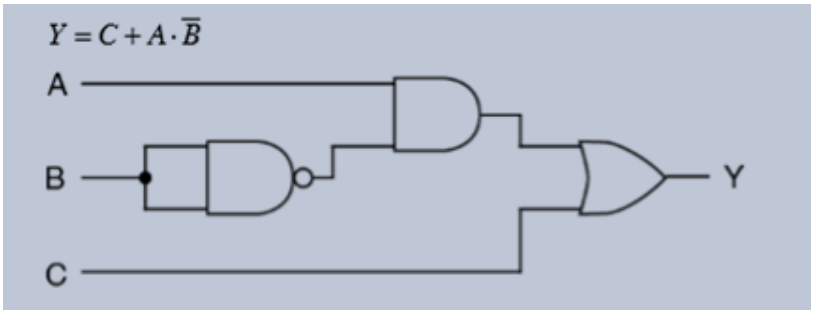
Question 1 [5 marks] Given the truth table below, draw the minimum cost logic circuit where you can only use two-input NAND gates, two-input AND gates, and two-input OR gates. The inputs are only A, B and C while the output is Y.

Inputs			Output
A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1



Question 2 [5 marks] For the Karnaugh map given below, determine the minimum cost Sum-of-Products (SOP) expression. Inverted signals have a cost.

x_3 x_4 \ x_1 x_2	0 0	0 1	1 1	1 0
0 0	1	0	0	0
0 1	1	1	0	0
1 1	0	1	1	0
1 0	0	0	0	0

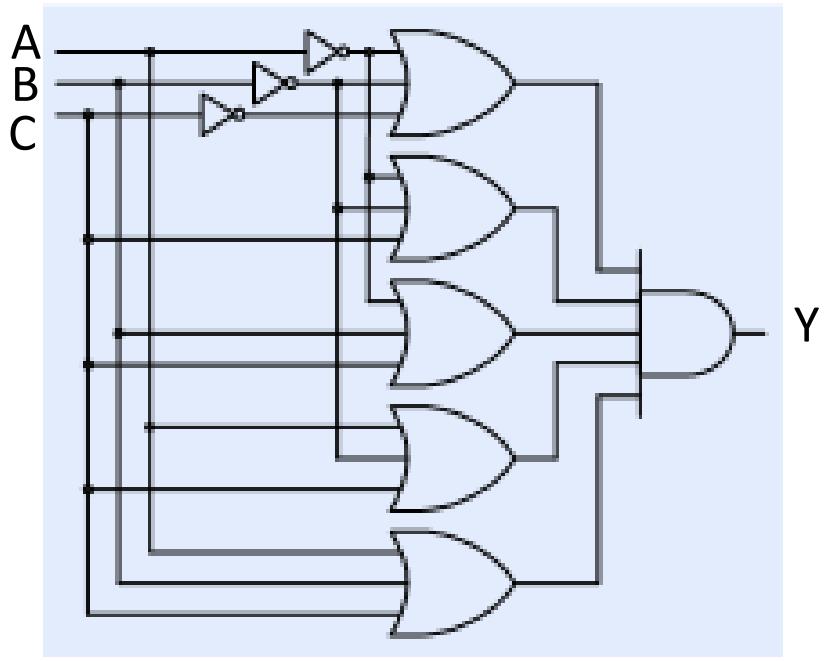
$f = \bar{x}_1\bar{x}_2\bar{x}_3 + x_2x_3x_4 + \bar{x}_1x_2x_4$

Question 3 [5 marks] Determine the minimum cost **<u>Product of Sums</u> (POS)** expression using the Karnaugh Map below. Inverted signals have a cost and d is don't-care.

