4- $f(a,b,c,d) = \sum_m (7,3,4,6,8,9,11,12,13)\ d(7,15)$ SOP minimization:

$f = a\bar{c} + \bar{a}c + \bar{a}b\bar{d} + cd$

or

$f = a\bar{c} + \bar{a}c + \bar{a}b\bar{d} + ad$

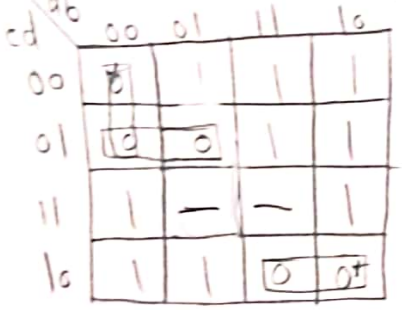
or

$f = a\bar{c} + \bar{a}c + b\bar{c}\bar{d} + cd$

or

$f = a\bar{c} + \bar{a}c + b\bar{c}\bar{d} + ad$

5- $f(a,b,c,d) = \sum_m (2,3,4,6,8,9,11,12,13)\ d(7,15)$ Pos minimization:



$f = (a+b+c).(\bar{a}+\bar{c}+d).(a+c+\bar{d})$

6- To use the Quine McCluskey Method I draws the following table:

	0 cubes	1 cubes	2 cubes	3 cubes
0011	0100 ✓	01X0	0X1X	
0010	0010 ✓	X100	0X1Y	
0100	1000 ✓	0010 ✓	1X0X	
0110	0011 ✓	0X10 ✓	1X0Y	
1100	0110 ✓	100X ✓	XX11	
1101	1100 ✓	1X00 ✓	XX11	
1000	1001 ✓	X011 ✓	1XX1	
1001	1011 ✓	0X11 ✓	1XX1	
1011	0111 ✓	011X ✓		
0111	1111 ✓	110X ✓		
1111		1X01 ✓		
		10X1 ✓		
		11X1 ✓		
		1X11 ✓		
		X111 ✓		

(ellipse: PI)

N/A

Now we draw the following block

	Cell / DI	0011	0010	0100	0110	1100	1101	1000	1001	1111
	01X0			✓	✓					
	X100			✓		✓				
EPI→	0X1X	✓	✓		✓					
EPI	1X0X					✓	✓	✓	✓	
	X X11	✓								✓
	1XX1						✓		✓	✓
	f	✓	✓	△	✓	✓	✓	✓	✓	○

we must choose the two EPIs Some always have 0X1X and 1X0X, now we must fill the cells indicated with △ and ○, for each we have two choices, from Combinatorial analysis we deduce that we have 4 different ways to represent this function which are as follows:

$$f = a\bar{c} + \bar{a}c + \bar{a}b\bar{d} + cd$$

or

$$f = a\bar{c} + \bar{a}c + \bar{a}b\bar{d} + ad$$

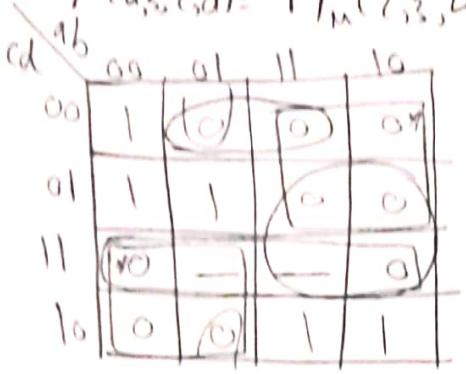
or

$$f = a\bar{c} + \bar{a}c + b\bar{c}\bar{d} + cd$$

or

$$f = a\bar{c} + \bar{a}c + b\bar{c}\bar{d} + ad$$

7- $f(a,b,c,d) = \prod_M(2,3,4,6,8,9,11,12,13)\, d(7,15)$ Pos minimizations:



