CALIFORNIA STATE UNIVERSITY FULLERTON™

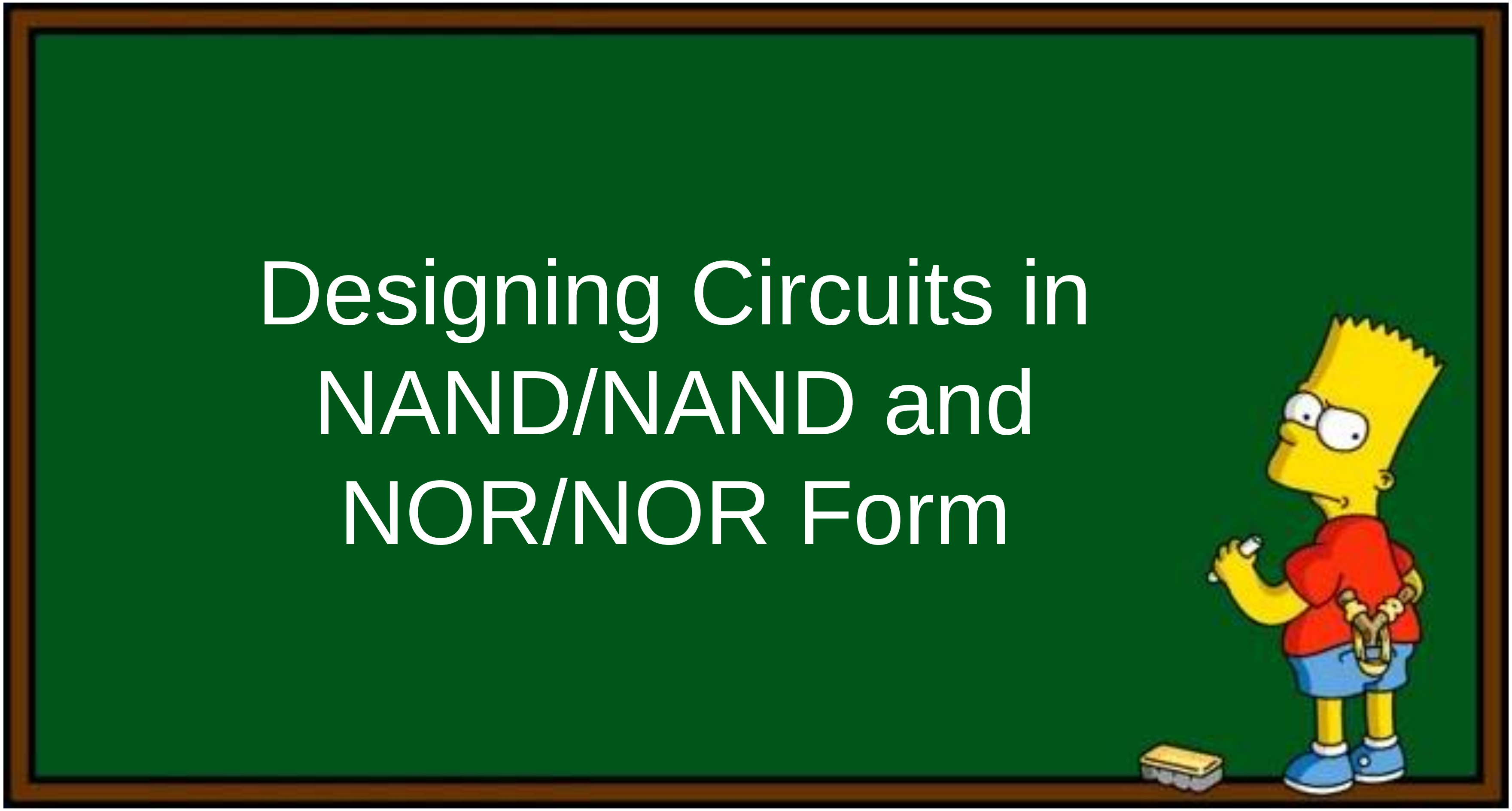
EGEC 180 – Digital Logic and Computer Structures

Spring 2024

Lecture 8: Designing Circuits in AND/NAND and OR/NOR(2.3.3)

Rakesh Mahto, Ph.D.
Office*: E 314, California State University, Fullerton*
Office Hour*: Monday and Wednesday 2:00 - 3:30 pm*
Or by appointment
Office Hour Zoom Meeting ID*: 891 2907 5346*
Email*: ramahto@fullerton.edu*
Phone No*: 657-278-7274*

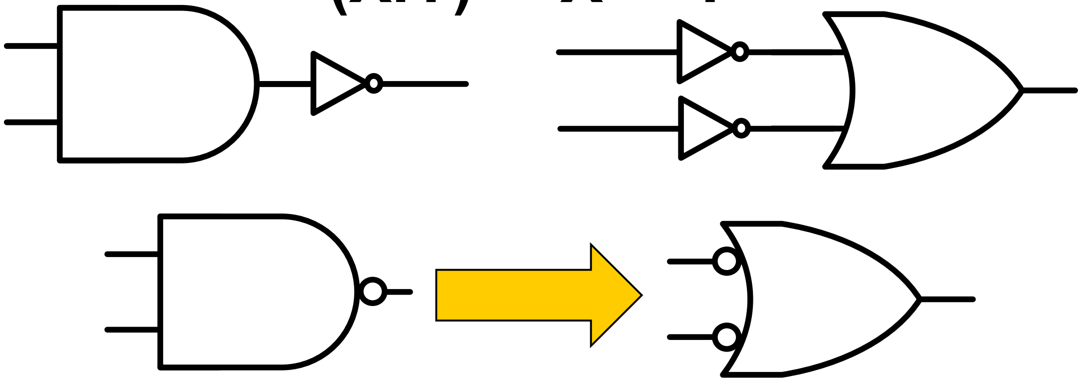
Designing Circuits in NAND/NAND and NOR/NOR Form



