Lecture 2 11 Jan 2017

Boolean Algebra: Exhaustive Proofs

Steps:

1. Make a table
2. Calculate all values on left-hand side
3. Calculate all values on the right-hand side
4. See if they agree

Second Distributive Law

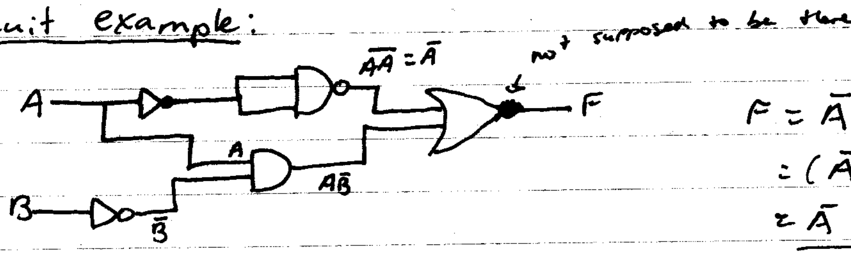
$$(A+C)(B+C) = AB + C$$

ex. Does $\bar{a} + \bar{b} = \overline{a+b}$?

a	b	X	Y	
0	0	1	1	✓
0	1	1	0	X

$\therefore \bar{a}+\bar{b} \neq \overline{a+b}$

Circuit example:



$$F = \bar{A} + A\bar{B}$$
$$= (\bar{A} + A)(\bar{A} + \bar{B})$$
$$= \bar{A} + \bar{B}$$

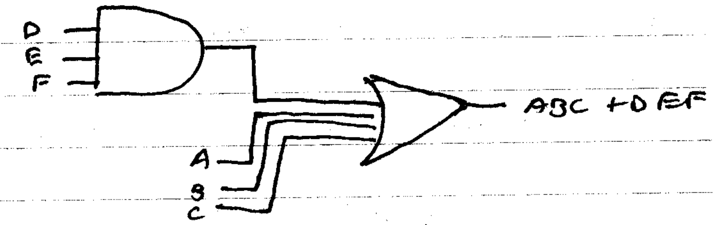
ex. Implement $(A+B+C+D)(A+B+C+E)(A+B+C+F)$ with one multi-input OR and one multi-input AND.

Let $Z = A+B+C$

$= (Z+D)(Z+E)(Z+F)$

$= (Z+DE)(Z+F)$

$= (Z+DEF)$



Duality

- take any valid Boolean identity
- sub AND for OR and 0 for 1
- the result is another valid Boolean identity.

ex. $x(y+z) = xy + xz$

$x(y+z) = (xy) + (xz)$

$x + (yz) = (x+y)(x+z)$

* NEVER say an expression is equal to its dual.

- some formulas are their own duals (ex. $\overline{\overline{x}} = x$)
- self-dual functions have $\overline{F(a,b,c)} = F(\bar{a}, \bar{b}, \bar{c})$

NOTE: $X \cdot \text{ANYTHING} + X = X$

ex. simplify $x + xzy + \bar{x} + x\bar{y}z + \bar{x}y\bar{z} + y\bar{z}$

$= \underbrace{x + \bar{x}}_{1} + xzy + x\bar{y}z \ldots$

$= 1 + xzy + \cdots$ ← 1 + anything is 1 ∴ no need to continue solving.

$= \underline{1}$

- duality is useful for simplifying more complex sets.

Other Dual Theorems

Consensus: $Ab + bc + c\bar{A} = Ab + c\bar{A}$

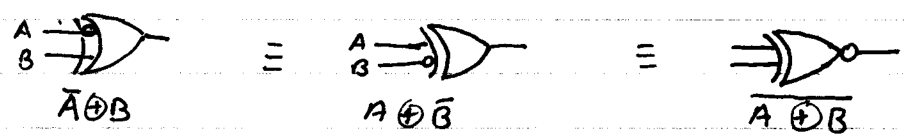
Swap: $(\bar{A} + b)(A + c) = Ab + \bar{A}c$

X-OR

↳ F is 1 if exactly one of a or b is 1.

Useful properties:

↳ inverting any lead changes XOR → XNOR



↳ making one input "1" makes XOR an inverter

Obtaining Formulas from Truth Tables

① Go through the truth table and write down the terms that make the result "1".

