Three sons and a father are partners in running a company. The three sons are "A", "B", and "C". The father is "D". To make things work smoothly in the company they have decided that all resolutions will pass if and only if there are a majority of votes (i.e., at least three out of four "yes" votes) except that the father will have veto power: if he votes "no" then the resolution fails no matter what. Furthermore, if there is a tie (i.e., two "yes" and two "no" votes), then the side that includes the father wins.

a) (0.2 points) Using the form that "1" represents a "yes" vote and "0" represents a "no" vote, fill in the truth table for a logic circuit that will compute the result of the voting. Here "F" equal to "1" represents that the resolution passes and F equal to "0" represents that it fails.

A	B	C	D	F

b) (0.1 points) Write the canonical sum-of-products expression for F.

c) (0.7 points) Next, using a Karnaugh map realize the simplest circuit using a sum-of-products form for this function.

AB \ CD	00	01	11	10
00				
01				
11				
10				

d) (0.3 points) Draw the circuit using "AND", "OR" and "NOT" gates

e) (0.2 points) Draw the circuit using only 2-input NAND gates

f) (1 point) Now suppose there is a change in the veto power of the father. If there is consensus on an issue (i.e., all sons agree on it) and the father vetoes, it is called an extreme veto. We

want to limit the father's power by allowing him to veto at most three times in a row. Design a Moore state machine that outputs P, which states the veto power of the father. If P=1, the father can veto and if P=0, he cannot. Note that if the father vetoes twice, and then does not veto, he still has a chance to veto the next three rounds.

g) (1.5 points) Show the state assignment and implement your state machine using JK flip-flops and AND, OR, NOT logic.