DeMorgan’s Theorem

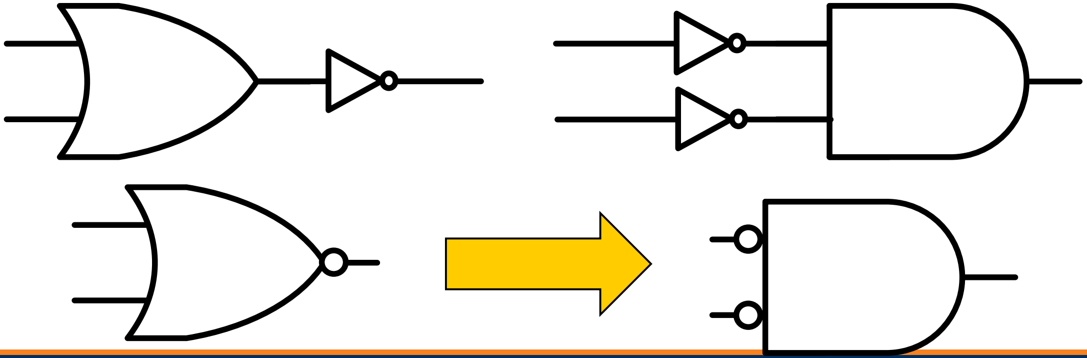
$X.Y = (X' + Y')'$

$(X.Y)' = X' + Y'$

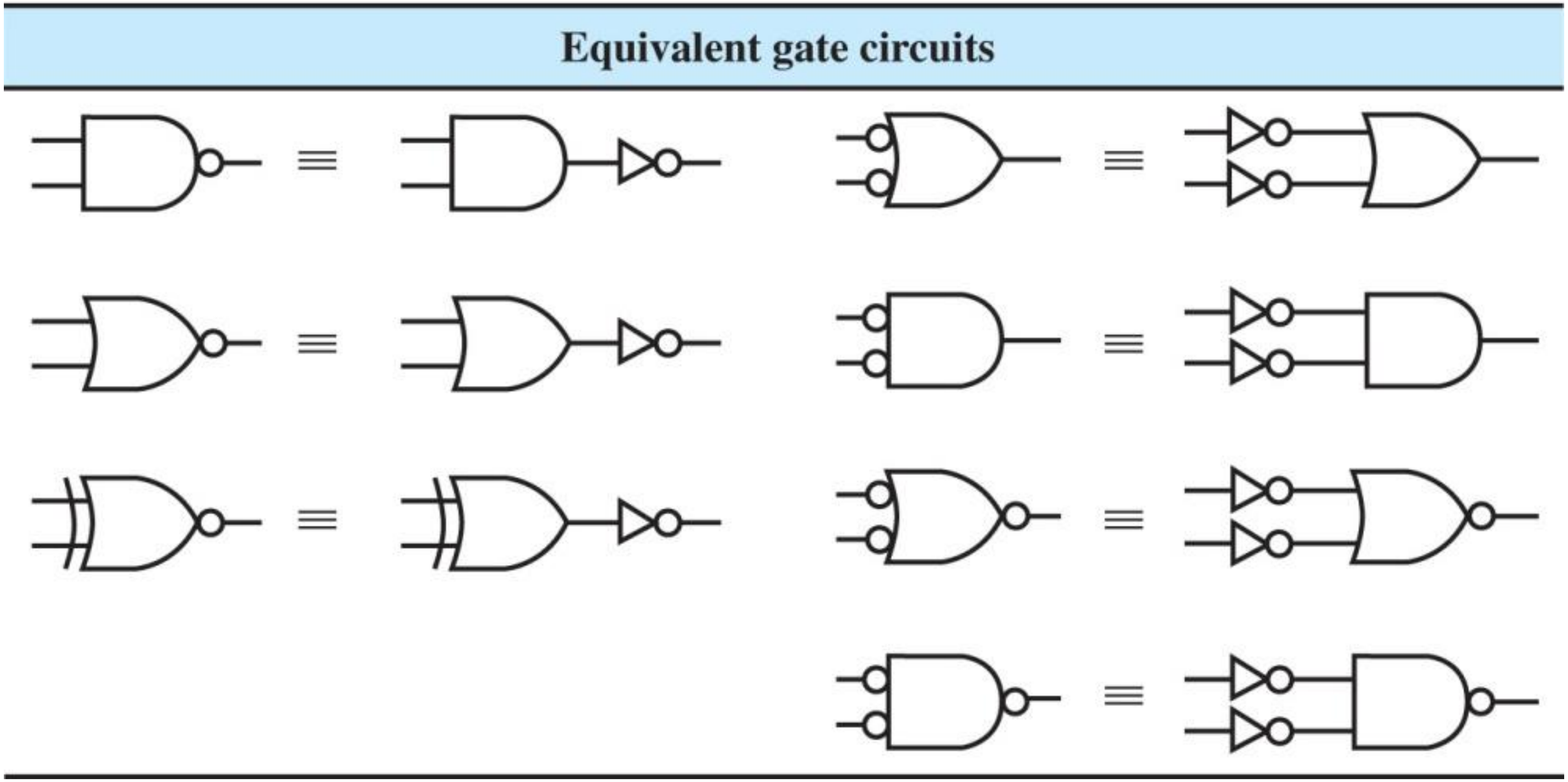


$X + Y = (X'.Y')'$

$(X+Y)' = X'.Y'$



Equivalent Gate Circuits

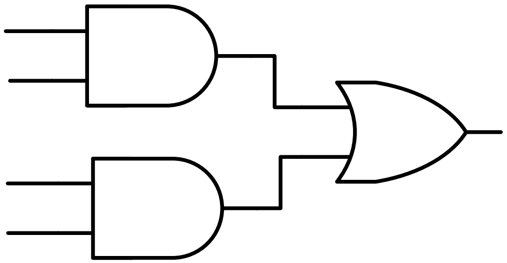


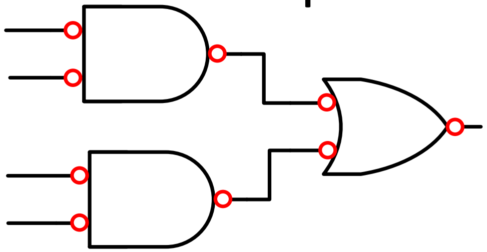
Equivalent gate circuits

DeMorgan Rules
Graphical Symbols

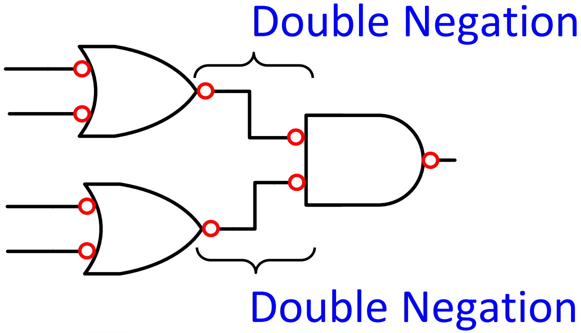
To obtain a De Morgan equivalent gate symbol for an AND, OR, NAND, or NOR gate:

1. Add bubbles to all inputs and bubbles to all outputs

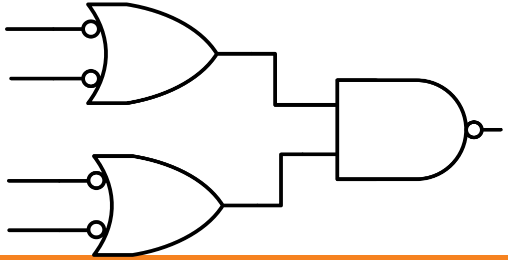




2. **Change ANDs to ORs and Change ORs to ANDs**



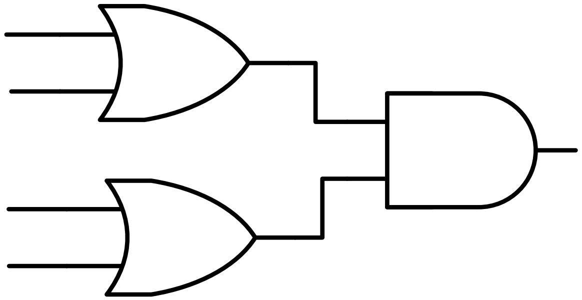
3. **Two bubbles result in no bubble; Double Negation Theorem**

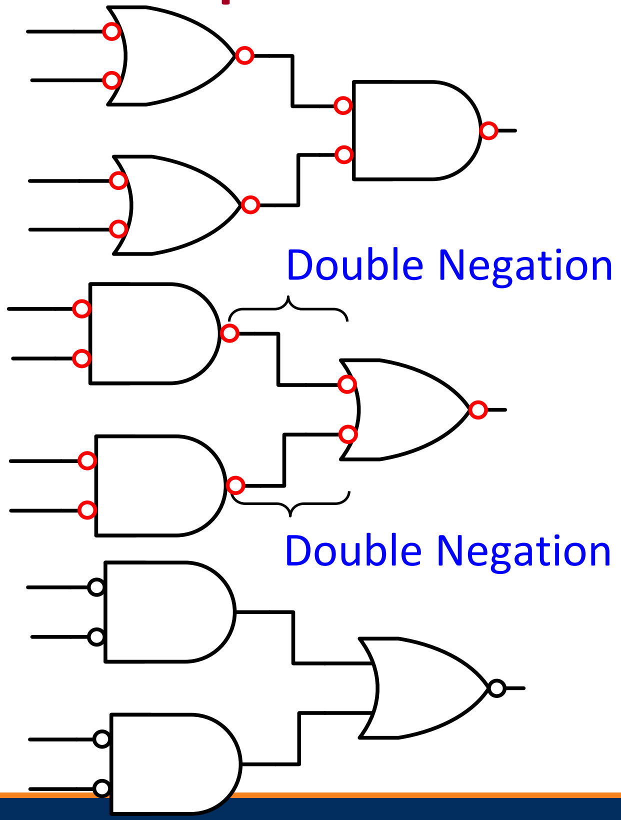


Example Problem #1
Graphical DeMorgan’s

To obtain a De Morgan equivalent gate symbol for an AND, OR, NAND, or NOR gate:

