University of Victoria
Department of Computer Science
CSC 355 Digital Logic and Computer Design
ASSIGNMENT 2 DUE Tuesday November 8, 2016 AT BEGINNING OF CLASS

Neatness Counts!

It is expected that answers to assignments are either typed or written *extremely* neatly. In all cases, the Karnaugh Maps formats below for *must* be used, either copied and edited to add the required bits and circles or printed and written on. Also, all circuits must be drawn electronically using a circuit drawing package.

1. Simplify the following expression, using Boolean algebra, to minimal SOP form. Show (absolutely) every Boolean algebra rule used in the simplification. (Less steps will yield better marks!)

   a. $F_1 = \overline{P}\ \overline{R}\ S + \overline{P}\ \overline{Q}\ R + \overline{P}\ Q\ \overline{R} + \overline{P}\ R\ \overline{S}$

   $= \overline{P}\left(\overline{R}\ S + \overline{Q}\ R + Q\ \overline{R} + R\ \overline{S}\right)$, distributive

   $= \overline{P}\left(\overline{R}\ S + \overline{Q}\ R + Q\ \overline{R} + R\ \overline{S} + Q\bar{S}\right)$, consensus (backwards! Add term)

   $= \overline{P}\left(\overline{R}\ S + \overline{Q}\ R + Q\ \overline{R} + Q\bar{S}\right)$, consensus (normal removes term)

   $= \overline{P}\left(\overline{R}\ S + \overline{Q}\ R + Q\bar{S}\right)$, consensus (normal removes term)

   b. $F_2 = (A + D + E)\left(\bar{A} + B + \overline{C}\right)\left(\overline{A} + B + \overline{D}\right)\left(A + \overline{C}\right)$

   $= (A + D + E)\left(\overline{A} + B + \overline{D}\right)\left((\bar{A} + B)A + \overline{C}\right)$, associative, distributive

   $= (A + D + E)\left(\overline{A} + B + \overline{D}\right)\left(AB + \overline{C}\right)$, distributive, complement, ident.

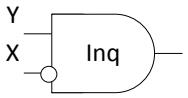
   $= (\cancel{A\bar{A}} + AB + A\overline{D} + \bar{A}D + \cancel{BD} + \cancel{D\overline{D}} + \bar{A}E + \cancel{BC} + \cancel{\overline{D}E})\left(AB + \overline{C}\right)$, distributive

   $= (AB + A\overline{D} + \bar{A}D + \bar{A}E)\left(AB + \overline{C}\right)$, complement, ident., consensus

   $= \cancel{AB\bar{C}} + A\bar{C}\overline{D} + \bar{A}\bar{C}D + \bar{A}\bar{C}E + AB + \cancel{AB\overline{D}}$, distributive, idempotent

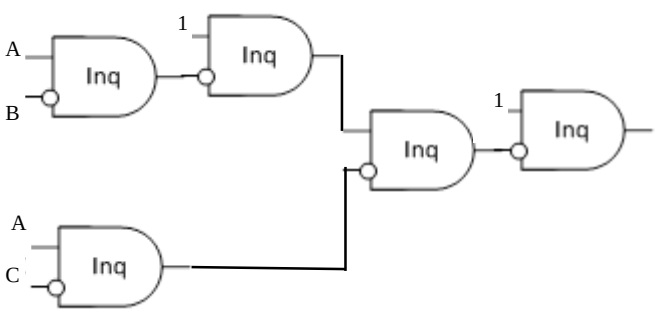
   $= A\bar{C}\overline{D} + \bar{A}\bar{C}D + \bar{A}\bar{C}E + AB$, absorption

2. An *Inquiry g*ate has two inputs (X and Y); the output is 0 except when X=0 and Y=1.



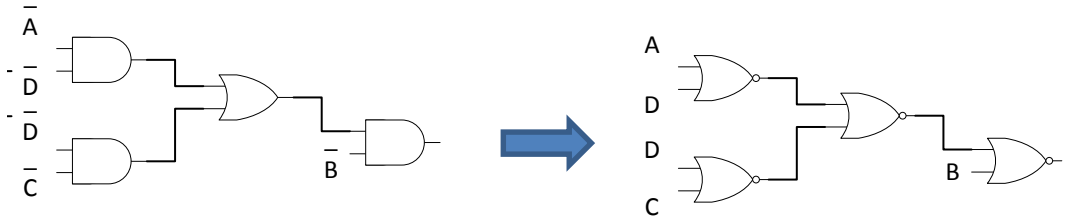
Realize the function $F = \overline{A}B + A\overline{C}$ using only inquiry gates. Note: Only A, B, C and 0 and 1 may be used as gate inputs.