$f = (\bar{a}+c).(a+\bar{c}).(a+\bar{b}+d).(\bar{c}+\bar{d})$

or

$f = (\bar{a}+c).(a+\bar{c}).(a+\bar{b}+d).(\bar{a}+\bar{d})$

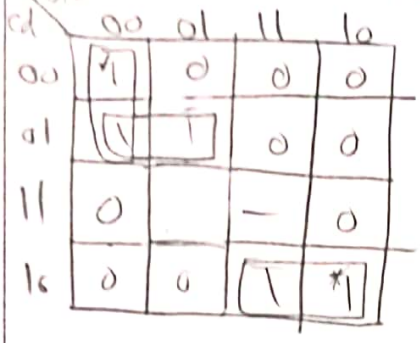
or

$f = (\bar{a}+c).(a+\bar{c}).(\bar{b}+c+d).(\bar{c}+\bar{d})$

or

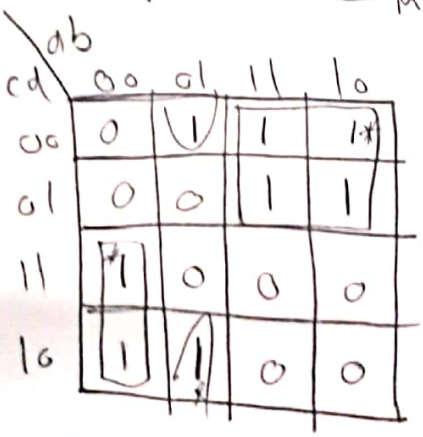
$f = (\bar{a}+c).(a+\bar{c}).(\bar{b}+c+d).(\bar{a}+\bar{d})$

8- $f(a,b,c,d) = \prod_M(2,3,4,6,8,9,11,12,13)\, d(7,15)$ SoP minimizations:



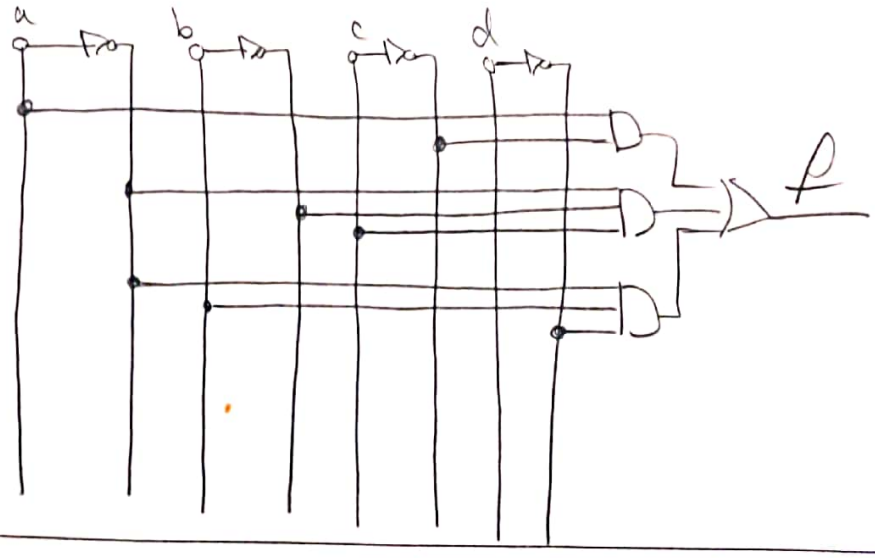
$f = \bar{a}\bar{b}\bar{c} + acd + \bar{a}\bar{c}\bar{d}$

3- All F is 1 of all, I shall minimize the function using a 4 variable k-map: $f = \sum_M(2,3,4,6,8,9,12,13)$