② OR all terms together.

ex. $X = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + AB\bar{C}$ ~~+ ABC~~ $+ ABC$

Inputs			Output	
A	B	C	X	
0	0	0	1	$\bar{A}\bar{B}\bar{C}$
0	0	1	0	
0	1	0	0	
0	1	1	0	
1	0	0	1	$A\bar{B}\bar{C}$
1	0	1	0	
1	1	0	1	$AB\bar{C}$
1	1	1	1	ABC

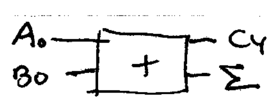
$\therefore X = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + AB\bar{C} + ABC$

Adder

↳ adds 1 bit #s

↳ Σ is an XOR gate

↳ carry (C_4) is an AND gate



Duality

- take any valid Boolean identity
- sub AND for OR and 0 for 1
- the result is another valid Boolean identity.

ex. $x(y+z) = xy + yz$

$x(y+z) = (xy) + (xz)$

$x + (yz) = (x+y)(x+z)$

* NEVER say an expression is equal to its dual.

- some formulas are their own duals (ex. $\overline{X} = x$)
- self-dual functions have $\overline{F(a,b,c)} = F(\bar{a}, \bar{b}, \bar{c})$

NOTE: $X \cdot \text{ANYTHING} + X = X$

ex. simplify $x + xzy + \bar{x} + x\bar{y}z + \bar{x}y\bar{z} + y\bar{z}$

$= \underbrace{x + \bar{x}}_{1} + xzy + x\bar{y}z \ldots$

$= 1 + xzy + \cdots$ ← 1 + anything is 1 ∴ no need to continue solving.

$= \underline{1}$

- duality is useful for simplifying more complex sets.

Other Dual Theorems

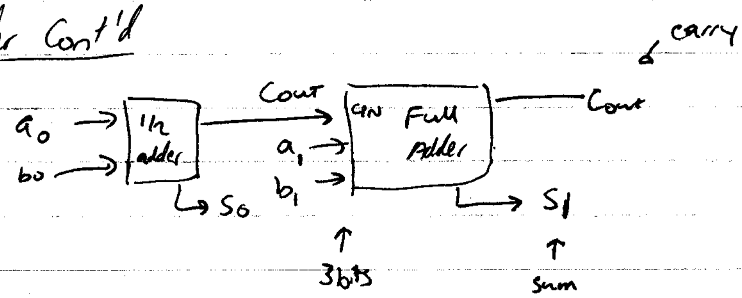
Consensus: $Ab + bc + c\bar{A} = Ab + c\bar{A}$

Swap: $(\bar{A} + b)(A + c) = Ab + \bar{A}c$

Lecture 3

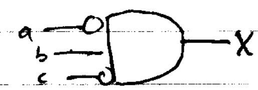
16 Jan 2017

Adder Cont'd



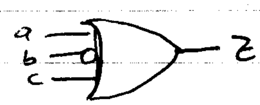
3-Input Gates

Select → identifies a particular bit combination



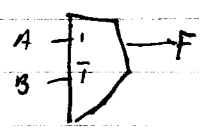
X = 1 when a, b, c = 0, 1, 0

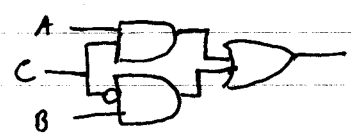
Reject → rejects a certain bit combo



Z = 1 unless a, b, c = 0, 1, 0

Multiplexer (MUX) → when C = 1, A goes through F.
When C = 0, B goes through F.

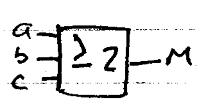


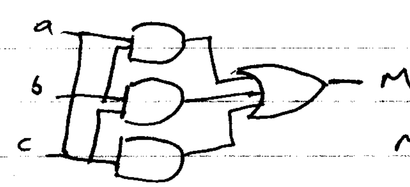


$F = AC + B\overline{C}$

$F = A \ (C=1)$
$F = B \ (C=0)$

Majority Gate (Carry) → Z = 1 if the majority of inputs are 1.





$M = ab + bc + ac + abc$

$abc + ab = ab(c+1) = 0$

$\therefore M = ab + bc + ac$

Error Detection

↳ parity to find transmission errors

$p = 1$ if odd # of 1 inputs

$z = 0$ if even # of 1 inputs

$z = 1$ → bit changed during transmission.

ex.