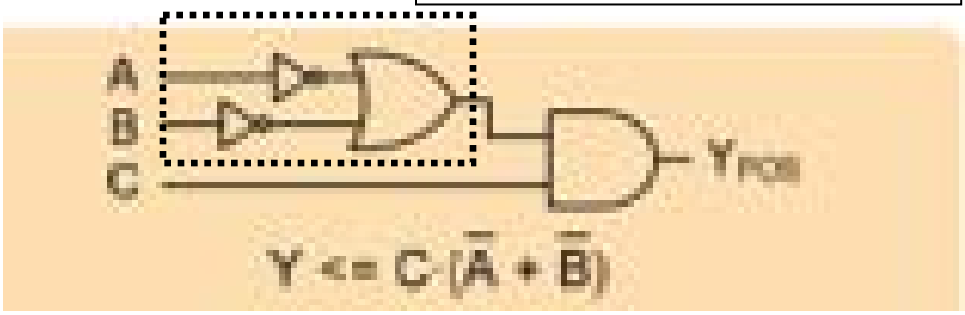
x_3 x_4 \ x_1 x_2	0 0	0 1	1 1	1 0
0 0	1	1	d	0
0 1	d	d	0	d
1 1	0	1	1	0
1 0	d	1	0	0

$f = (\overline{x_3} + x_2)(x_3 + \overline{x_1})(x_4 + \overline{x_1})$

Question 4 [10 marks] Replace the logic circuit given below with its minimum cost logic circuit. Inverters have a cost.



NAND gate can be used as well

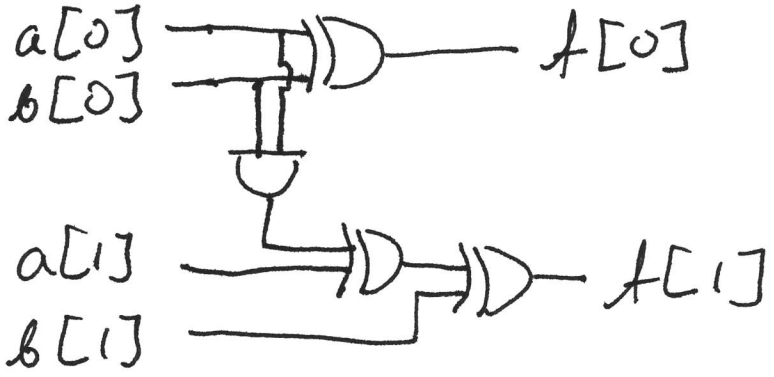


Additional space on the next page

Question 5 [15 marks total] Using basic gates (AND, OR, NOT, NAND, NOR, XOR, XNOR) and flip-flops and latches, draw the circuit for each of the following portions of Verilog code.

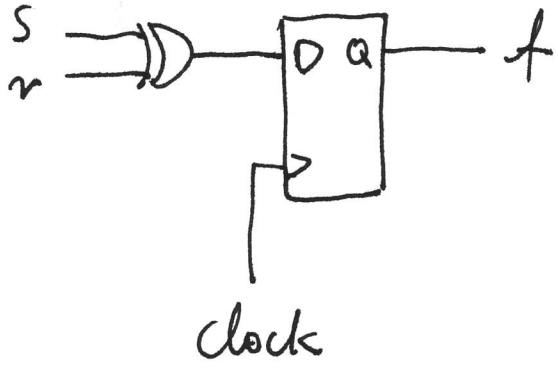
a) (2 marks)

```
module test (a, b, f);
  input [1:0] a;
  input [1:0] b;
  output [1:0] f;
  assign f = a+b;
endmodule
```



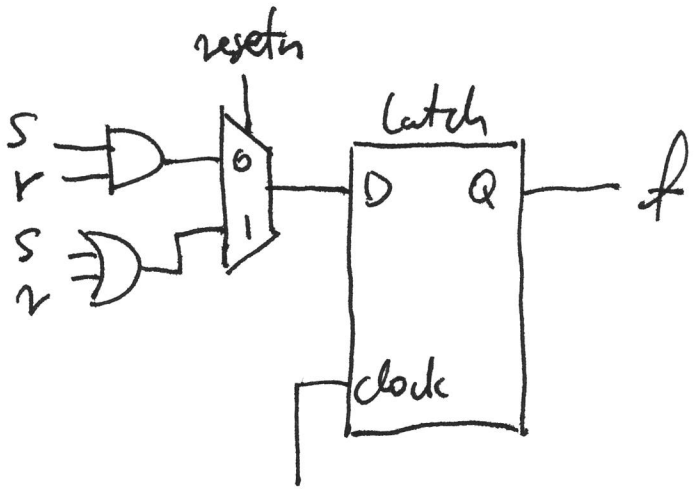
b) (2 marks)

```
module test2 (clock, s, r, f);
  input clock, s, r;
  output reg f;

  always@(posedge clock)
  begin
    f <= s^r;
  end
endmodule
```