1. **Add bubbles to all inputs and bubbles to all outputs**





2. **Change ANDs to ORs and Change ORs to ANDs**

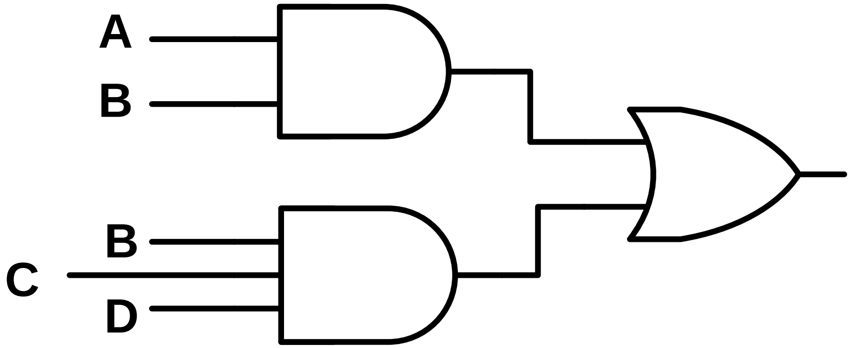
3. **Two bubbles result in no bubble; Double Negation Theorem**

DeMorgan’s Theorem
Practice Problem #1

Given the following truth function $F_1(A,B,C,D) = ABC' + ABCD' + BCD$

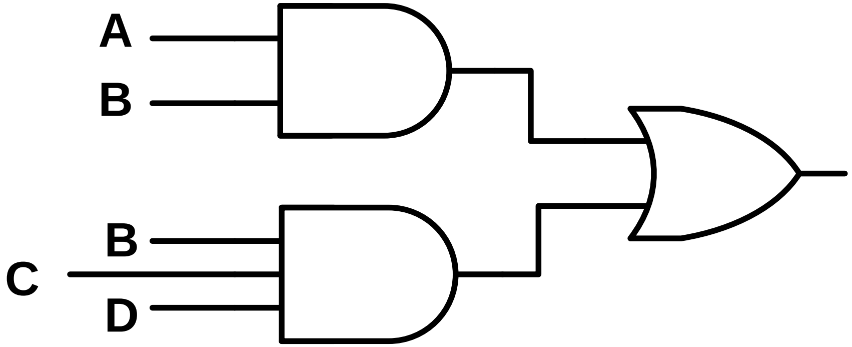
$F_1(A,B,C,D) = AB + BCD$

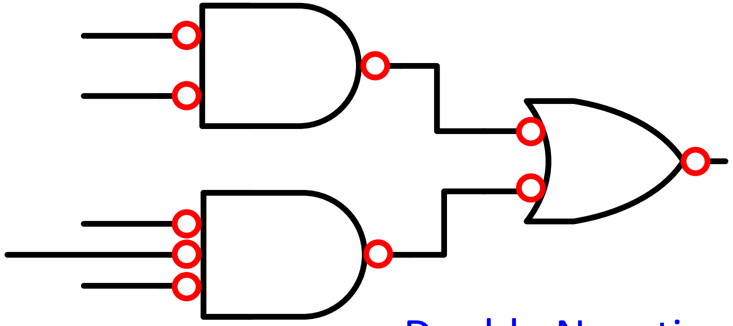
AB	CD	C’D’ 00	C’D 01	CD 11	CD’ 10
A’B’	00	0	0	0	0
A’B	01	0	0	1	0
AB	11	1	1	1	1
AB’	10	0	0	0	0



Practice Problem #1
Graphical DeMorgan's

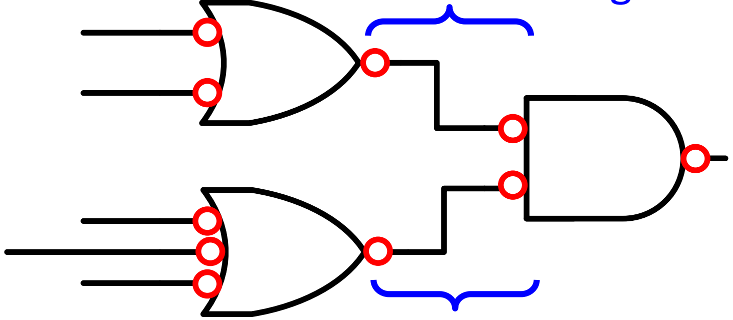
1. Add bubbles to all inputs and bubbles to all outputs





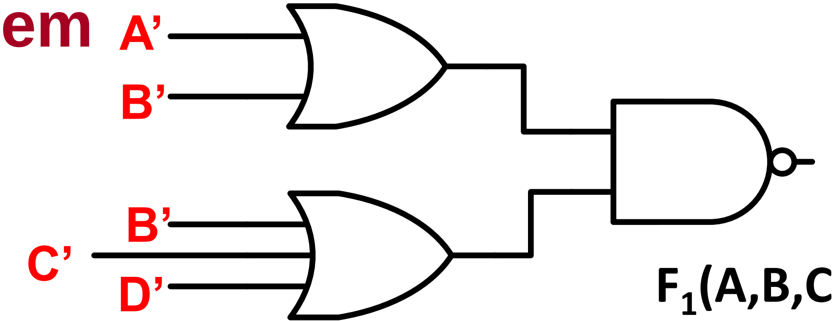
2. Change ANDs to ORs and Change ORs to ANDs

