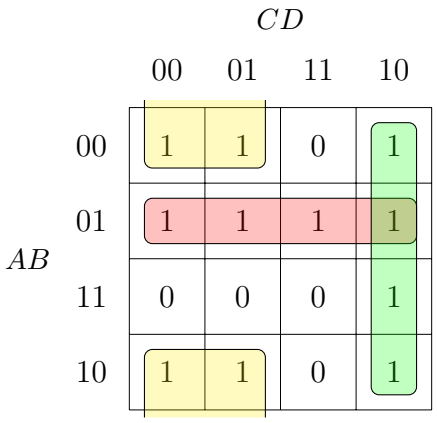
EECS151/251A - Homework 3 Solutions

Problem 1

(1.1)

$$Out = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}CD + \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + \overline{A}BCD$$
$$+A\overline{B}\overline{C}\overline{D} + A\overline{B}\overline{C}D + A\overline{B}C\overline{D} + ABC\overline{D}$$

(1.2)



$$Q = \overline{A}B + \overline{B}\overline{C} + C\overline{D}$$

(1.3)

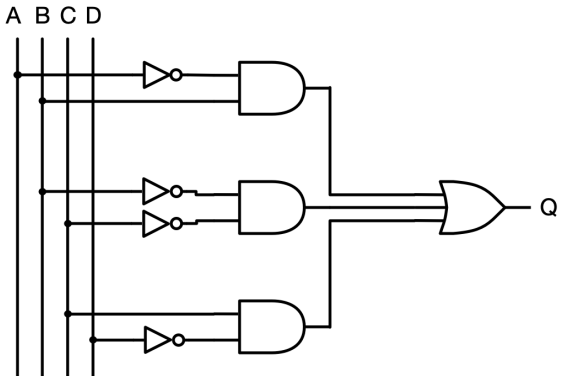


Figure 1: *Direct implementation of the simplified SOP representation.*

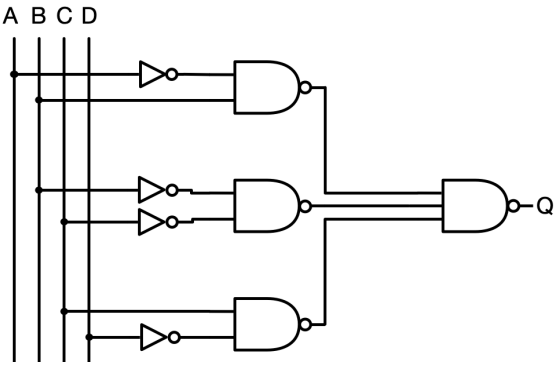
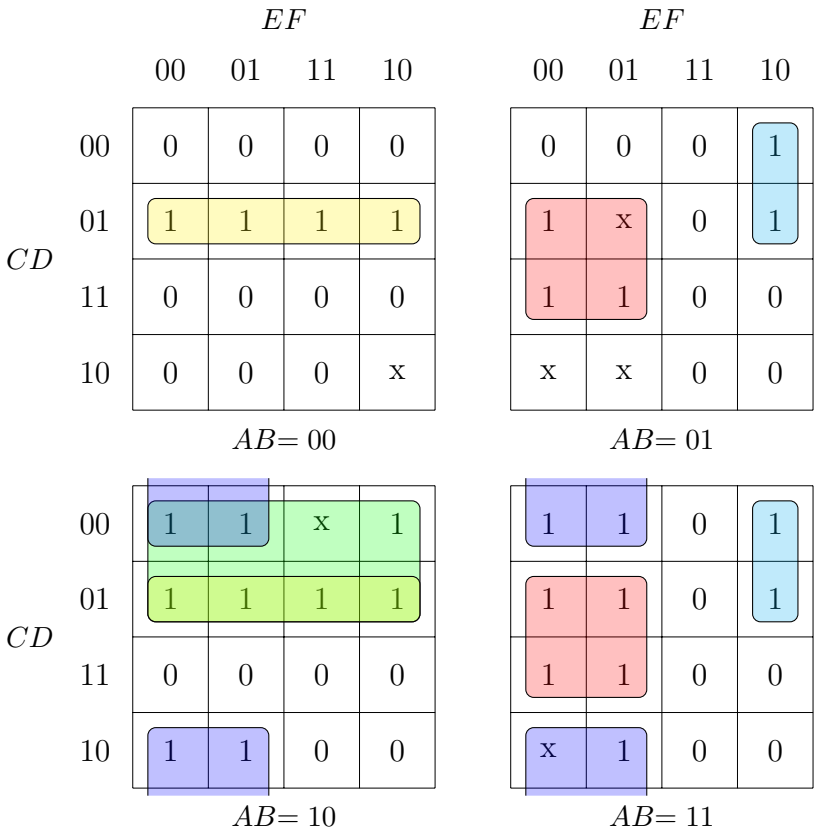