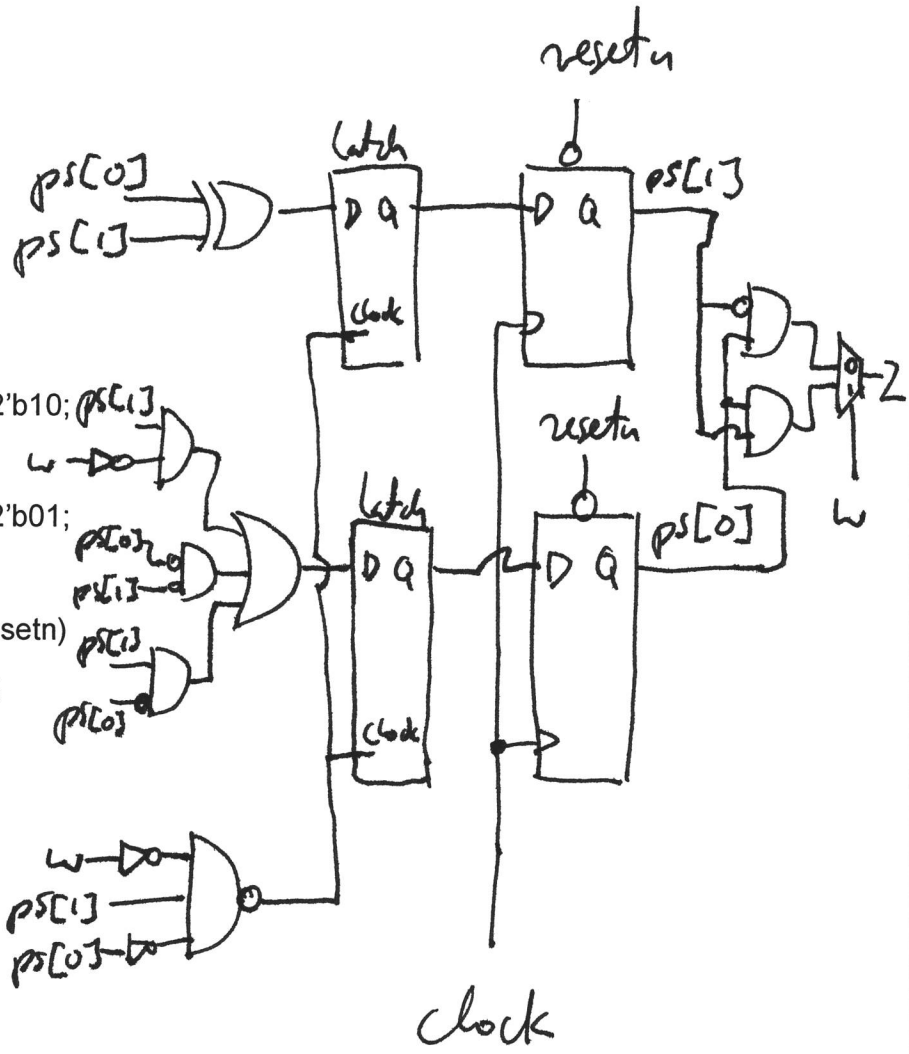
c) (4 marks)

```
module test3 (clock, s, r, resetn, f);
  input clock, s, r, resetn;
  output reg f;

  always@(*)
  begin
    if (clock == 1'b0)
      if (~resetn)
        f = s & r;
      else
        f = s | r;
  end
endmodule
```



d) (7 marks)

```
module test4 (clock, w, z, resetn);
  input clock, w, resetn;
  output reg z;

  reg [1:0] ps;
  reg [1:0] ns;

  always@(*)
   case(ps)
      2'b00: ns = 2'b01;
      2'b01: if (w) ns = 2'b10; else ns = 2'b10;
      2'b10: if (w) ns = 2'b11;
      2'b11: if (w) ns = 2'b00; else ns = 2'b01;
    endcase
  always@(posedge clock or negedge resetn)
   if (~resetn) ps <= 2'b00; else ps <=ns;

 always@(*)
   if (w) z = (ps == 2'b11);
   else z = (ps == 2'b01);

endmodule
```



