University of Victoria
Department of Computer Science
CSC 355 Digital Logic and Computer Design
ASSIGNMENT 2 DUE Thursday October 14, 2015 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. $F_1 = (\overline{A} + B + D)(A + C)(A + \overline{B} + D)(\overline{A} + \overline{C} + \overline{D})(\overline{A} + C)$

   F1= (A'+B+D)(A'+C)(A+B'+D)(A+C)((A'+C'+D')(A'+C) commutative & Idempotent
   F1=[A'+(B+D).C)][A+(B'+D).C][A'+(C'+D').C] distributive
   F1=(A'+BC+CD)(A+B'C+CD)(A'+CC'+D'C) distributive
   F1=(A'+BC+CD) (A'+CC'+D'C) (A+B'C+CD) commutative
   F1=[A'+(BC+CD)(CD')] (A+B'C+CD) distributive
   F1=(A'+BCCD'+CCDD') (A+B'C+CD) distributive
   F1=(A'+BCD') (A+B'C+CD) Idempotent & Complement
   F1=AA'+A'B'C+A'CD+ABCD'+BB'CCD'+BCCDD') distributive
   F1=A'B'C+A'CD+ABCD' Complement

   b. $F_2 = \overline{W}\ \overline{Y}\ Z + \overline{W}\ \overline{X}\ Y + \overline{W}\ X\ \overline{Y} + \overline{X}\ Y\ \overline{Z}$

   $= \overline{W}\ \overline{Y}\ Z + \overline{W}\ \overline{X}\ Y(Z + \overline{Z}) + \overline{W}\ X\ \overline{Y} + \overline{X}\ Y\ \overline{Z}$ complement, identity

   $= \overline{W}\ \overline{Y}\ Z + \overline{W}\ \overline{X}\ YZ + \overline{W}\ X\ \overline{Y} + \overline{W}\ \overline{X}\ Y\overline{Z} + \overline{X}\ Y\ \overline{Z}$ distributive, commutative

   $= \overline{W}\ Z(\overline{Y} + \overline{X}\ Y) + \overline{W}\ X\ \overline{Y} + \overline{X}\ Y\ \overline{Z}$ absorption, distributive

   $= \overline{W}\ Z(\overline{Y} + \overline{X}) + \overline{W}\ X\ \overline{Y} + \overline{X}\ Y\ \overline{Z}$ simplification

   $= \overline{W}(Z\overline{Y} + Z\overline{X} + X\ \overline{Y}) + \overline{X}\ Y\ \overline{Z}$ distributive, distributive

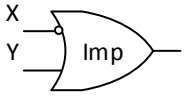
   $= \overline{W}(Z\overline{X} + X\ \overline{Y}) + \overline{X}\ Y\ \overline{Z}$ consensus

   $= \overline{W}\ \overline{X}\ Z + \overline{W}X\ \overline{Y} + \overline{X}\ Y\ \overline{Z}$ distributive

   c. $F_3 = (V + Y + Z)(\overline{V} + W + \overline{X})(\overline{V} + X + \overline{Y})(V + \overline{X})$

   F3=(VV+VX'+VY+VZ+X'Y+X'Z)(V'V'+V'X+V'Y'+V'W+V'X'+XW+Y'W+X'Y') Distributive
   F3=(V+VX'+VY+VZ+X'Y+X'Z)(V'+V'X+V'Y'+V'W+V'X'+XW+Y'W+X'Y') Idempotent
   F3=(V+X'Y+X'Z)(V'+XW+Y'W+X'Y') Absorption
   F3=VXW+VY'W+VX'Y'+V'X'Y+V'X'Z+X'Y'WZ+X'Y'Z Distributive
   F3= VXW+VY'W+VX'Y'+V'X'Y+V'X'Z+X'Y'WZ+X'Y'Z Distributive
   F3= VXW+VY'W+VX'Y'+V'X'Y+V'X'Z+X'Y'Z Absorption
   F3= V(XW+Y'W+X'Y')+X'(V'Y+V'Z+Y'Z) Distributive
   F3= V(XW+X'Y')+X'(V'Y+Y'Z) Consensus

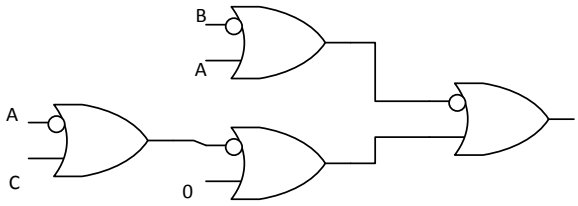
2. An *Implication* gate has two inputs (X and Y); the output is 1 except when X=1 and Y=0



   Realize the function $F = \overline{A}B + A\overline{C}$ using only implication gates.