Figure 2: *NAND-only implementation of the simplified SOP representation.*

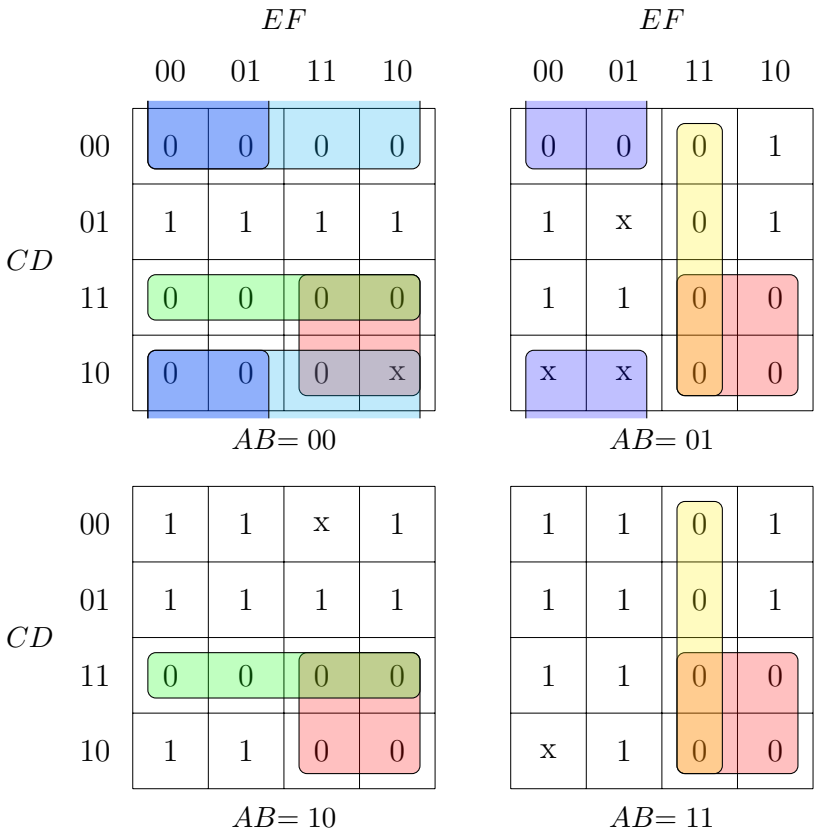
Problem 2

(2.1)



$$Q = \overline{B}\overline{C}D + A\overline{B}\overline{C} + A\overline{D}\overline{E} + BD\overline{E} + B\overline{C}E\overline{F}$$

(2.2)



$$Q = \overline{(CE + \overline{A}\overline{B}\overline{D} + \overline{A}\overline{D}\overline{E} + \overline{B}CD + BEF)}$$

