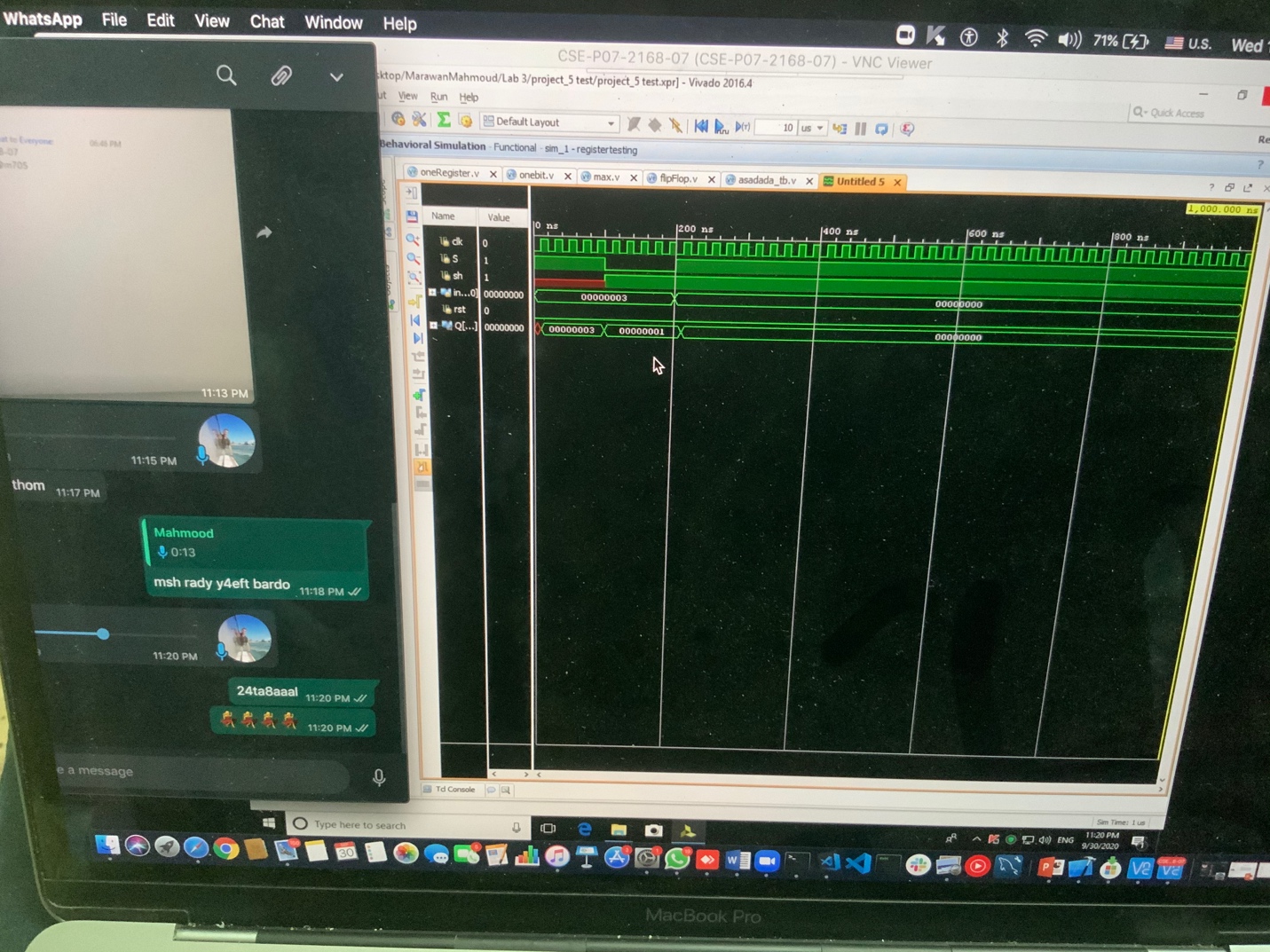
Question 1:  
I add the case If it is 01, then shift to left

|  |
| --- |
| **` `timescale** **1**ns / **1**ps  **module** OneReg(**input** clk,**input** **S**,**input** sh,**input** [**31**:**0**] in,**input** rst, **output** **reg**[**31**:**0**] **Q** );  **wire** [**1**:**0**] Sl;  **assign** **SL**= {**S**,sh};  **wire**[**31**:**0**] out;  **genvar** i;  **generate**  **for** (i=**0**;i<=**31**;i=i+**1**)  **begin**  onebitReg mod1 (in[i],**S**,clk,rst,out[i]);  **end**  **endgenerate**  **always** @(\*)  **begin**  **case**(**SL**)  **2'b01**: **Q**={**1'b0**,**Q**[**31**:**1**]};  **default**: **Q**=out;  **endcase**  **end**  **endmodule** |