Additional space on next page

Question 6 [10 marks total] Suppose you have a system which in each clock period produces a signal 'w' that is set to 0 or 1. The system produces useful output in batches. A batch begins with a pair of 1s (i.e. **11**) on the output 'w'. After the pair of 1s, the system sends a 3-bit message, which is only valid if the message represents an odd 3-bit number. An example of an arbitrary batch is: **11**011. An example of an arbitrary pair of batches is: **11**011**11**010.

Design a Finite State Machine (FSM) with minimal number of states that takes 'w' as input and produces two outputs; 'r' and 'v'. Output 'r' is asserted for one clock period once a complete batch has been detected and output 'v' is asserted at the same time if the message is valid.

Answer the following questions:

a) (2 marks) Write a state table to describe your state machine.
b) (2 marks) Create a state-assigned table for your FSM, using as few bits as possible for your state encoding.
c) (4 marks) Show logic expressions for your next-state logic and your outputs.

a)

PS	NS w=0	NS w=1	r v
A	A	B	0 0
B	A	C	0 0
C	D	D	0 0
D	E	E	0 0
E	F	G	0 0
F	A	B	1 0
G	A	B	1 1

Additional space on the next page

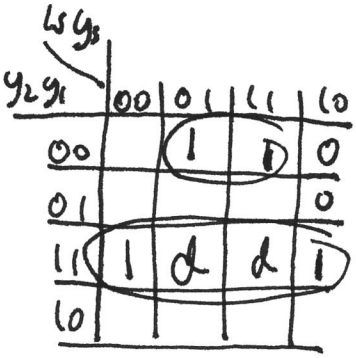
Additional space for question 6

b)

PS $(y_3 y_2 y_1)$	NS $(Y_3 Y_2 Y_1)$ $w=0$	$w=1$	z	v
000	000	001	0	0
001	000	010	0	0
010	011	011	0	0
011	100	100	0	0
100	101	110	0	0
101	000	001	1	0
110	000	001	1	1

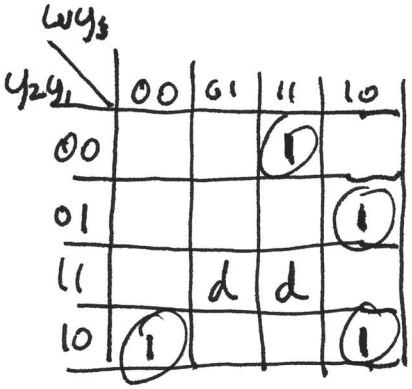
c)

Y_3



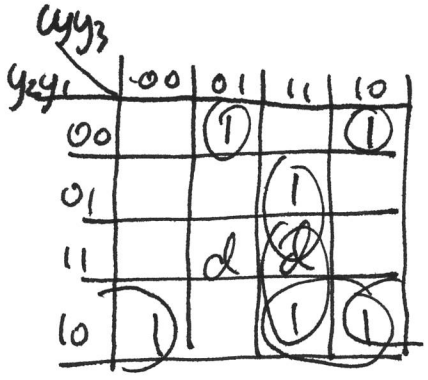
$Y_3 = \bar{y}_2 \bar{y}_1 y_3 + y_2 y_1$

Y_2



$Y_2 = w y_3 \bar{y}_2 \bar{y}_1 + w \bar{y}_3 \bar{y}_2 y_1 + \bar{w} y_3 y_2 \bar{y}_1 + w \bar{y}_3 y_2 \bar{y}_1$

Y_1



$Y_1 = \bar{w} y_3 \bar{y}_2 \bar{y}_1 + w \bar{y}_3 \bar{y}_2 \bar{y}_1 + w y_3 y_1 + w y_3 y_2 + \bar{y}_3 y_2 \bar{y}_1$

$z = y_3 y_2 + y_3 y_1$

$v = y_3 y_2$