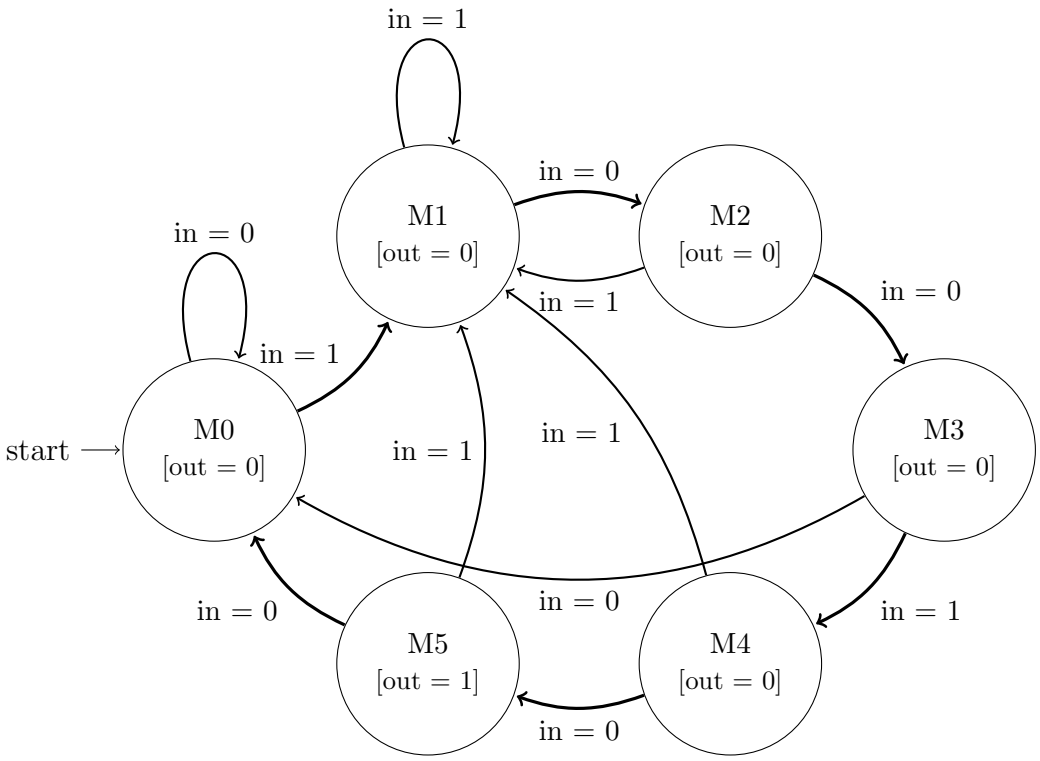
$$Q = (\overline{C} + \overline{E})(A + B + D)(A + D + E)(B + \overline{C} + \overline{D})(\overline{B} + \overline{E} + \overline{F})$$

(2.3)

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)