Double Negation



Double Negation

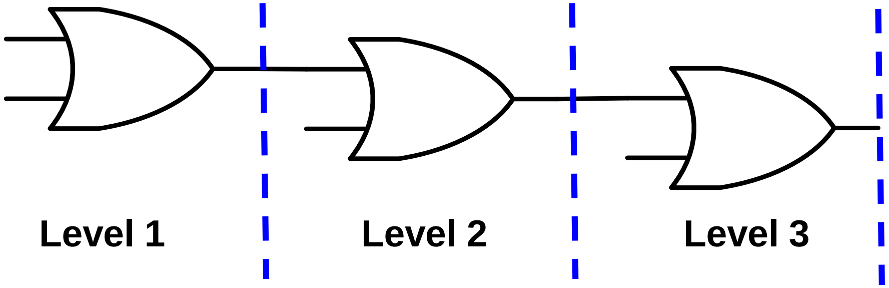
3. Apply Double Negation Theorem



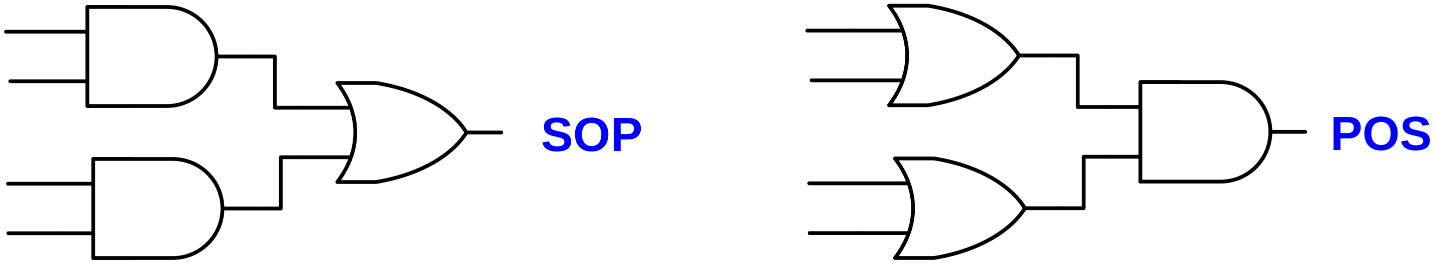
$F_1(A,B,C,D) = [(A'+B')(B'+C'+D')]'$

Two-Level Gate Circuits

The maximum number of gates cascaded in series between a circuit input and output is referred to as the number of levels.



A function written in SOPs form or POSs form corresponds directly to a two-level gate circuit.



We assume that all literals are available as circuit inputs. Inverters used to form the variable complements are not counted when determining the number of levels in a circuit.

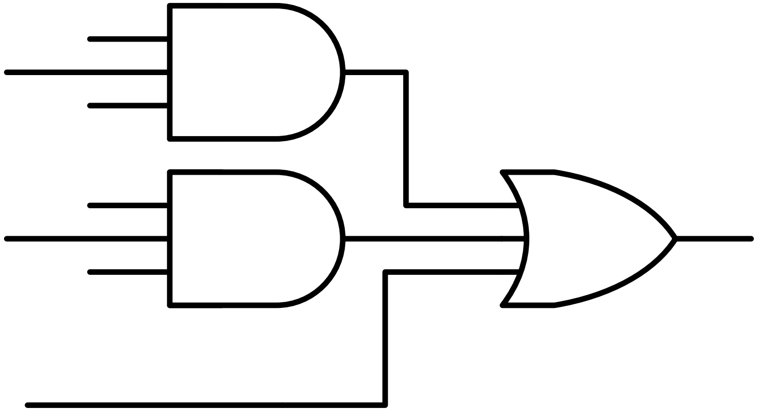
AND-OR to NAND-NAND Transformation

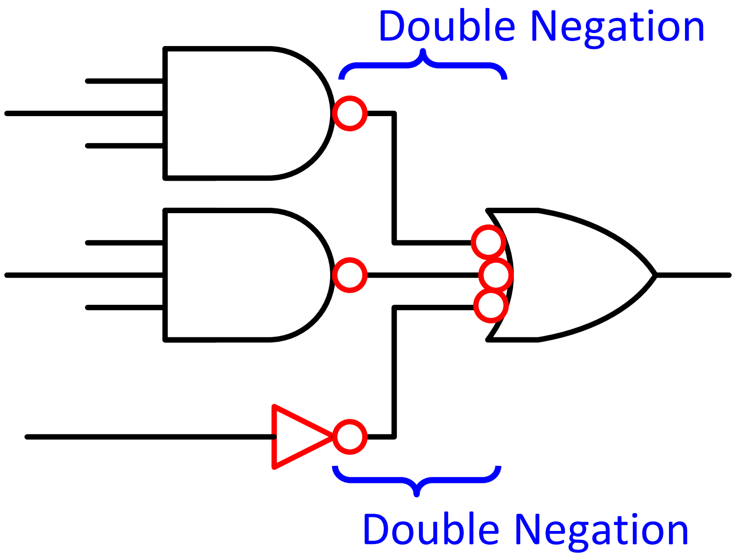
Procedure for designing a minimum two-level NAND-NAND circuit:

1. **Find a minimum *sum-of-products* expression for F using a Karnaugh Map**

2. **Draw the corresponding two-level AND-OR circuit.**

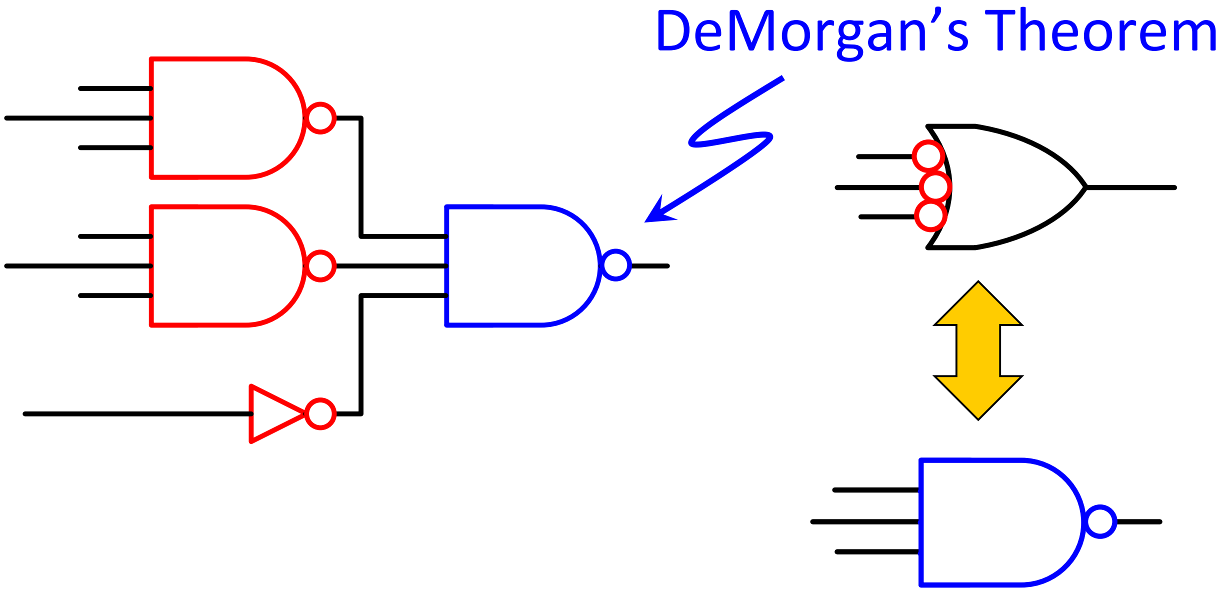
3. **Add bubbles to inputs and bubbles to outputs**





AND-OR to NAND-NAND Transformation

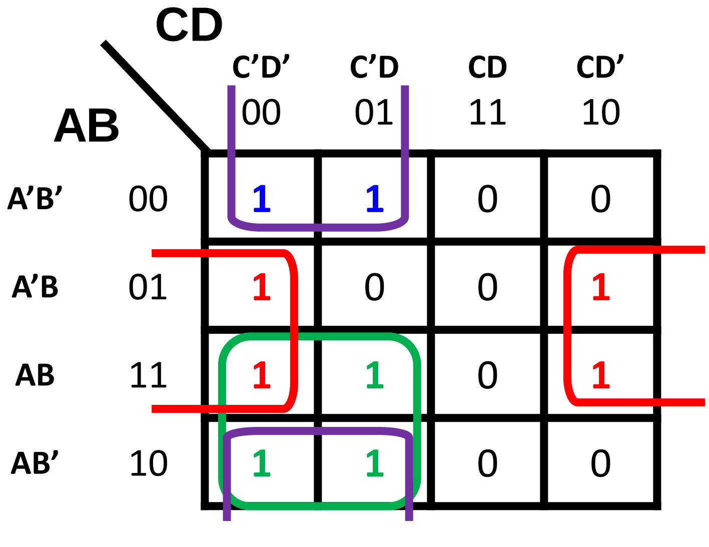
4. **Replace all gates with NAND gates leaving the gate interconnection unchanged.**



AND-OR to NAND-NAND Transformation (Practice Problem #2)

Given $F_2(A,B,C,D) = A'B'C' + BD' + AC'$

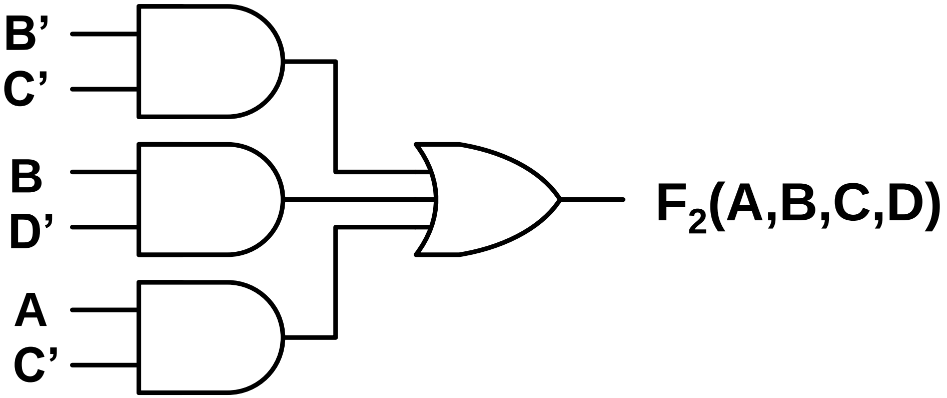
1. Find a minimum *sum-of-products* expression for F using a Karnaugh Map