Hint: 5 gates are sufficient



3. Find a minimum network to realize $F = \overline{A}\,\overline{B}\,\overline{D} + \overline{B}\,\overline{C}\,\overline{D}$ using only 2-input NOR gates. Only A, B, C and D are available and not their complements. (4 gates are sufficient).

$$F = \overline{A}\,\overline{B}\,\overline{D} + \overline{B}\,\overline{C}\,\overline{D} = \overline{B}\left(\overline{A}\,\overline{D} + \overline{C}\,\overline{D}\right)$$



4. Find a minimum network to realize Z=abe'f+c'e'f+d'e'f+gh using only 2-input NAND gates.
   Z=abe'f+c'e'f+d'e'f+gh = e'f (ab + c' + d') +gh

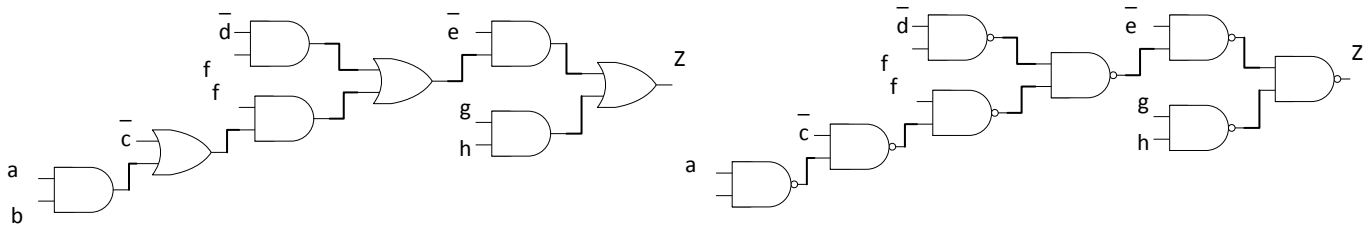
   Observe that G and H appear in only 1 term and, thus, do not need to be used in a simplification.

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

ef=01

   The above K-map shows that the expression is already minimal SOP.
   Now, to find a configuration of Z that uses only 2-input gates (and/or gates):

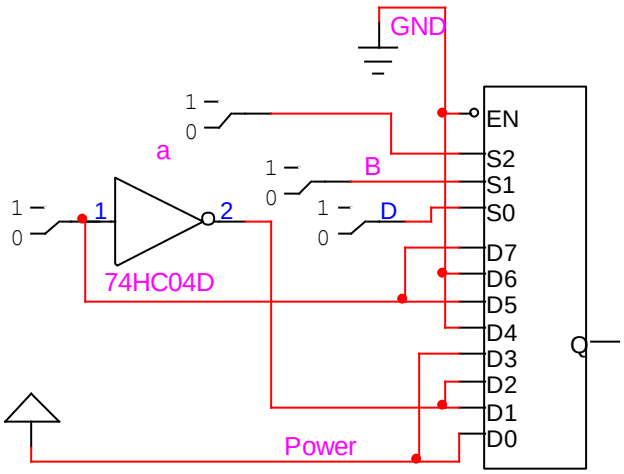
   Z=abe'f+c'e'f+d'e'f+gh = $\bar{e}\left(fab + f\bar{c} + f\bar{d}\right) + gh$ (Observe: two 3-input gates needed)
   = $\bar{e}\left(f(ab + \bar{c}) + f\bar{d}\right) + gh$ (Good, these are all 2-input gates, now draw!)