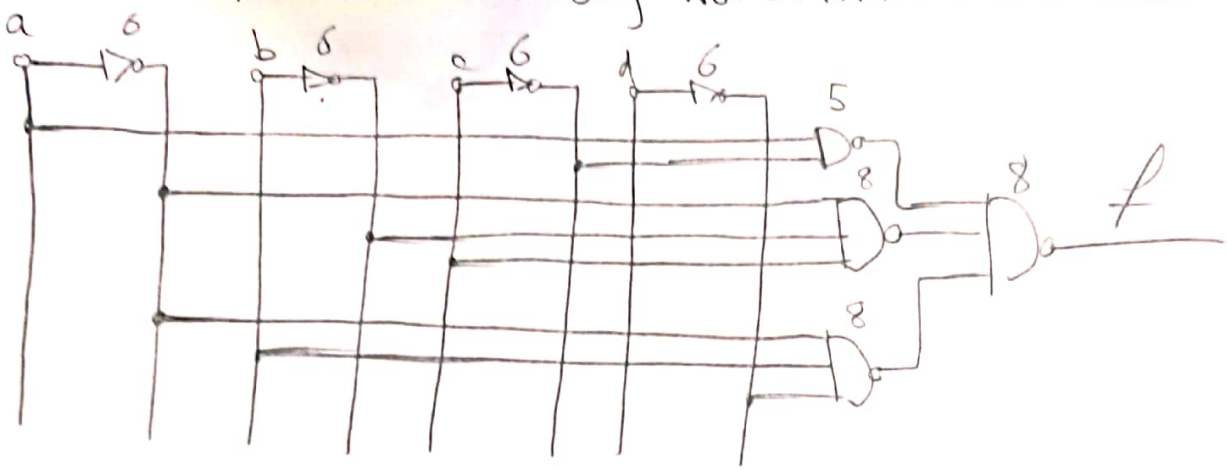


$f = a\bar{c} + \bar{a}\bar{b}c + \bar{a}b\bar{d}$

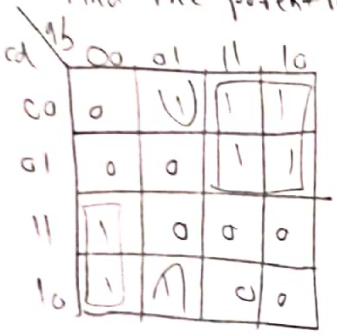
implementation (not, and, or)



Now to implement it with only NOT s YNANDs and we have:



To find the potential Hazards we take a look at the k-map



Potential hazards: (Static 1 hazards)

a) 0010 ⟷ 0110

b) 1100 ⟷ 0100

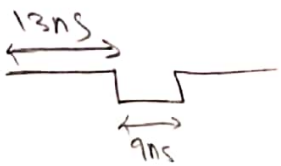
B) Now we show the logical hazards:

a) 0010 ⟷ 0110 , occurs when b changes path 1 length: 8+8=16ns

Path2 length: 22 ns 

happens 16ns after b changes with a 6ns duration

b) 1100 ⟷ 0100 , occurs when a changes path 1 length: 5+8=13ns

Path2 length: 6+8+8=22ns 

happens 13ns after a changes with a 9ns duration

C) The delay of our output gate is 8ns So we compare it with a, and b

a) 6 < 8 → logical not electrical b) 9 > 8 logical and electrical

D) As previously discussed we had two potential Hazards: a) 0010 ⟷ 0110 b) 1100 ⟷ 0100

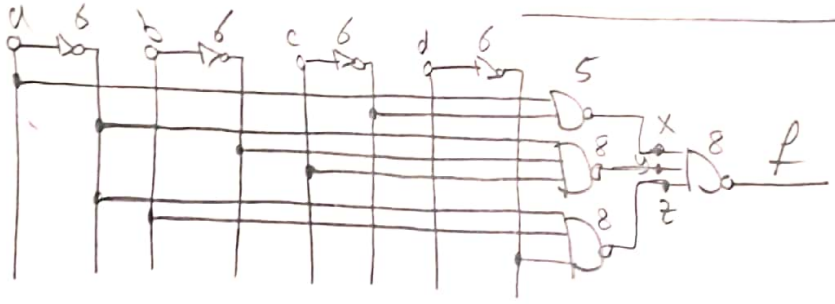
where b was already an electrical hazard, Now by driving two more gates we increase the delay hence b(1100 ⟷ 0100) shall also become only a logical hazard.

1- A) First I shall minimize and design the circuit with said gates.

ab \ cd	00	01	11	10
00	0	1	1	1
01	0	0	1	1
11	1	0	0	0
10	1	1	0	0

$f = a\bar{c} + \bar{a}\bar{b}c + \bar{a}b\bar{d}$

circuit implementation:

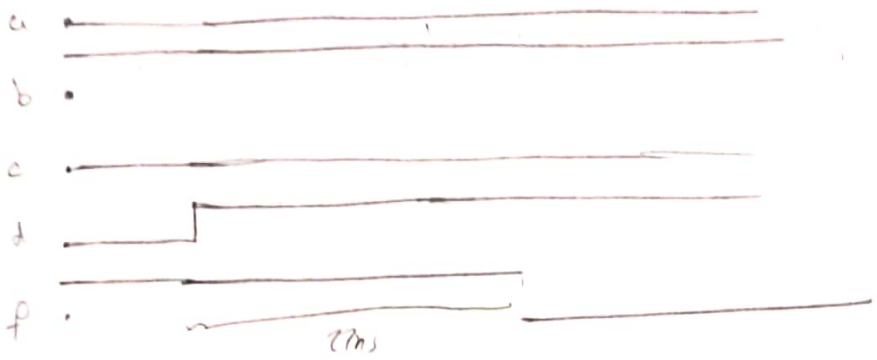


System Verilog:

