EECS151/251A - Homework 3 Solutions
Problem 1

(1.1)  
\[ \text{Out} = \overline{ABC}D + \overline{AB}CD + \overline{AB}C\overline{D} + \overline{AB}CD + \overline{ABC}D + \overline{ABCD} + \overline{ABC}\overline{D} + \overline{ABCD} + \overline{ABC}D + \overline{ABCD} \]

(1.2)  
\[ Q = \overline{A}B + \overline{BC} + C\overline{D} \]
(1.3)

Figure 1: Direct implementation of the simplified SOP representation.

Figure 2: NAND-only implementation of the simplified SOP representation.
Problem 2

(2.1)

\[ Q = \overline{B}\overline{C}D + A\overline{B}\overline{C} + \overline{A}\overline{D}\overline{E} + BD\overline{E} + B\overline{C}E\overline{F} \]
\( Q = (CE + \overline{AB}D + \overline{A}DE + BCD + BEF) \)
\( Q = (\overline{C} + \overline{E})(A + B + D)(A + D + E)(B + \overline{C} + \overline{D})(\overline{B} + \overline{E} + \overline{F}) \)

For this choice of \( x \) values, the functions are identical - they have the same truth table if evaluated. However, there can be configurations such that the same \( x \) is utilized differently for SOP and POS representations, resulting in different functions representing the same specification truth table.
Problem 3

(3.1)

(3.2)

```verilog
module pattern_detector (
  input wire in,
  input wire clk,
  output wire out);

reg [2:0] state = 0;
localparam M0 = 0;
localparam M1 = 1;
localparam M2 = 2;
localparam M3 = 3;
localparam M4 = 4;
localparam M5 = 5;
```
assign out = (state == M5) ? 1'b1 : 1'b0;

always @ (posedge clk)
begin

case(state)
M0:
begin
  if(in == 1) state <= M1;
  else if(in == 0) state <= M0;
end

M1:
begin
  if(in == 1) state <= M1;
  else if(in == 0) state <= M2;
end

M2:
begin
  if(in == 1) state <= M1;
  else if(in == 0) state <= M3;
end

M3:
begin
  if(in == 1) state <= M4;
  else if(in == 0) state <= M0;
end

M4:
begin
  if(in == 1) state <= M1;
  else if(in == 0) state <= M5;
end

M5:
begin
  if(in == 1) state <= M1;
  else if(in == 0) state <= M0;
end
endcase
end
endmodule
module pattern_detector (in, clk, out);

reg [2:0] state = 0;
localparam M0 = 0;
localparam M1 = 1;
localparam M2 = 2;
localparam M3 = 3;
localparam M4 = 4;
localparam M5 = 5;
assign out = (state == M5) ? 1'b1 : 1'b0;

always @(posedge clk)
begin
  case(state)
    M0:
      begin
        if(in == 1) state <= M1;
        else if(in == 0) state <= M0;
      end
    M1:
      begin
        if(in == 1) state <= M1;
        else if(in == 0) state <= M2;
      end
    M2:
      begin
        if(in == 1) state <= M1;
        else if(in == 0) state <= M3;
      end
    M3:
      begin
        if(in == 1) state <= M4;
        else if(in == 0) state <= M0;
      end
    M4:
      begin
        if(in == 1) state <= M1;
        else if(in == 0) state <= M5;
      end
    M5:
      begin
        if(in == 1) state <= M1;
        else if(in == 0) state <= M3;
      end
  endcase
end
module pattern_detector (
  input wire in,
  input wire clk,
  output reg out);

reg [2:0] state = 0;
localparam M0 = 0;
localparam M1 = 1;
localparam M2 = 2;
localparam M3 = 3;
localparam M4 = 4;

always @(posedge clk)
begin

case(state)
M0:
begin
    out <= 1'b0;
    if(in == 1) state <= M1;
    else if(in == 0) state <= M0;
end

M1:
begin
    out <= 1'b0;
    if(in == 1) state <= M1;
    else if(in == 0) state <= M2;
end

M2:
begin
    out <= 1'b0;
    if(in == 1) state <= M1;
    else if(in == 0) state <= M3;
end

M3:
begin
    out <= 1'b0;
    if(in == 1) state <= M4;
    else if(in == 0) state <= M0;
end

M4:
begin
    if(in == 1) state <= M1;
    else if(in == 0)
    begin
        state <= M2;
        out <= 1'b1;
    end
end
endcase
end
endmodule