Testbench:

|  |
| --- |
| // file: asadada\_tb.v  // author: @andrewhany  // Testbench asadada\_tb  **`timescale** **1**ns / **1**ps  **module** registertesting() ;  **reg** clk;  **reg** **S**;  **reg** sh;  **reg** [**31**:**0**] in;  **reg** rst;  **wire** [**31**:**0**] **Q** ;  OneReg regis(clk, **S**,sh, in, rst,**Q**);  **initial**  **begin**  clk = **0**;  **forever** #**10**  clk=~clk;  **end**  **initial**  **begin**  **S**=**1**;  in=**3**;  rst=**0**;    #**100**  **S**=**0**;  sh=**1**;  #**100**  **S**=**1**;  in=**0**;      **end**  **endmodule** |

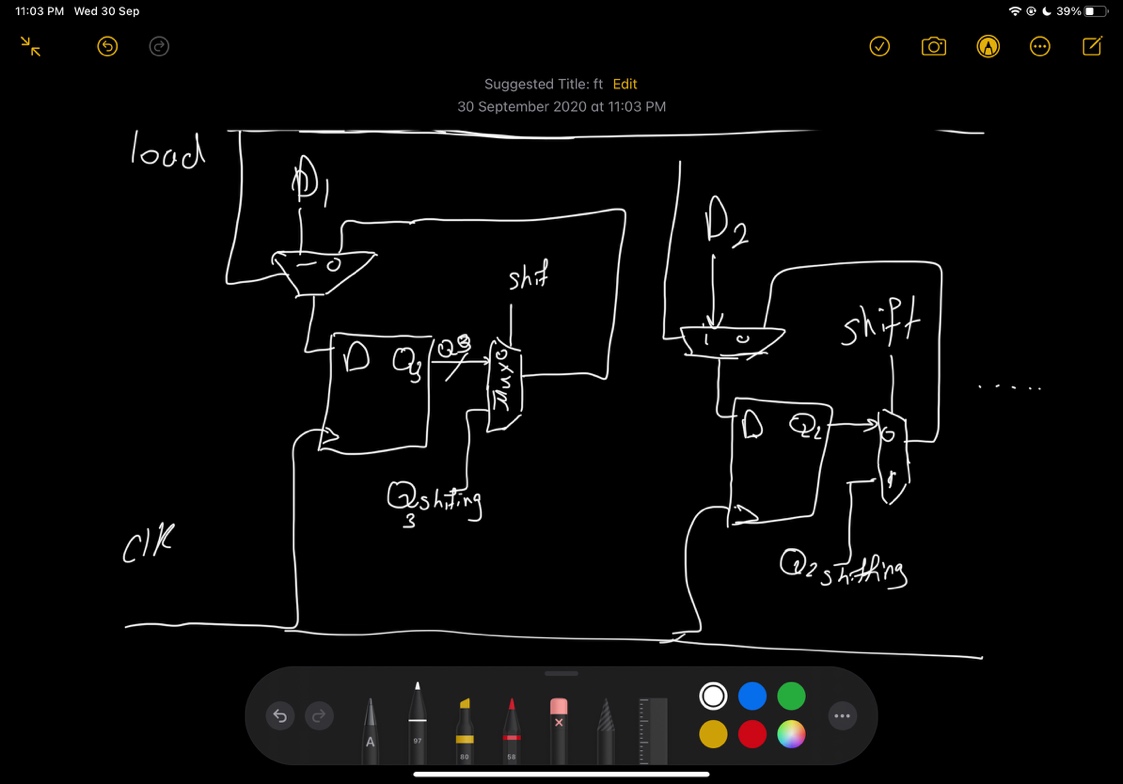
Simulation:



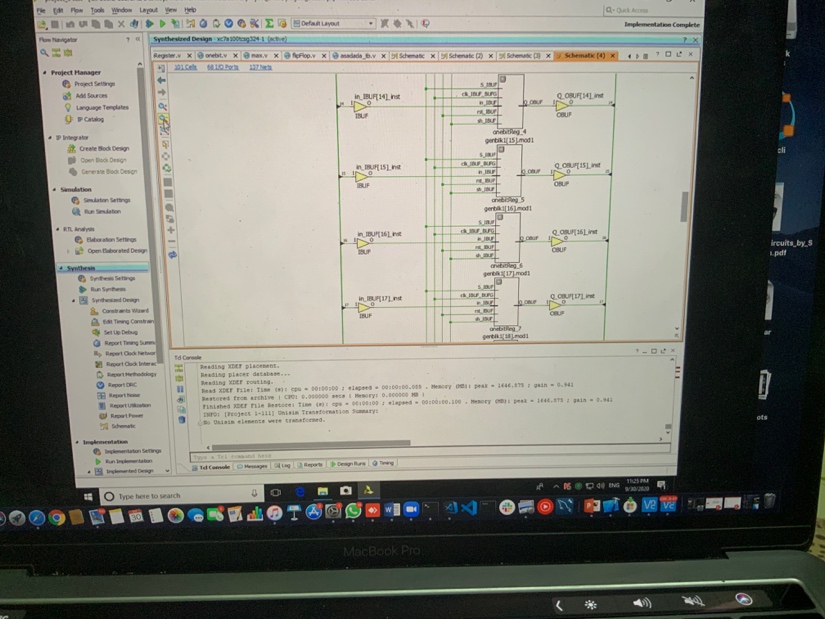
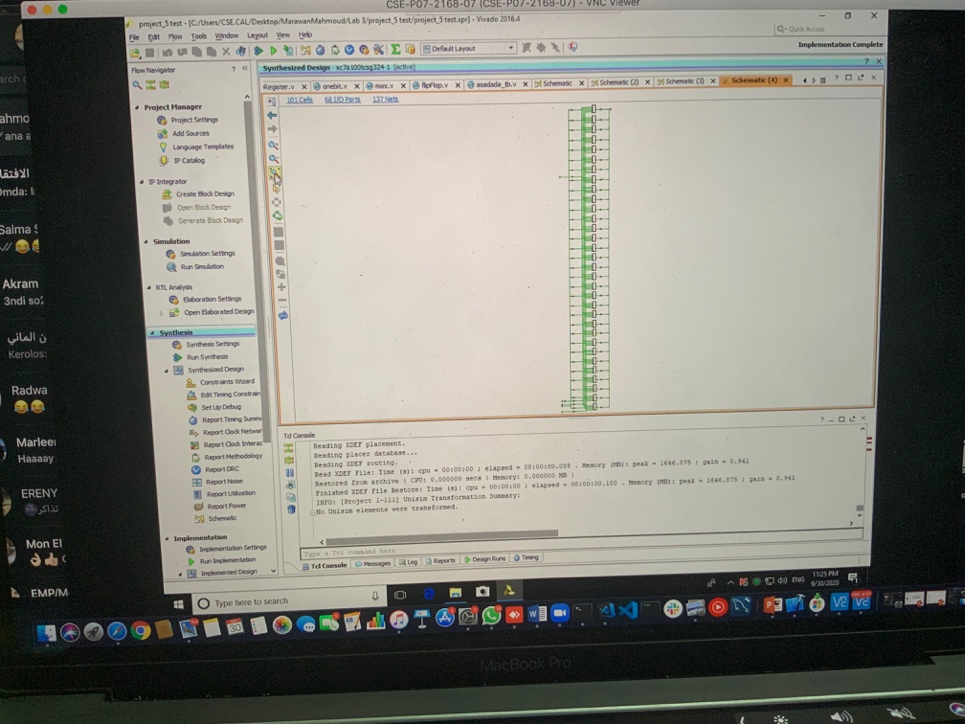
When I make the load by 1, I entered the value 3, when I make the load by zero and the shifting flag with 1, it shifts it to the right 011 to 001

Question 2:

* I tried to mak it by hand, to figure out the logic
* Here if the load = 1 then it will read a new data
* If it is load ==0, then we load at shif flag if it is 1 then we shift the Q and pass it to the flipFlop, otherwise it will pass the normal Q.



Schematic:



Question 3:

* To support only 8 operation, so what we need is 3 bits only from the opcode