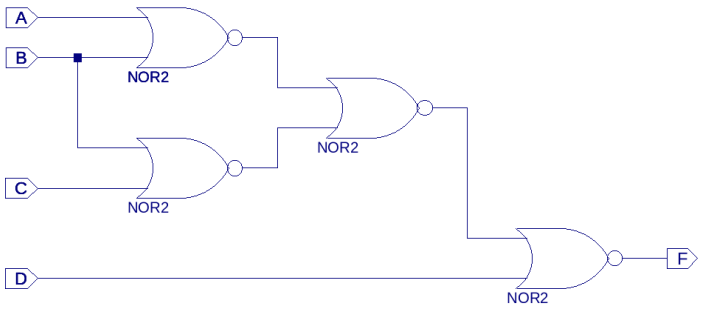
   - Note: Only A, B, C and 0 and 1 may be used as gate inputs.
   - Hint: 4 gates are sufficient.

F=A'B+AC' = ((A'B+AC')')' = ((A'.B)' . (A.C')')' = ((A+B') . (A'+C))' = (A+B')' + (A'+C)'

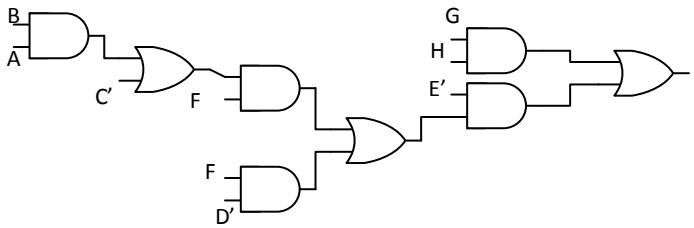


3. Find a minimum network to realize $F = \overline{A}\,\overline{B}\,\overline{D} + \overline{B}\,\overline{C}\,\overline{D}$ using only 2-input NOR gates. Only A, B, C and D are available and not their complements. (4 gates are sufficient).
F= (A+B+D)' + (C+B+D)'

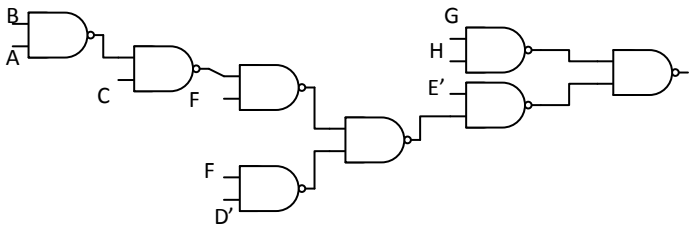
$$F = D'(A'B' + B'C') = \overline{D + (\overline{A'B' + B'C'})} = \overline{D + \left((\overline{A + B}) + (\overline{B + C})\right)}$$



4. Find a *minimum* network to realize Z=abe'f+c'e'f+d'e'f+gh using only 2-input NAND gates.
   - First try to minimize in SOP form:
     Note that G and H are only in 1 term, so can't be combined differently.
     The remaining terms all contain E'F, factoring that out, indicates clear minimal result:
     $$Z = E'F(AB + C' + D') + GH$$
   - Okay, so we need to get everything into 2-input gates:
     $$Z = E'(F(AB + C') + FD') + GH$$
   - Next make from And & Or gates:



   - Then convert all to NAND gates:

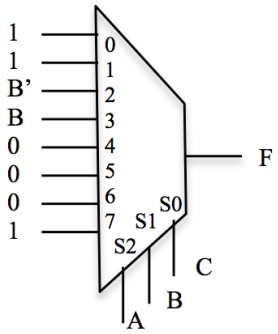


   Thus, 8 NAND gates plus 2 interters, or 10 NAND gates.

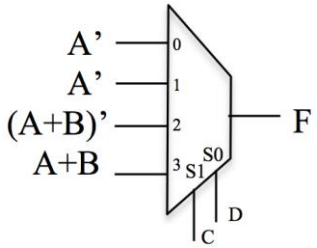
5. Consider the function specified by the Karnaugh map below:

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, C 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.



