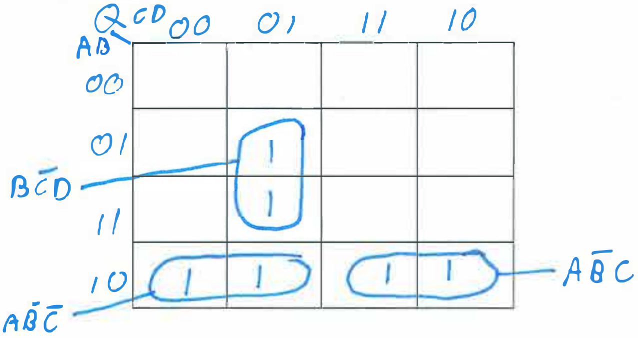
Name Answers

CE-1901-11 – Dr. Durant – Quiz 5
Winter 2016-‘17, Week 6 Quiz

1. (1 point) ***Draw*** the K-map (including implicants) that goes with the following equation: .Q = AB’C + BC’D + AB’C’



AB \ CD	00	01	11	10
00				
01		1		
11		1		
10	1	1	1	1

2. (1 point) ***Explain*** whether the K-map above was used to generate the simplest possible SOP equation (assume there were no don’t cares).

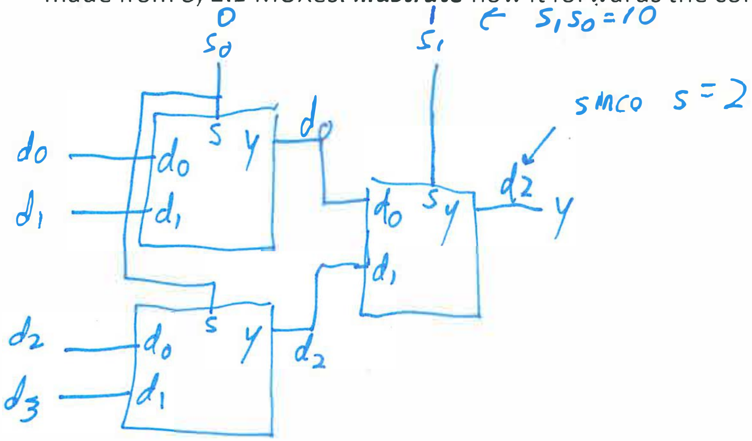
not simplest. Bottom row should be $A\bar{B}$. C cancels.

3. (3 points) MUXes

   a. (1 point) ***Write*** the logic equation for the 4:1 MUX, $y = f(s_{1..0}, d_{0..3})$. Remember that there is a product term for each of the data terms; this product term checks all values of s to make sure they match the term’s corresponding s-minterm.

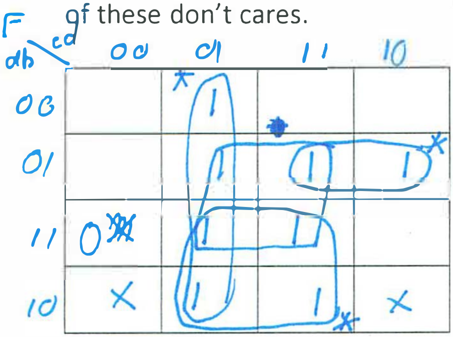
$$y = \bar{s}_1\bar{s}_0 d_0 + \bar{s}_1 s_0 d_1 + s_1\bar{s}_0 d_2 + s_1 s_0 d_3$$

   b. (2 points) Using a block diagram (without the internal gates) ***show*** how a 4:1 MUX can be made from 3, 2:1 MUXes. ***Illustrate*** how it forwards the correct d input when s= 2 = 10_2.



4. (4 points) Simplification and implementation: F(abcd) = Σ_m(1, 5, 6, 7, 9, 13, 11, 15) + d(8, 10)

a. (2 points) ***Draw*** the K-map and use it to ***derive*** a simplified SOP expression taking advantage of these don't cares.



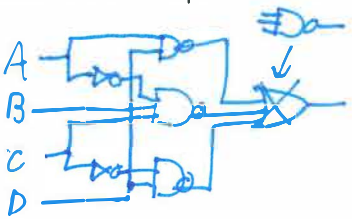
F

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

$F = \bar{C}D + \bar{A}BC + AD$

The 4th term is redundant.
The X don't help here.

b. (1 point) ***Draw*** the NOT-NAND-NAND form of the SOP circuit. Hint: Begin by placing 2 NOTs in series on each input to the OR.



c. (1 point) ***Why*** is the NOT-NAND-NAND form usually preferable to the NOT-AND-OR form?

requires fewer transistors

5. (1 point) ***Contrast*** the information you need to write a with-select statement with that needed for a direct assignment (e.g., f <= (a and not b) or c;).

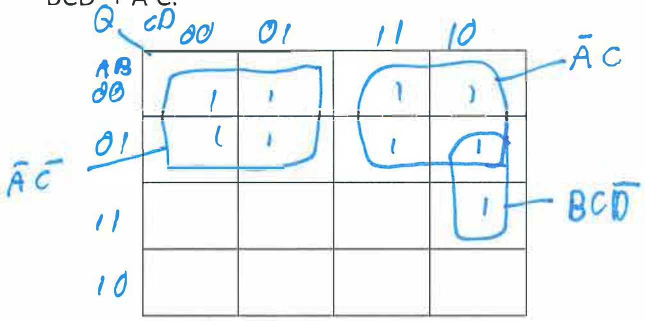
with-select: just need truth table

direct assignment: need equation

Name Answers

CE-1901-12 – Dr. Durant – Quiz 5
Winter 2016-‘17, Week 6 Quiz

1. (1 point) ***Draw*** the K-map (including implicants) that goes with the following equation: .Q = A’C’ + BCD’ + A’C.



