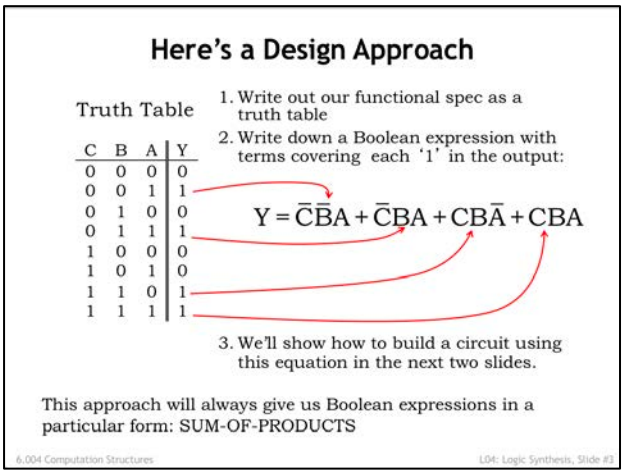
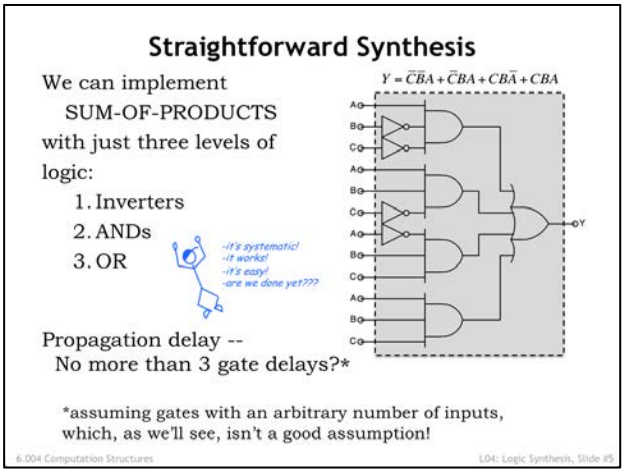
Computation Structures

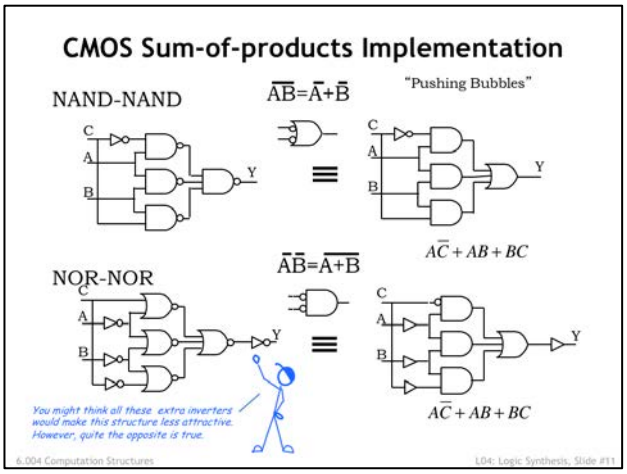
Combinational Logic Worksheet

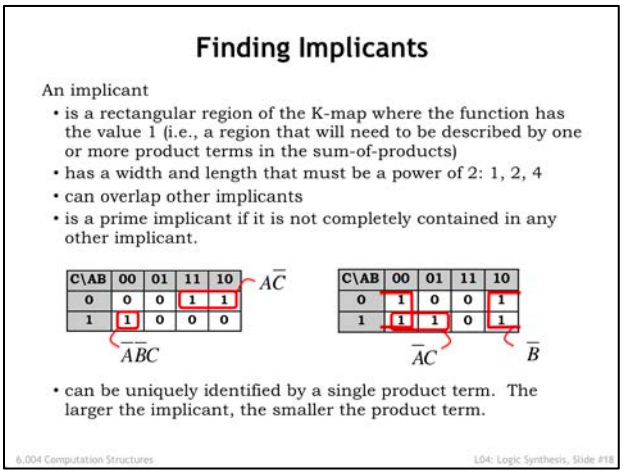
Concept Inventory:

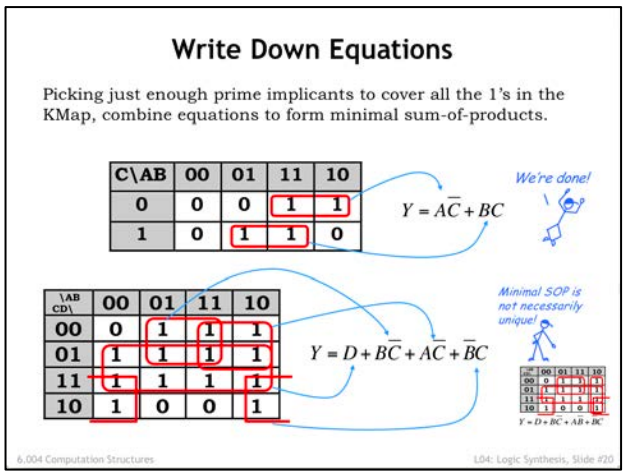
- Truth tables ↔ sum-of-products equations
- implementation using NOT/AND/OR
- Demorgan's Law, implementation using NAND/NOR
- Simplification, truth tables w/ don't cares
- Karnaugh maps
- Implementation using MUXes and ROMs

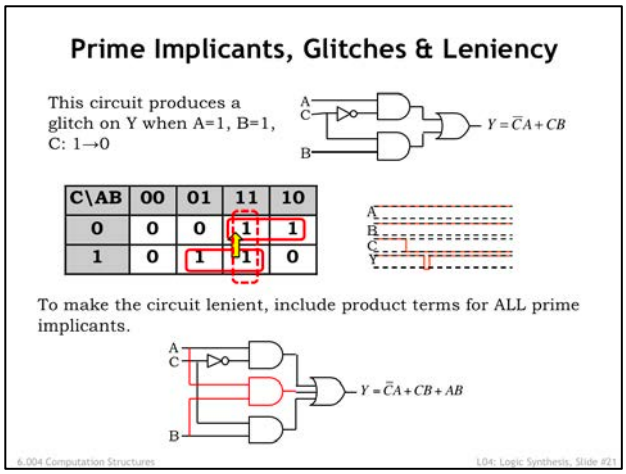








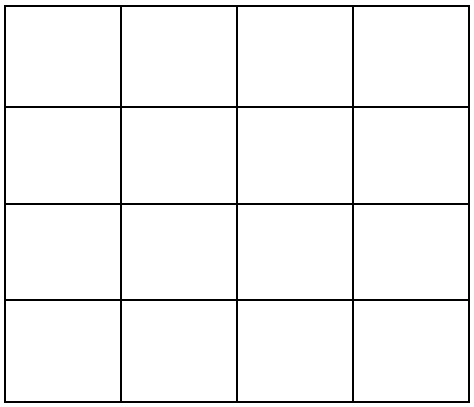




Problem 1.

Given a function F defined by the truth table to the right, provide a minimal sum-of-products expression for F. Hint: Use a Karnaugh Map.

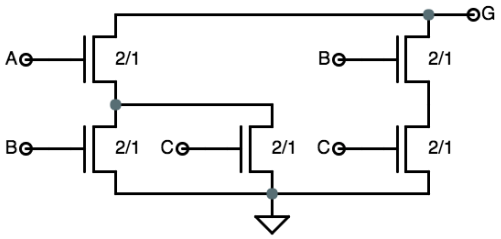
A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0



Minimal Sum-of-products Expression for F: _____

Problem 2.

(A)A correctly-formed CMOS gate implementing G(A,B,C) uses the pulldown circuit shown on the right. Please give a *minimal sum-of-products expression* for G(A,B,C). Hint: use a Karnaugh map!



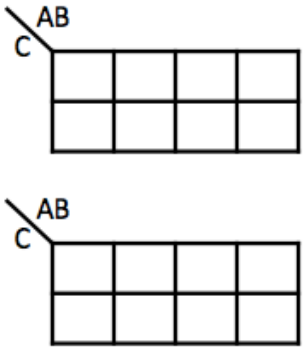
Minimal sum-of-products expression for G(A,B,C): _____

(B) Is the function G(A,B,C) from part (B) universal? In other words, can we implement any Boolean function using combinational circuits built only from G gates and the Boolean constants 0 and 1?

Problem 3.

Consider the following truth table which defines two functions F and G of three input variables (A, B, and C).