5. Consider the function specified by the Karnaugh map below:

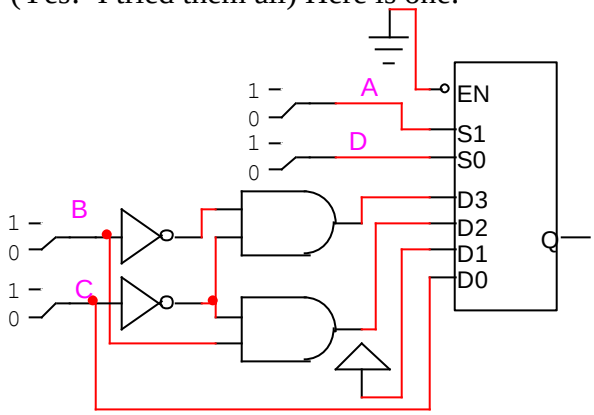
AB\CD	00	01	11	10
00	1	1		1
01	1	1	1	
11			1	
10			1	

   a. Realize the function using an 8-to-1 multiplexer with control inputs A, B and D.



b. Repeat, this time realize the function using a 4-to-1 multiplexer. Select the control inputs to minimize the number of added gates.

Trying all possible configurations, there are 2 the minimize the number of gates, using either AD or CD on the control (or select) inputs. (Yes! I tried them all) Here is one:

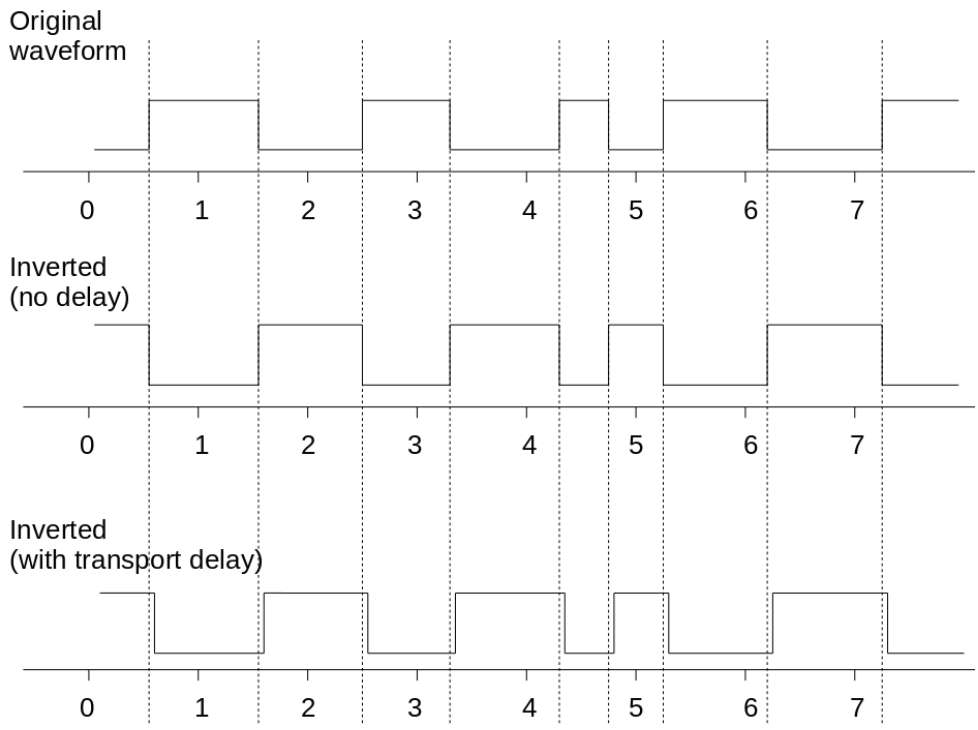


6. One of:
   a. Text (Mano, Kime, Martin, 5th edition) , page 106 #2-29; or
   b. Text (Mano, Kime, 4th edition) , page 330 #6-4.

   Propagation delay is the delay of the path with largest number of gates(from D' or C to F).
   PD = 0.073 + 0.073 + 0.048 + 0.073 = 0.267

7. One of:
   a. Text (Mano, Kime, Martin, 5th edition) , page 107 #2-30; or
   b. Text (Mano, Kime, 4th edition) , page 330 #6-5.

   As the rejection time is much smaller than two successive changes in the waveform, the waveform of b) and c) are the same.



8. One of:
    a. Text (Mano, Kime, Martin, 5th edition) , page 107 #2-31; or
    b. Text (Mano, Kime, 4th edition) , page 331 #6-6.

a) Using 0.36 and 0.20:

Longest path: Observe that the longest path has 4 NAND gates and it is not possible to have all 4 giving output '0' at the same time. In particular, if one has an output '0' the next in line will have output '1'. Thus, the longest path from an input change to the final output F will have could have two '0' outputs and two '1' outputs. If all change, the maximum propagation delay will be 2*0.36+2*0.20 = 1.12 ns. On the other hand, the shortest delay occurs when all 4 outputs on that longest path are currently 1 (ie, when ABCD=010X) and two of them (the max number) change to 0: 2*0.20 = 0.40 ns. Thus, the longest path's T_{PD}= (1.12 + 0.40)/2 = 0.76 ns.

