1.) For the following K-Map, produce a minimized Sum of Products equation, and the resulting circuit diagram using ONLY NAND and/or INVERTER gates.
2.) Using the rules of Boolean Algebra, minimize this function. You do not need to list the rules, but you must show the steps you use.

\[ F = (A + BC) + A(B + BCD + CDE) \]
3.) Draw the **state diagram** for the following circuit – your diagram should be as simple as possible:

Our FSM has a single input, and determines whether **the last three input values, including the current one**, contained an odd or even number of trues. That is, if we have just seen two 0’s, and are now seeing a 1, that means we’ve seen an odd number of trues. Output TRUE if we’ve seen an odd number of trues, and false otherwise.

Note that at startup, we assume the two previous inputs were both false. Thus, if the first input is 1 we output TRUE, and if the first input is 0 we output FALSE.
4.) For the following Verilog, produce the corresponding minimized circuit diagram. Your diagram can use any basic gates and DFFs.

```verilog
module test (clk, in, out);
  input logic clk, in;
  output logic out;
  enum { A=2'b00, B=2'b01, C=2'b10, D=2'b11} ps, ns;

  always_comb begin
    if (~in)
      ns = A;
    else
      case (ps)
        A: ns = B;
        B: ns = C;
        C: ns = D;
        D: ns = C;
      endcase
    end
    out = (ps == D);
  end

  always_ff @(posedge clk) begin
    ps <= ns;
  end
endmodule
```