A	B	C	F	G
0	0	0	1	1
0	0	1	1	1
0	1	0	0	1
0	1	1	1	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	1
1	1	1	1	0



Give the **minimal sum of products** (minimal SOP) logic equation for each of the two functions. Then determine if the minimum sum of products expression would result in a **lenient** implementation of the function. If it does, then enter "**SAME**" for the lenient SOP expression. **If not**, specify what sum of products expression would result in a lenient implementation. Hint: Use Karnaugh maps above to determine the minimal sum of products.

Minimal sum of products F(A,B,C) = _____

Does minimal SOP for F result in a lenient circuit (circle one)? Yes No

If "No", give lenient SOP expression for F(A,B,C) = _____

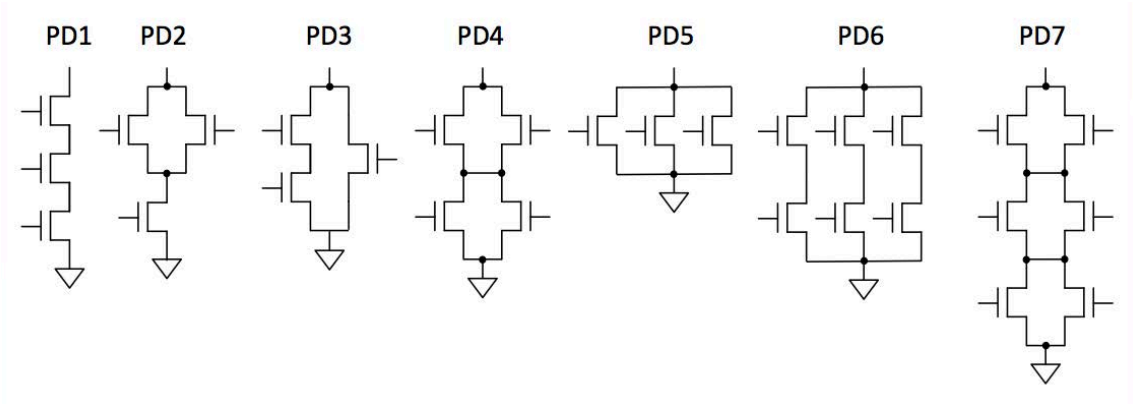
Minimal sum of products G(A,B,C) = _____

Does minimal SOP for G result in a lenient circuit (circle one)? Yes No

If "No", give lenient SOP expression for G(A,B,C) = _____

Problem 4.

You are trying to select pulldowns for several 3- and 4-input CMOS gate designs. The Pulldowns-R-Us website offers seven different pulldowns, given names PD1 through PD7, diagrammed below:



The web site explains that the customer can choose which inputs or constants (GND, VDD) are connected to each NFET, allowing their pulldowns to be used in various ways to build gates with various numbers of inputs. Since Pulldowns-R-Us charges by transistor, you are interested in selecting pulldowns using the minimum number of transistors for each of the 3-input gates you are designing.

For each of the following 3- and 4-input Boolean functions, choose the appropriate pulldown design, i.e., the one which, properly connected, implements that gate's pulldown using the *minimum number* of transistors. This may require applying Demorgan's Laws or minimizing the logic equation first. If none of the above pulldowns meets this goal, write "NONE".

(A) $F(A,B,C) = \overline{A + (B \cdot C)}$ **Choice or NONE: _____**

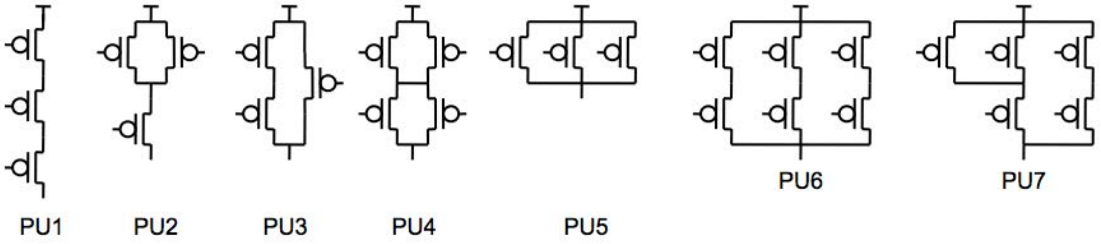
(B) $F(A,B,C) = A + \overline{B \cdot C}$ **Choice or NONE: _____**

(C) $F(A,B,C) = (\overline{A} \cdot \overline{B}) + \overline{C}$ **Choice or NONE: _____**

(D) $F(A,B,C,D) = \overline{A + C \cdot (B + D)}$ **Choice or NONE: _____**

Problem 5.