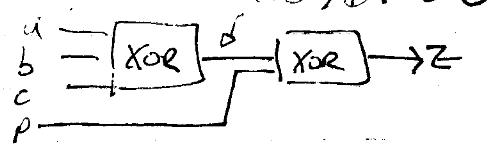
Transmission			
a	b	c	p
1	0	0	1

→ ⊕ (noise) →

Received			
a	b	c	p
1	0	1	1

∴ $Z = 1$

$(1 \oplus 0) \oplus 1 = 0$



Transmission			
a	b	c	p
1	0	0	1

Received			
a	b	c	p
0	1	0	1

→ no error (still even # of 1s)

Circuits From Truth Tables

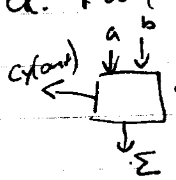
ex.

a	b	c	X
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

$X = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + AB\bar{C} + ABC$

- create circuit.

ex. Full adder.



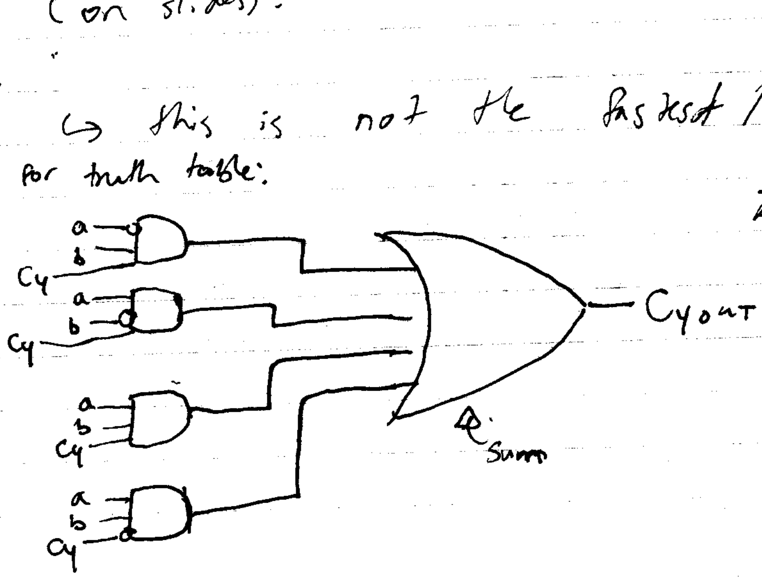
a	b	Cyi	Cyo	
0	0	0	0	
0	0	1	0	
0	1	1	1	$\bar{a}bC_y$
0	1	0	0	
1	0	0	0	
1	0	1	1	$a\bar{b}C_y$
1	1	1	1	$ab\bar{C_y}$
1	1	0	1	abC_y

$\therefore C_y(o) = \bar{a}b \cdot C_y + a\bar{b}C_y + abC_y + ab\bar{C_y}$

- do for sum too (on slides).

NB.
↳ this is not the fastest / most efficient way.

Circuit for truth table:



$\bar{A}BC_y + a\bar{b}C_y + ab\bar{C_y} + abC_y$

↑ OR

use or gate

only one needs

to be 1 to

get Cy out.

Simplifying Full Adder Eq'ns

ex. $C_{y(o)} = \bar{a}b\,C_y + a\bar{b}C_y + abC_y + ab\overline{C_y}$

$= \bar{a}bC_y + abC_y + a\bar{b}C_y$

etc.

De Morgan's Theorem (L3 +4)

-used to find the inverse of expressions

$$\boxed{\overline{AB} = \bar{A} + \bar{B}}$$

Inverse:

$\overline{\bar{A}+\bar{B}} = AB$

Dual:

$\overline{D+E} = \bar{D}\bar{E}$

Dual Inverse:

$\overline{\bar{D}\bar{E}} = D+E$

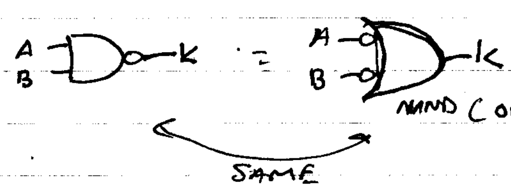
ex.