AB \ CD	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11				1
10				

2. (1 point) ***Explain*** whether the K-map above was used to generate the simplest possible SOP equation (assume there were no don’t cares).

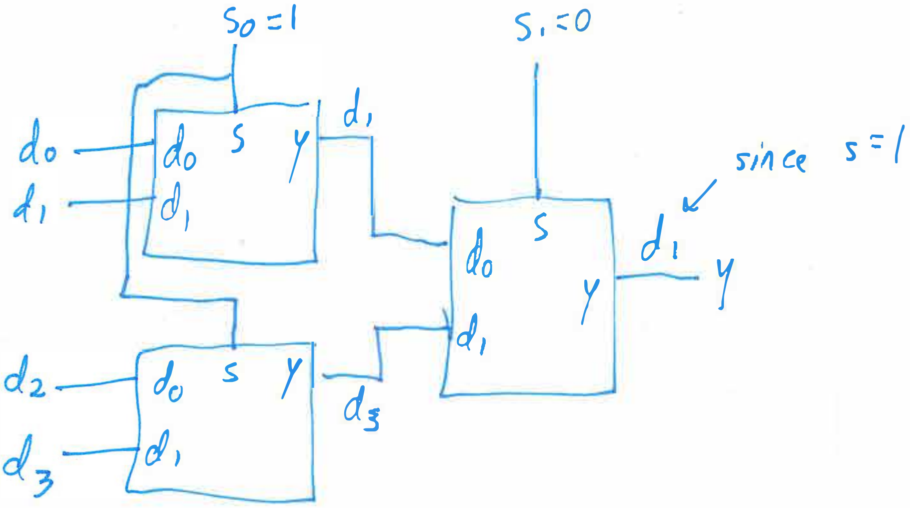
no, $A\bar{C}+\bar{A}C=\bar{A}$ (PI of 8)
↑
prime implicant

3. (3 points) MUXes

   a. (1 point) ***Write*** the logic equation for the 4:1 MUX, y = f($s_{1..0}$,$d_{0..3}$). Remember that there is product term for each of the data terms; this product term checks all values of s to make sure they match the term’s corresponding s-minterm.

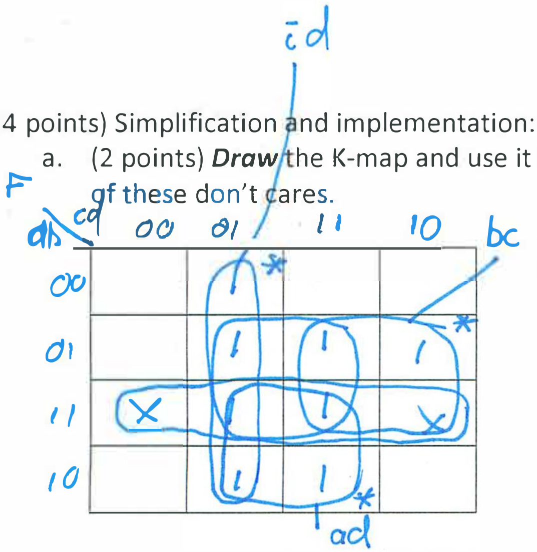
$$y = \cancel{d}\ \bar{s}_1\bar{s}_0 d_0 + \bar{s}_1 s_0 d_1 + s_1\bar{s}_0 d_2 + s_1 s_0 d_3$$

   b. (2 points) Using a block diagram (without the internal gates) ***show*** how a 4:1 MUX can be made from 3, 2:1 MUXes. ***Illustrate*** how it forwards the correct d input when s= 1 = 01_2.



4. (4 points) Simplification and implementation: $F(abcd) = \Sigma_m(1, 5, 6, 7, 9, 13, 11, 15) + d(12, 14)$

a. (2 points) ***Draw*** the K-map and use it to ***derive*** a simplified SOP expression taking advantage of these don't cares.



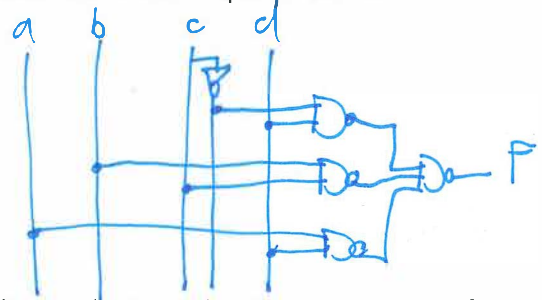
ab \ cd	00	01	11	10
00		1		
01		1	1	1
11	X	1	1	X
10		1	1	

The 3 essential PIs cover all the 1s!

$F = \bar{c}d + bc + ad$ ←

$(= (\bar{c}+a)d + bc)$

b. (1 point) ***Draw*** the NOT-NAND-NAND form of the SOP circuit. Hint: Begin by placing 2 NOTs in series on each input to the OR.



c. (1 point) ***Why*** is the NOT-NAND-NAND form usually preferable to the NOT-AND-OR form?

NANDs have 2 fewer transistors than AND/OR for a given number of inputs.

5. (1 point) ***Contrast*** the purpose of the entity section with the architecture section of a VHDL file.

entity: specify just what the inputs & outputs are

architecture: explain how to implement a circuit, eg. truth table or which gates to use & how to connect them