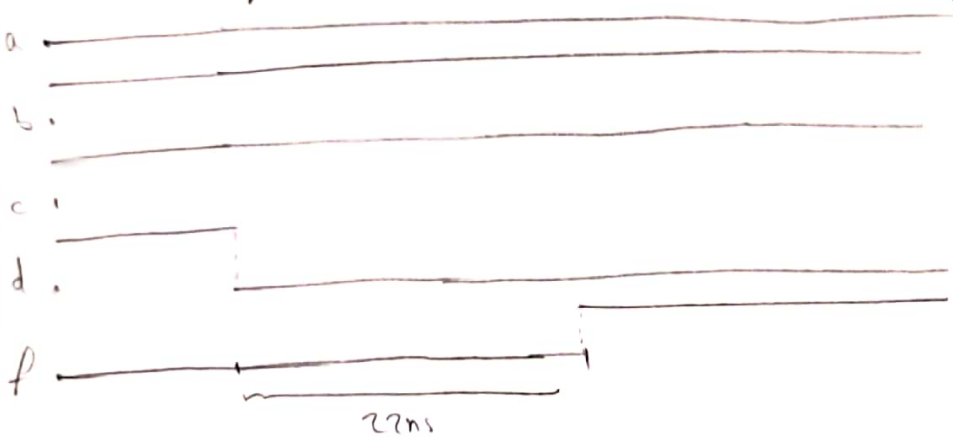
```
~ timeScale 1ns/1ns
module 01PTV(input a,b,c,d, output f);
    wire abar, bbar, cbar, dbar, x,y,z;
    not #6 (abar, a);
    not #6 (bbar, b);
    not #6 (cbar, c);
    not #6 (dbar, d);
    nand #5 (x, a, cbar);
    nand #8 (y, abar, bbar, c);
    nand #8 (z, abar, b, dbar);
    nand #8 (f, x,y,z);
endmodule
```

B) If we observe the circuit closely we see that the worst paths to the output consist of a not gate and two, three input NANDS, which shall give us: 8+8+8 = 22ns delay. Now consider the following transitions:

0100 ⟶ 0101 : This is a 1 to 0 transition, and we pass the above gates so we shall have 22ns delay (worst case to 0)



0111 ⟶ 0110 : This is a 0 to 1 transition, which also passes the discussed gates, hence we shall have a 22ns delay. (worst case to 1)



C) System Verilog using assign:

```
~timescale 1ns/1ns
module Q1ASSIGN (input a,b,c,d, output f);
  assign #(22,22) f = a&~c|~a&~b&c|~a&b&~d
endmodule
```

Soroush Mesforush Mashhad Student Number: 810198472	Digital Logic Design HW4	Due Date: 01 / 1 / 20	7/8

D) In the original circuit all the delays aren't 22 ns, for example we have a 16ns delay when transitioning from: 0000 ⟶ 0100 but when we defined this circuit using assign and its worst case delays (which was 22ns for 1 and 0 respectively.) all transitions to 1 and 0 shall have 22 ns delay which shall result in different timing behaviors between this code (part C) and the previous code (part A) because all the to 1 and to 0 delays in part A aren't its worst case scenario. To be able to find these transitions, I instantiated the circuit of part A and C with the help of Modelsim.

cd \ ab	00	01	11	10
00	0	1	1	1
01	0	0	1	1
11	1	0	0	0
10	1	1	0	0

⟶ with the help of this map, I find all the transitions with 1 input logic change, and so we have:

transition	Delay in A	Delay in C	Difference in Behavior
0000 ⟷ 1000	13	22	✓
0000 ⟷ 0010	16	22	✓
0000 ⟷ 0100	16	22	✓
0101 ⟷ 1101	13	22	✓
0101 ⟷ 0100	22	22	×
0001 ⟷ 0011	16	22	✓
0001 ⟷ 1001	13	22	✓
1111 ⟷ 1101	19	22	✓
1011 ⟷ 1001	19	22	✓
1011 ⟷ 0011	22	22	×
0111 ⟷ 0011	22	22	×
0111 ⟷ 0110	22	22	×
1010 ⟷ 0010	22	22	×
1010 ⟷ 1000	19	22	✓
1110 ⟷ 1100	19	22	✓
1110 ⟷ 1010	22	22	×

Soroush Mesforush Mashhad Student Number: 810198472	Digital Logic Design Hw 4	Due Date: 01 / 1 / 20	8/8

So in the end the transitions with different timing behaviors are:

$0000 \longleftrightarrow 1000$

$0000 \longleftrightarrow 0010$

$0000 \longleftrightarrow 0100$

$0101 \longleftrightarrow 1101$

$0001 \longleftrightarrow 0011$

$0001 \longleftrightarrow 1001$

$1111 \longleftrightarrow 1101$

$1011 \longleftrightarrow 1001$

$1010 \longleftrightarrow 1000$

$1110 \longleftrightarrow 1100$

2- It is exactly the same as question 3, which I have answered.