You are trying to select pullups for several 3-input CMOS gate designs. The Pullups Galore web site offers seven different pullups, given names PU1 through PU7, diagrammed below:



The web site explains that the customer can choose which inputs are connected to each PFET, allowing their pullups to be used in various ways to build gates with various numbers of inputs. Since Pullups Galore charges by transistor, you are interested in selecting pullups using the minimum number of transistors for each of the 3-input gates you are designing.

For each of the following 3-input Boolean functions, choose the appropriate pullup design, i.e., the one which, properly connected, implements that gate's pullup using the *minimum number* of transistors. This may require minimizing the logic equation first. If none of the above pullups meets this goal, write "NONE".

(A) $F(A,B,C) = \overline{A} + \overline{B} + \overline{C}$ **Choice or NONE: _____**

(B) $F(A,B,C) = \overline{A} + \overline{B \cdot C}$ **Choice or NONE: _____**

(C) $F(A,B,C) = \overline{A + B \cdot C}$ **Choice or NONE: _____**

(D) $F(A,B,C) = A + \overline{B \cdot C}$ **Choice or NONE: _____**

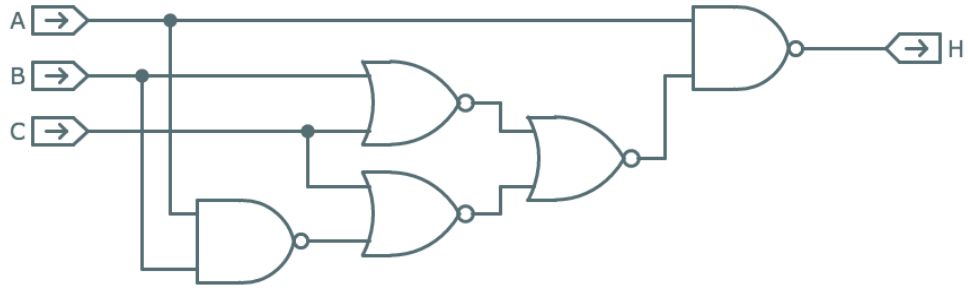
(E) $F(A,B,C) = \overline{(A+B)} + \overline{(B+C)} + \overline{(A+C)}$ **Choice or NONE: _____**

(F) $F(A,B,C) = \overline{(A+C) \cdot B}$ **Choice or NONE: _____**

Problem 6.

Consider the Boolean function $H(A,B,C) = \overline{A} + \overline{B} \cdot \overline{C} + A \cdot B \cdot \overline{C}$. Its truth table is shown to the right and a possible implementation is shown in the schematic below.

A	B	C	H
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0



(A) Give a minimal sum-of-products expression for H. A couple of scratch 3-input Karnaugh map templates are provided for your convenience.

minimal sum-of-products expression for H: _____

	00	01	11	10
0				
1				

	00	01	11	10
0				
1				

(B) What is the largest number of product terms possible in a minimal sum-of-products expression for a 3-input, 1-output Boolean function?

Largest number of product terms possible: _____

	00	01	11	10
0				
1				

Problem 7.

A minority gate has three inputs (call them A, B, C) and one output (call it Y). The output will be 0 if two or more of the inputs are 1, and 1 if two or more of the inputs are 0.

(A) Give a *minimal* **sum-of-products** Boolean expression for the minority gates output Y, in terms of its three inputs A, B, and C.