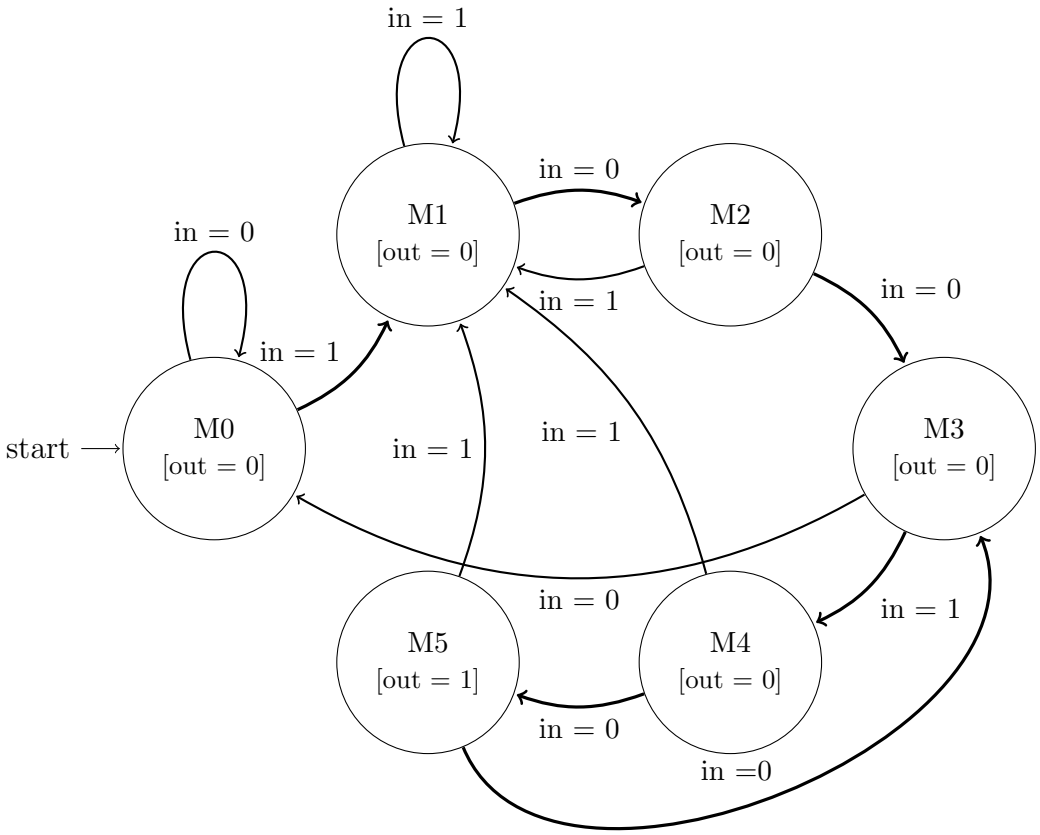


(3.2)

```
1 module pattern_detector (
2   input wire in,
3   input wire clk,
4   output wire out);
5
6   reg [2:0] state = 0;
7   localparam M0 = 0;
8   localparam M1 = 1;
9   localparam M2 = 2;
10  localparam M3 = 3;
11  localparam M4 = 4;
12  localparam M5 = 5;
13
```

```
14   assign out = (state == M5) ? 1'b1 : 1'b0;
15
16   always @ (posedge clk)
17   begin
18
19   case(state)
20   M0:
21   begin
22     if(in == 1) state <= M1;
23     else if(in == 0) state <= M0;
24   end
25
26   M1:
27   begin
28     if(in == 1) state <= M1;
29     else if(in == 0) state <= M2;
30   end
31
32   M2:
33   begin
34     if(in == 1) state <= M1;
35     else if(in == 0) state <= M3;
36   end
37
38   M3:
39   begin
40     if(in == 1) state <= M4;
41     else if(in == 0) state <= M0;
42   end
43
44   M4:
45   begin
46     if(in == 1) state <= M1;
47     else if(in == 0) state <= M5;
48   end
49
50   M5:
51   begin
52     if(in == 1) state <= M1;
53     else if(in == 0) state <= M0;
54   end
55   endcase
56   end
57 endmodule
```