A	B	$\bar{A}$	$\bar{B}$	$\bar{A}+\bar{B}$	AB	$\overline{AB}$
0	0	1	1	1	0	1
0	1	1	0	1	0	1
1	0	0	1	1	0	1
1	1	0	0	0	1	0

Gate Implementation of De Morgans

$\overline{AB} = K = \bar{A} + \bar{B}$

↑ NAND



NAND (OR GATE w/ INVERTED INPUTS)

De Morgan Complementing Expressions

↳ an OR gate w/ inverted inputs = an AND gate w/ inverted output

↳ an AND gate w/ inverted input = an OR gate w/ inverted output

↳ inverting inputs + outputs of an OR makes an AND

↳ " " AND " OR.

ex. convert $\overline{(a+b)(a+c)}$ to an exp. w/ 3 letters

$\overline{(a+b)(a+\bar{c})} = \overline{(a+b)} + \overline{(a+\bar{c})}$

$= (\bar{a}\bar{b} + \bar{a}\,\bar{\bar{c}})$

$= (\bar{a}\bar{b} + \bar{a}c) = \boxed{\bar{a}(\bar{b}+c)}$

↳ any logic circuit can be made from AND + NOT gates using De Morgan's Law.

ex. reduce $\overline{(a+\bar{b})} + \overline{\bar{\bar{a}b}}$ to 4 letters + single inversion bars.

$$F = \overline{\overline{(a+\bar{b})} + (\overline{\bar{a}b})}$$
$$= \overline{(\overline{a+\bar{b}}) + ab} \qquad A+\bar{B} = \bar{A}\bar{B} \ (DM)$$
$$= \overline{(\overline{a+b})}\,\overline{(ab)}$$
$$= (a+b)(\overline{ab})$$
$$= (a+b)(\bar{a}+\bar{b}) \qquad \overline{AB} = \bar{A}+\bar{B}$$
$$= a\bar{a}_{0} + a\bar{b} + \bar{a}b + b\bar{b}_{0}$$
$$= \boxed{a\bar{b} + \bar{a}b} \quad (XOR)$$

(CMOS → Complementary Metal Oxide Semiconductor)

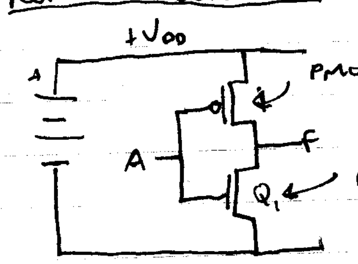
Real CMOS gates

↳ you can't make AND or OR gates directly

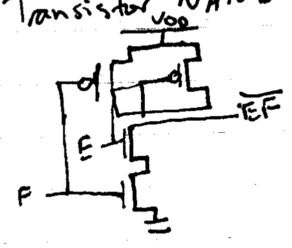
↳ real ones are NAND + NOR $\qquad x$ —▷o— $\bar{x}$

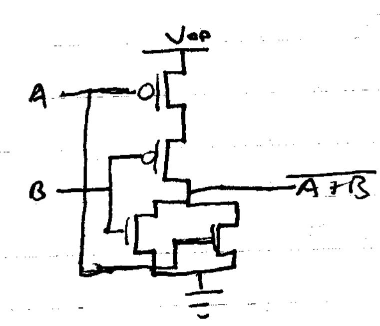
↳ use DM to convert.

Not Gate (Inverter)



Transistor NAND (=D○—) $\qquad$ Transistor NOR (⊐D○—)





A	B	$\bar{A}+\bar{B}$
1	0	1

E	F	$\overline{EF}$
1	0	1

0 = gnd

1 = Vdd

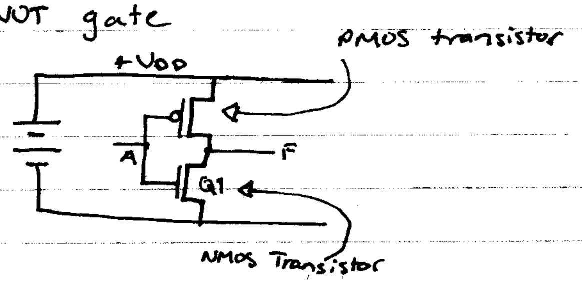
Lecture 4 18 Jan 2017

* Assignment 1 due Jan. 27 on cuLearn *

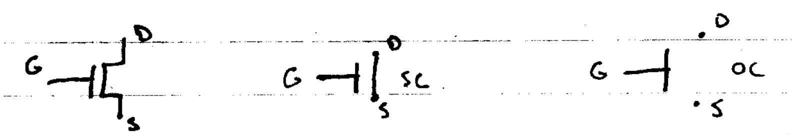
Real Gates Cont'd

- ↳ you can't make AND + OR gates directly
- ↳ all CMOS gates invert
- ↳ real gates are NAND, NOR, + NOT.