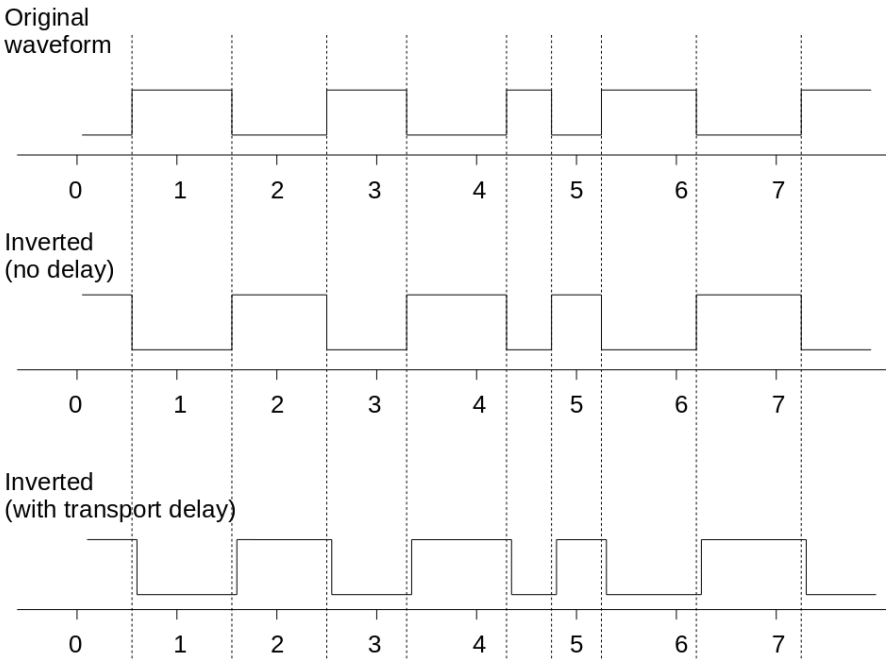
6. One of:
   a. Text (Mano, Kime, Martin, 5th edition) , page 106 #2-29; or
   b. Text (Mano, Kime, 4th edition) , page 330 #6-4.

   Propagation delay of the circuit is on 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.

a)



b) c)

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) The longest path, with 4 NAND gates, will have the longest delay. If all gates have output 0 and transition to 1, the longest possible delay would occur. However, we must first analyze to see if it is possible to have all gates in output 0:

A	B	C	D	1	2	3	F	#0's
0	0	0	0	1	0	1	0	2
0	0	0	1	1	0	1	0	2
0	0	1	0	1	0	1	0	2
0	0	1	1	0	1	1	0	1
0	1	0	0	1	1	1	1	0
0	1	0	1	1	1	1	1	0
0	1	1	0	1	1	1	0	1
0	1	1	1	0	1	1	0	1
1	0	0	0	1	0	1	0	2
1	0	0	1	1	0	1	0	2
1	0	1	0	1	0	1	0	1
1	0	1	1	0	1	0	1	2
1	1	0	0	1	1	0	1	1
1	1	0	1	1	1	0	1	1
1	1	1	0	1	1	0	1	1
1	1	1	1	0	1	0	1	2

By brute force (!!) the longest delay will occur when 2 gates have output 0 and transition to 1, while the other 2 on the longest path have output 1 and transition to 0:

$$T_{pd} = 2 \times T_{PHL} + 2 \times T_{PLH} = 2 \times 0.20 + 2 \times 0.36 = 1.12$$

b) We calculate tpd for NAND gates as follows:

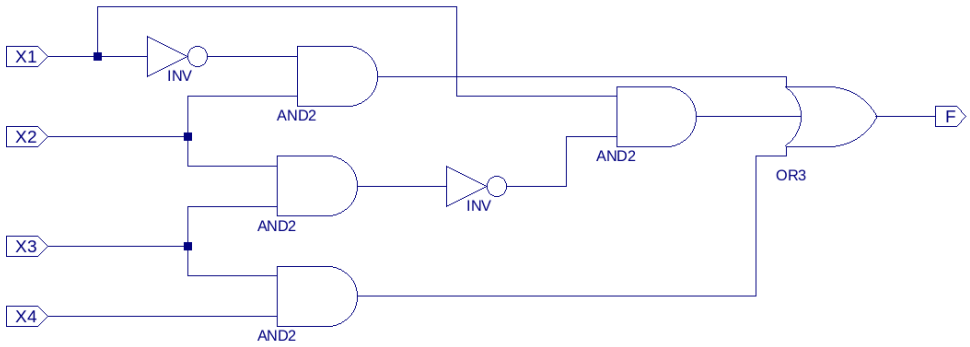
$$T_{pd} = \frac{T_{PHL} + T_{PLH}}{2} = \frac{0.20 + 0.36}{2} = 0.28$$

If an input appears more than once at input (e.g. B and B') we consider longer delay as its input to output delay.

Input to on longest path (4 gates) = 4 x 0.28 = 1.12

c) they are the same.

9. One of:
   a. Text (Mano, Kime, Martin, 5th edition) , page 109 #2-34
   b. Text (Mano, Kime, 4th edition) , page 201 #4-20



Some Karnaugh maps for your editing pleasure

A\BC	00	01	11	10
0				
1				

AB\CD	00	01	11	10
00				
01				
11				
10				

Change the variables if your expressions require different variable names!