Middle path: The middle length path has 3 NAND gates. Again, it is not logically possible to have all with output '0' at the same time. We choose the worst case, where two are 0, including F: 2*0.36 + .20 = 0.92 ns. And since it is possible to have all 3 with output '1' and 2 change to output '0', we have a shortest delay of 2*0.2 = 0.40. Thus, the average T_{PD}= (0.92 + 0.40)/2 = 0.62 ns.

Short paths: There are two shortest paths with only 2 NAND gates. The worst case is one has output 0 and changes to 1: 0.36 + 0.20 = 0.56. The best case is both have output 1 and one changest to 0: 0.20. Thus, the average T_{PD}= (0.52 + 0.20)/2 = 0.36 ns.

b) Now, using average of 0.28 for each gate:

Longest path: 4 gates * 0.28 = 1.12 ns. (Exactly equal to the worst case!)
Middle path: 3 gates * 0.28 = 0.84 ns
Short paths: 2 gates * 0.28 = 0.56. (Exactly equal to the worst case!)

9. Consider the following VHDL process. **A** and **B** are input ports of type integer, and **Tout** is an output port of type integer.

```
ARCHITECTURE test2 IS
SIGNAL T1,T2 : integer;
PROCESS(A,B)
   VARIABLE V1: integer;
BEGIN
   T1 <= A+B;
   IF (A=5) THEN
      V1 := 3;
      T2 <= V1 + T1 +2;
   ELSE
      V1 := 4;
      T2 <= T1 + B;
   END IF;
   Tout <= T2 + A;
END PROCESS;
END test2;
```

Assume that **A=0,B=0,T1=0,T2=0,V1=0** and that **A** changes from **0** to **5**. What are the final values of **T1,T2,V1,Tout** after the process is executed and it goes back to the suspended state? Justify and explain your answer. (4 marks for the correct answers and 4 marks for a logical explanation).

Observe that the process is executed sequentially but the change in the signals are planned, but do not occur until the END process statement. However, the local variable, V1 will get a new value as soon as it is executed. Thus:

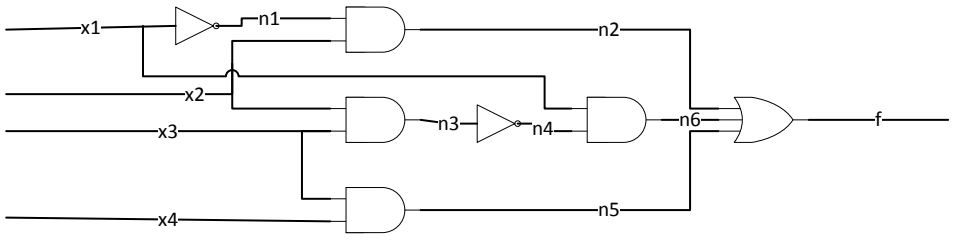
When A changes from 0 to 5 the process begins execution. The following new values of signals are planned, but do not yet occur:

- T1 will be assigned value 5
- V1 is immediately change to 3
- T2 will be assigned value 3 (the 3 from V1 and the old value of T1 which is 0)
- Tout will be assigned value 5 (the 5 from A that occurred before the process started plus the old value of T2.

When the end process occurs, the signals are changed to T1=5, T2=3, Tout=5.