ex. NOT gate

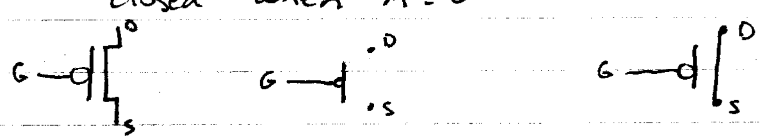


NMOS: open switch when $A = 0$
closed when $A = 1$



NMOS (Reg.) NMOS ($V_g = 1$) NMOS ($V_g = 0$)

PMOS: open switch when $A = 1$
closed when $A = 0$



PMOS (Reg.) PMOS ($V_g = 1$) PMOS ($V_g = 0$)

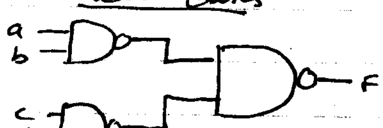
PMOS → turn off with a one input
NMOS → turn on with a one input

To make gates → make AND + OR gates and then invert the output to get NAND + NOR

- to design circuits more easily, use AND + OR and then use DeMorgan's Laws to invert.

ex. $F = \overline{\overline{(ab)} \cdot \overline{(cd)}}$ is equivalent to $F = ab + cd$

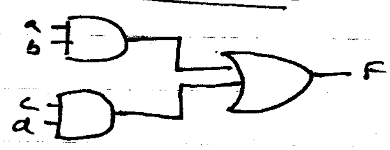
Real Gates



$F = \overline{\overline{(ab)} \cdot \overline{(cd)}}$

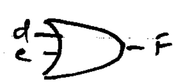
=

Simple Gates



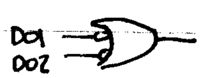
$F = ab + cd$

Note:



is eq. to

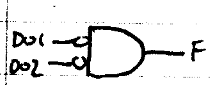
ex. Doors on a train



DO1	DO2	Output	
0	0	1	✓
0	1	1	X
1	0	1	X

(we want both doors 0)

} logic error - we shouldn't be starting with one open + one closed

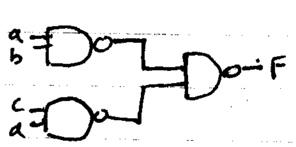


DO1	DO2	Output	
0	0	1	✓
0	1	0	✓
1	0	0	✓
1	1	0	✓

Logic is now correct - we only proceed if both are closed.

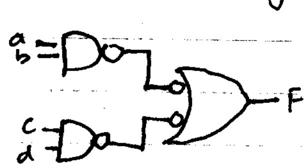
- NAND - NOR logic is more confusing and make harder to read digrams.

ex. Converting from real to simple gates

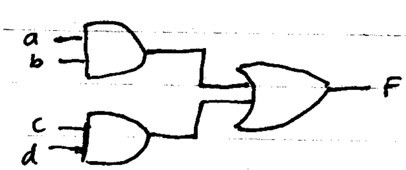


$F = \overline{\overline{ab} \cdot \overline{cd}}$

(real gates)



use DeMorgan to convert NAND gate. negatives cancel out.



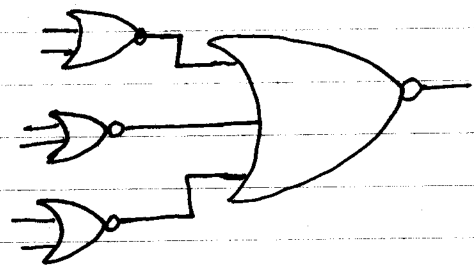
$F = ab + cd$

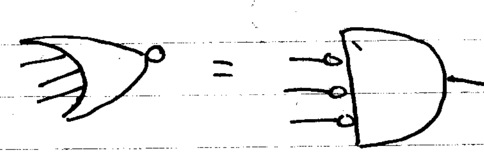
(simple gates)

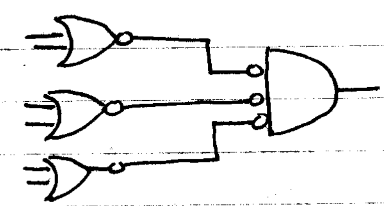
Transforming NAND/NOR to AND/OR

- one output circle cancels all the input circles it feeds
- use DeMorgan alternate symbols

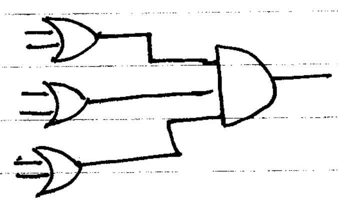
ex. Transform this circuit into simple gates.