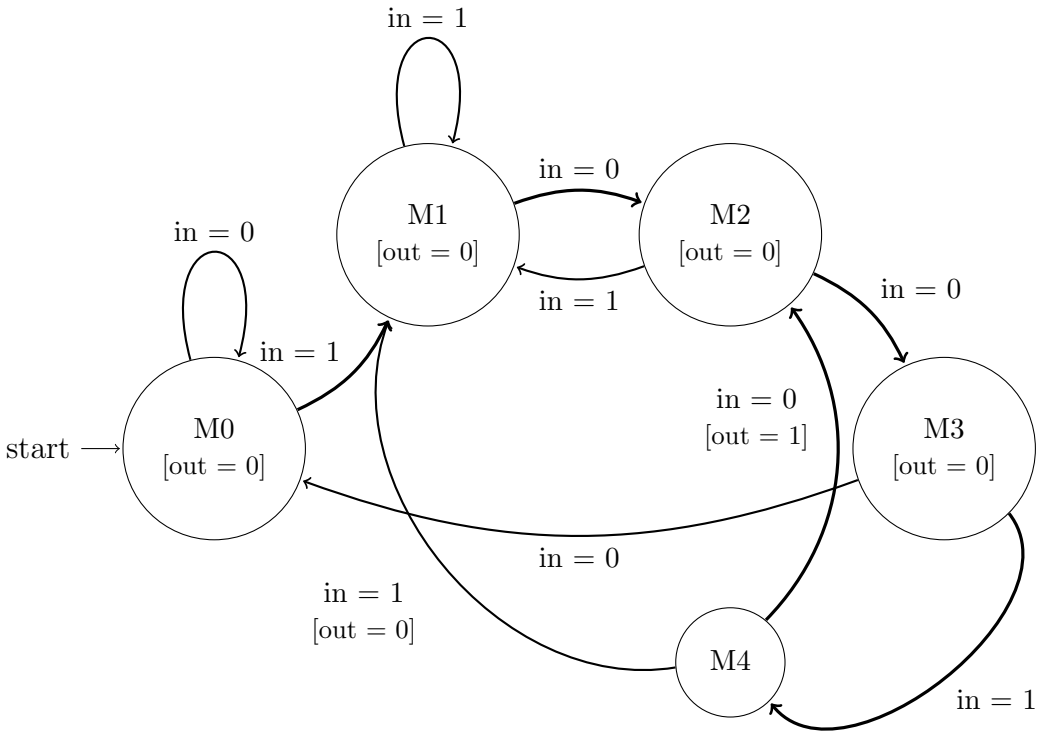
(3.3)




```
1 module pattern_detector (
2   input wire in,
3   input wire clk,
4   output wire out);
5
6   reg [2:0] state = 0;
7   localparam M0 = 0;
8   localparam M1 = 1;
9   localparam M2 = 2;
10  localparam M3 = 3;
11  localparam M4 = 4;
12  localparam M5 = 5;
13
```

```
14    assign out = (state == M5) ? 1'b1 : 1'b0;
15
16    always @ (posedge clk)
17    begin
18
19    case(state)
20    M0:
21    begin
22      if(in == 1) state <= M1;
23      else if(in == 0) state <= M0;
24    end
25
26    M1:
27    begin
28      if(in == 1) state <= M1;
29      else if(in == 0) state <= M2;
30    end
31
32    M2:
33    begin
34      if(in == 1) state <= M1;
35      else if(in == 0) state <= M3;
36    end
37
38    M3:
39    begin
40      if(in == 1) state <= M4;
41      else if(in == 0) state <= M0;
42    end
43
44    M4:
45    begin
46      if(in == 1) state <= M1;
47      else if(in == 0) state <= M5;
48    end
49
50    M5:
51    begin
52      if(in == 1) state <= M1;
53      else if(in == 0) state <= M3;
54    end
55    endcase
56    end
57  endmodule
```

(3.4)



(3.5)

```
1 module pattern_detector (
2   input wire in,
3   input wire clk,
4   output reg out);
5
6   reg [2:0] state = 0;
7   localparam M0 = 0;
8   localparam M1 = 1;
9   localparam M2 = 2;
10  localparam M3 = 3;
11  localparam M4 = 4;
12
13
14  always @ (posedge clk)
```

```
15   begin
16
17   case(state)
18   M0:
19   begin
20     out <= 1'b0;
21     if(in == 1) state <= M1;
22     else if(in == 0) state <= M0;
23   end
24
25   M1:
26   begin
27     out <= 1'b0;
28     if(in == 1) state <= M1;
29     else if(in == 0) state <= M2;
30   end
31
32   M2:
33   begin
34     out <= 1'b0;
35     if(in == 1) state <= M1;
36     else if(in == 0) state <= M3;
37   end
38
39   M3:
40   begin
41     out <= 1'b0;
42     if(in == 1) state <= M4;
43     else if(in == 0) state <= M0;
44   end
45
46   M4:
47   begin
48     if(in == 1) state <= M1;
49     else if(in == 0)
50     begin
51        state <= M2;
52        out <= 1'b1;
53     end
54   end
55   endcase
56   end
57 endmodule
```