-

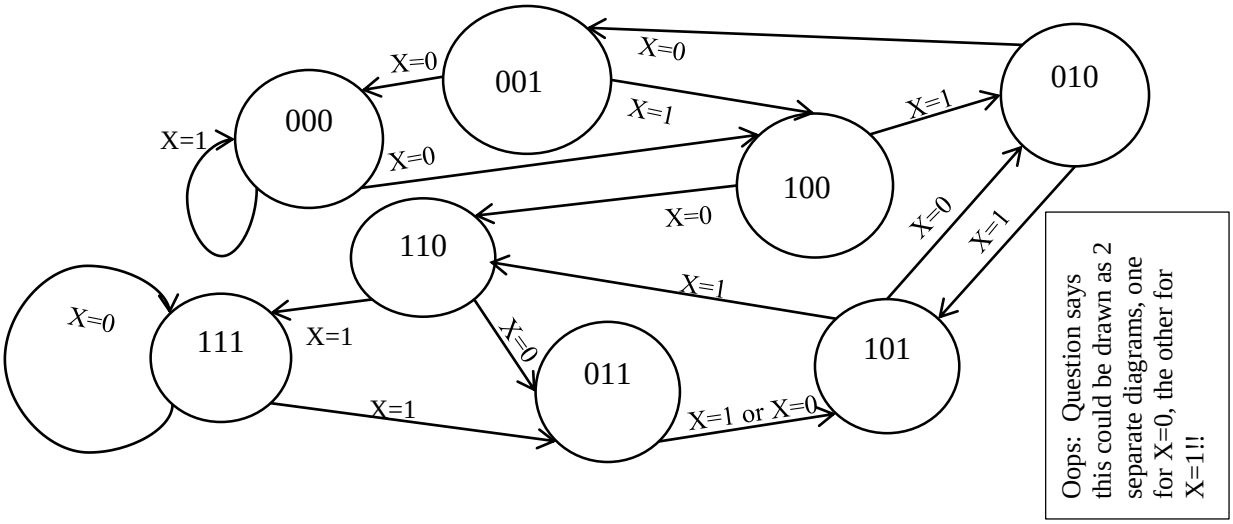
10. Text (Mano, Kime, Martin, 5th edition) , page 109 #2-34, or
    Text (Mano, Kime, Martin, 4th edition) , page 201 #4-20



11. Text (Mano, Kime, Martin, 5th edition) , page 281 #4-7, or
Text (Mano, Kime, Martin, 4th edition) , page 282 #5-7 *Recall D flip-flop has: Q+=D*

$D_A = (B\bar{C} + \bar{B}C)X + (BC + \bar{B}\,\bar{C})\bar{X}$ Thus, $A^+ = (B\bar{C} + \bar{B}C)X + (BC + \bar{B}\,\bar{C})\bar{X}$
$D_B = A$ Thus, $B^+ = A$
$D_c = B$ Thus, $C^+ = B$

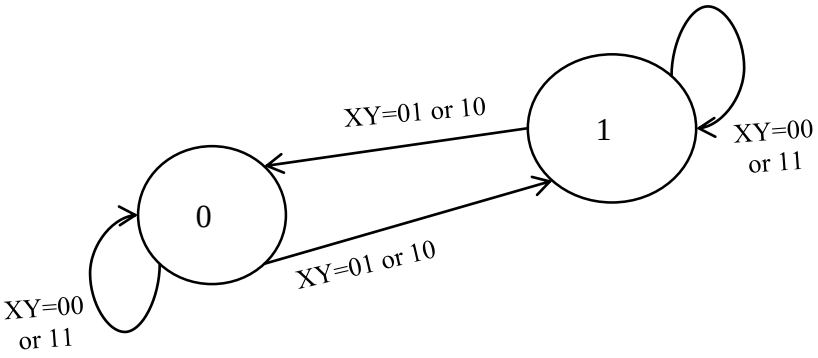
Present	Next	
ABC	X=0	X=1
	$A^+B^+C^+$	$A^+B^+C^+$
000	100	000
001	000	100
010	001	101
011	101	101
100	110	010
101	010	110
110	011	111
111	111	011



12. Text (Mano, Kime, Martin, 5th edition) , page 282 #4-8, or
Text (Mano, Kime, Martin, 4th edition) , page 282 #5-8 *Recall D flip-flop has: Q+=D*

$D_S = X \oplus Y \oplus S$, Thus $S^+ = X \oplus Y \oplus S$

Present	Next			
S	XY=00	XY=01	XY=10	XY=11
	S^+	S^+	S^+	S^+
0	0	1	1	0
1	1	0	0	1



13. Text (Mano, Kime, Martin, 5th edition) , page 282 #4-9, or
Text (Mano, Kime, Martin, 4th edition) , page 282 #5-9

Input X = 100110111110
Output Z = 010001000001

14. Text (Mano, Kime, Martin, 5th edition) , page 282 #4-10, or
Text (Mano, Kime, Martin, 4th edition) , page 283 #5-10