① 

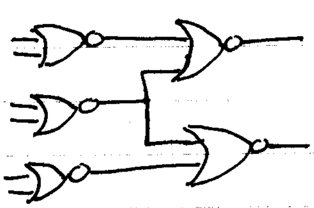
② 

- circles cancel out

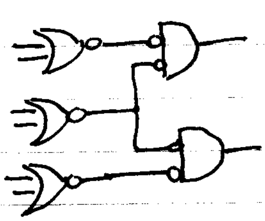
③ 

← final simple gate circuit

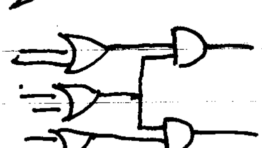
ex. 2



① only change outputs w/ AND



② circles cancel



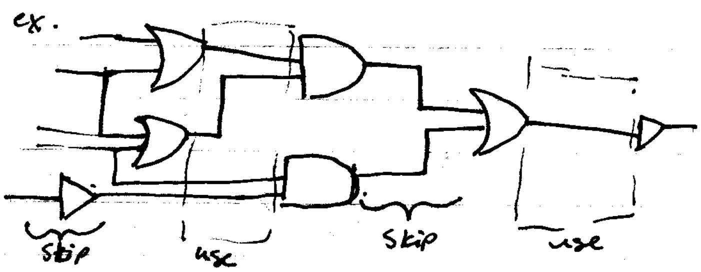
③ Final answer

Graphical Form

↳ start with AND/OR circuit

↳ select alternating connection layers
  - every 2nd layer of wires b/w gates.

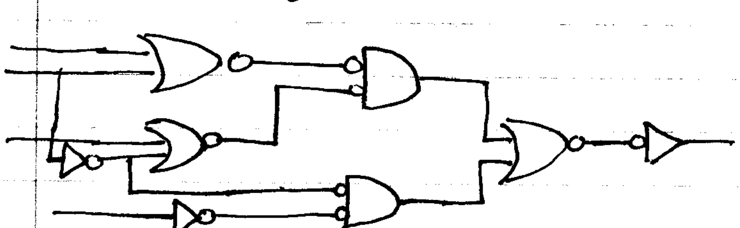
ex.



① Starting circuit (simple gates)

↳ put back to back inverting circles on both ends of the leads

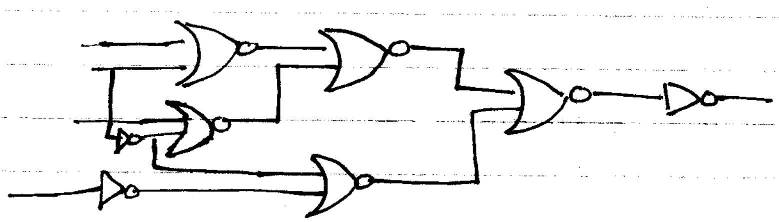
↳ move gates to make logic more simple



② converted to real gates

↳ select unconventional gates (w/ input bubbles) + convert them.

Final circuit:



- see a second example in the lecture slides

[redacted]

Generalized De Morgan

$\overline{F(A, B, C, \ldots +, \cdot)} = F(\bar{A}, \bar{B}, \bar{C} \ldots \cdot, +)$

↳ convert whole bar into single ones

↳ convert and to or & v.v.

Try #49 (lec. slides) for homework.

Using general De Morgan

1. Take Boolean expression
2. Bracket all groups of ANDS
3. Change AND → OR & OR → AND
4. Clean up brackets
5. Invert variables.

Do #51 (lec. slides). Answer: $H' = ((a'+b')c + g')(d'+e'+a'b')$

($\ast$ $X' = \bar{X}$).

#50

a) Convert $\overline{(a+b)(a+\bar{c})}$ using Generalized De Morgan.

1. Let $F = (a+b)(a+\bar{c})$.
2. Take the dual.

   $F = (ab) + (a\bar{c})$

3. Invert.

   $F = \bar{a}\bar{b} + \bar{a}c$

   $= \bar{a}(\bar{b} + c)$ ← final answer.

b) $F = \overline{(a+b)(a+\bar{c})}$ use $\overline{AB} = \bar{A} + \bar{B}$

$= \overline{(a+b)} + \overline{(a+\bar{c})}$ use $\overline{A+B} = \bar{A}\bar{B}$

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

$= \bar{a}(\bar{b} + c)$ ← same answer as general De Morgan.