AB \ CD		C'D' 00	C'D 01	CD 11	CD' 10
A'B'	00	1	1	0	0
A'B	01	1	0	0	1
AB	11	1	1	0	1
AB'	10	1	1	0	0

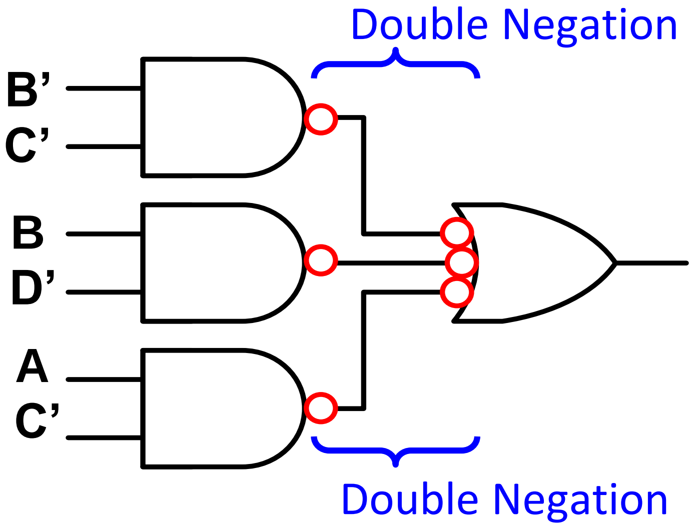
$F_2(A,B,C,D) = B'C' + BD' + AC'$

2. Draw the corresponding two-level AND-OR circuit.

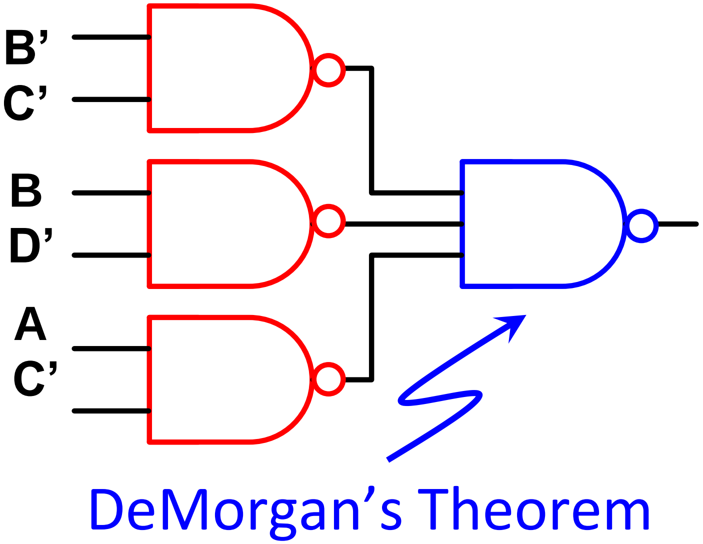


AND-OR to NAND-NAND Transformation

3. Add bubbles to inputs and bubbles to outputs



4. Replace all gates with **NAND** gates leaving the gate interconnection unchanged.



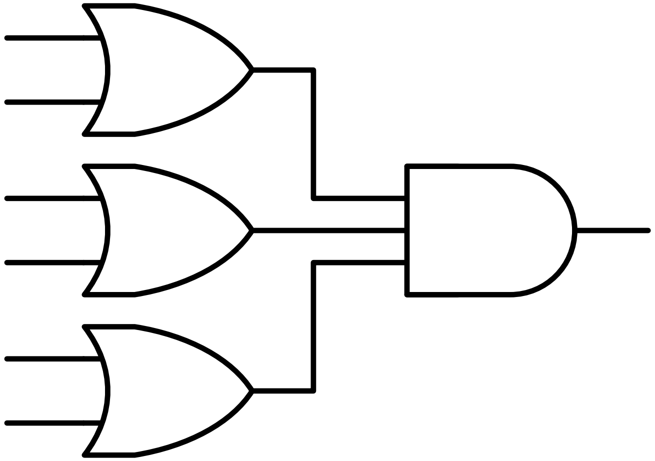
OR-AND to NOR-NOR Transformation

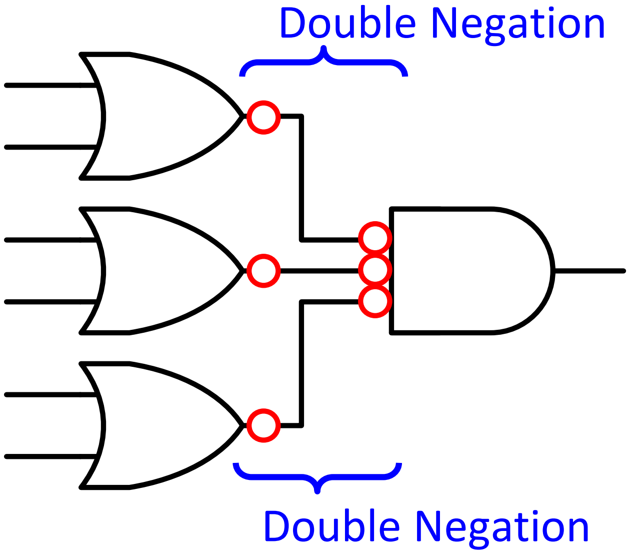
Procedure for designing a minimum two-level NAND-NAND circuit:

1. **Find a minimum *product-of-sum* expression for F using a Karnaugh Map**

2. **Draw the corresponding two-level OR - AND circuit.**

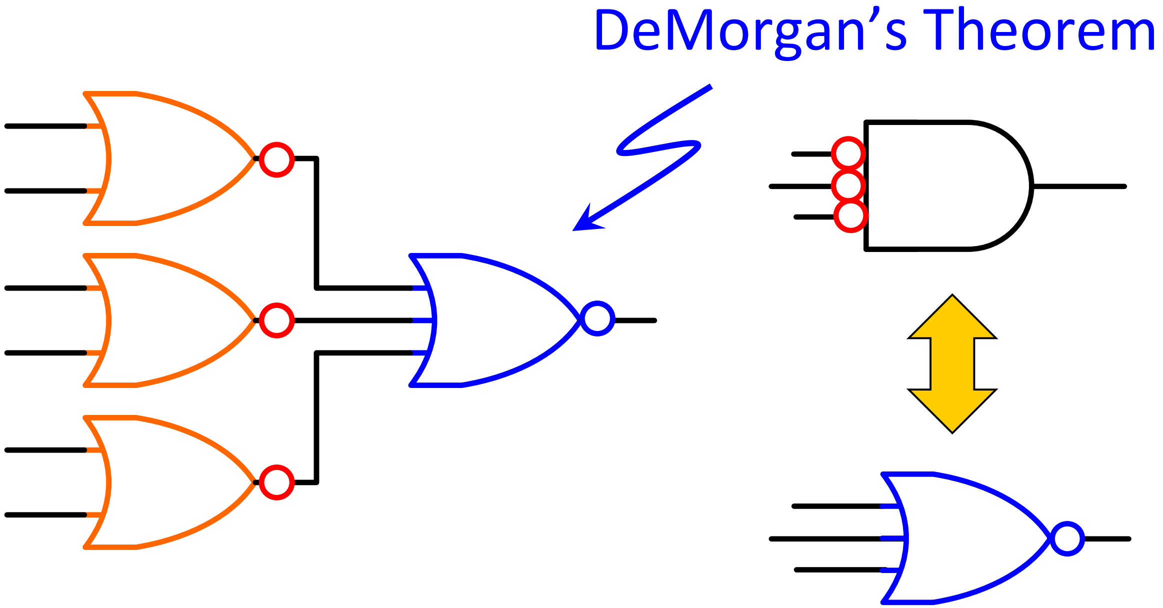
3. **Add bubbles to inputs and bubbles to outputs**





OR-AND to NOR-NOR Transformation

4. **Replace all gates with NOR gates leaving the gate interconnection unchanged.**



OR/AND to NOR/NOR (Practice Problem #3)

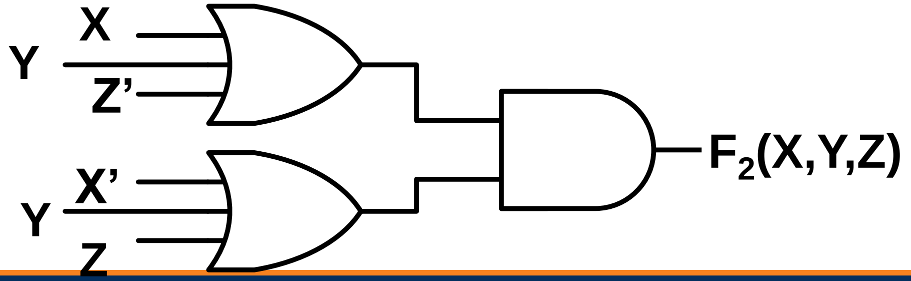
Given $F_2(X,Y,Z) = X'Z' + XZ + Y$

1. Find a minimum *product-of-sum* expression for F using a Karnaugh Map

X \ YZ	Y'Z' 00	Y'Z 01	YZ 11	YZ' 10
X' 0	1	0	1	1
X 1	0	1	1	1

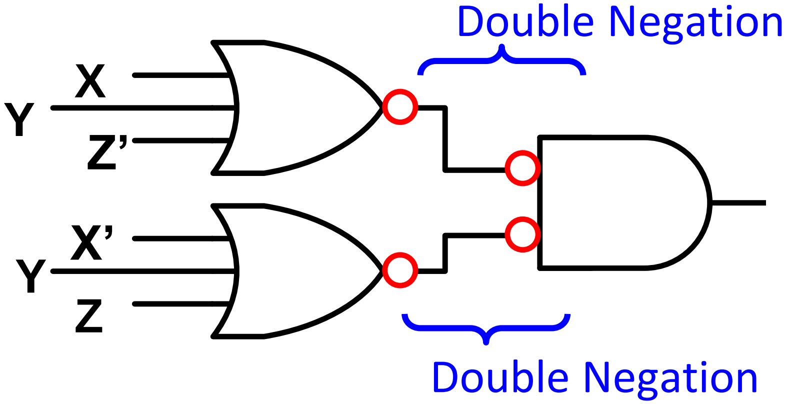
$$F_2(X,Y,Z) = (X+Y+Z')(X'+Y+Z)$$

2. Draw the corresponding two-level **OR-AND** circuit.

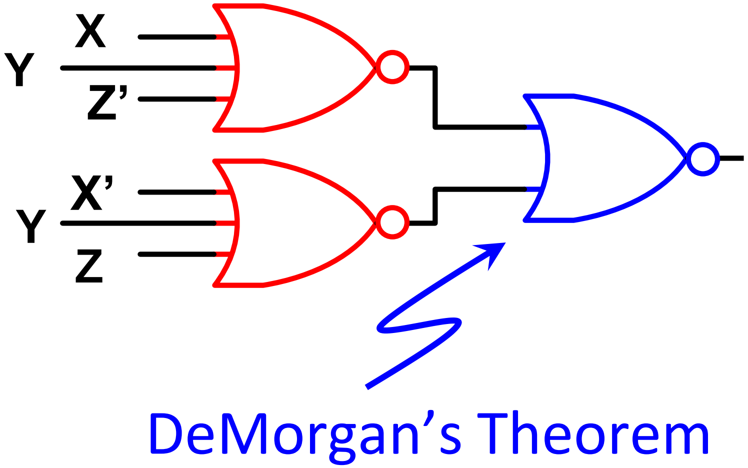


OR/AND to NOR/NOR (Practice Problem #3)

3. Add bubbles to inputs and bubbles to outputs



4. Replace all gates with NOR gates leaving the gate interconnection unchanged.



Q&A