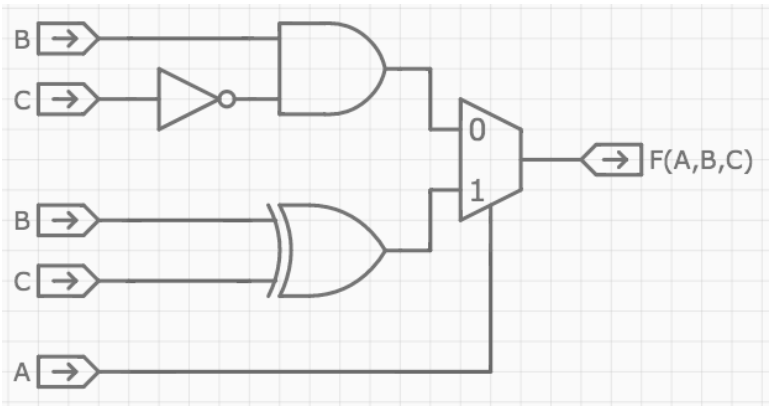
Minimal SOP expression: Y = _____

(B) Is a minority gate *universal*, in the sense that using only minority gates (along with constants 0 and 1) its possible to implement arbitrary combinational logic functions?

Universal? Circle one: YES can't tell NO

Problem 8.

A 6.004 intern at Intel has designed the combinational circuit shown below. His boss can't figure out what it does and has asked for your help.



A	*B*	*C*	*F(A,B,C)*
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

(A) Please fill in the truth table for F(A,B,C) above.

Fill in truth table above

(B) Express F(A,B,C) in minimal sum-of-products form. Hint: use a Karnaugh map!

minimal sum-of-products expression for F(A,B,C) = _____

(C) The boss isn't quite sure what it means but he knows his engineers are always impressed if he asks "is the circuit universal?" Is it? Circle YES or NO.

F(A,B,C) universal? YES … NO

Problem 9.

The full subtractor (FS) implements a one-column binary subtraction of two bits (X and Y) producing their difference (D), accepting a borrow-in (B_{IN}) from the previous column and producing a borrow-out (B_{OUT}) for the next column. Numerically FS computes $X-Y-B_{IN}$ and encodes the possible answers (1, 0, −1, −2) using D and B_{OUT} as shown in the truth table to the right.

X	Y	B_{IN}	D	B_{OUT}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

(A) (2 Points) Give a sum-of-products expression for B_{OUT}.

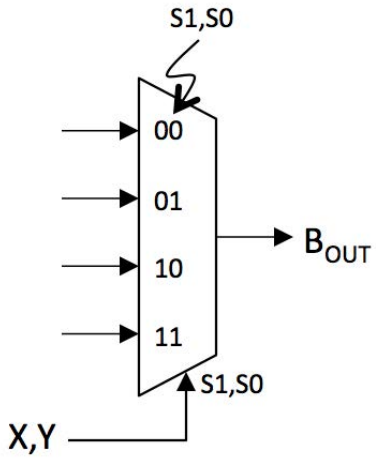
Sum of products expression: B_{OUT} = _____

(B) (1 Point) Is the FS(X,Y,B_{IN}) circuit universal in the sense that 2-input NOR and 2-input NAND are universal? In other words, using only acyclic networks of FS circuits (perhaps with one or more of their inputs tied to "0" or "1"), can one implement any combinational logic function?

FS Universal: … YES … NO …

(C) (2 Points) You're trying to build an implementation for the B_{OUT} part of the FS circuit (see truth table above) but discover that the NITGFOC supply room only has 4-to-1 multiplexors in stock. In desperation, you call up your 6.004 TA who says "No problem! In fact, you can produce B_{OUT} with just a single 4-to-1 mux: connect X to S1 and Y to S0, then hook each of the data inputs to the appropriate choice of '0', '1' or B_{IN}." Using this hint, finish off the implementation shown below.

Show connections for data inputs using only '0', '1' or B_{IN}



Problem 10.

The 3-input Boolean function G(A,B,C) computes $\overline{A} \cdot \overline{C} + A \cdot \overline{B} + \overline{B} \cdot \overline{C}$.

(A) How many 1's are there in the output column of G's 8-row truth table?

(B) Give a minimal sum-of-products expression for G.

(C) There's good news and bad news: the bad news is that the stockroom only has G gates. The good news is that it has as many as you need. Using only combinational circuits built from G gates, one can implement (choose the best response)

(A) only inverting functions
(B) only non-inverting functions
(C) any function (G is universal)
(D) only functions with 3 inputs or less
(E) only functions with the same truth table as G

(D) Can a sum-of-products expression involving 3 input variables with greater than 4 product terms *always* be simplified to a sum-of-products expression using fewer product terms?

6.004 Computation Structures
Spring 2017