CALIFORNIA STATE UNIVERSITY
FULLERTON™

# EGEC 281: Designing with VHDL

Fall 2024

## Lecture 7: Designing Circuit using NAND or NOR Gate


*Rakesh Mahto, Ph.D.*
***Office**: E 314, California State University, Fullerton*
***Office Hours**: Monday and Wednesday 2:00 - 3:30 pm*
*Or by appointment*
***Office Hour Zoom Meeting ID**: 894 4126 5483*
***Email**: ramahto@fullerton.edu*
***Phone No**: 657-278-7274*

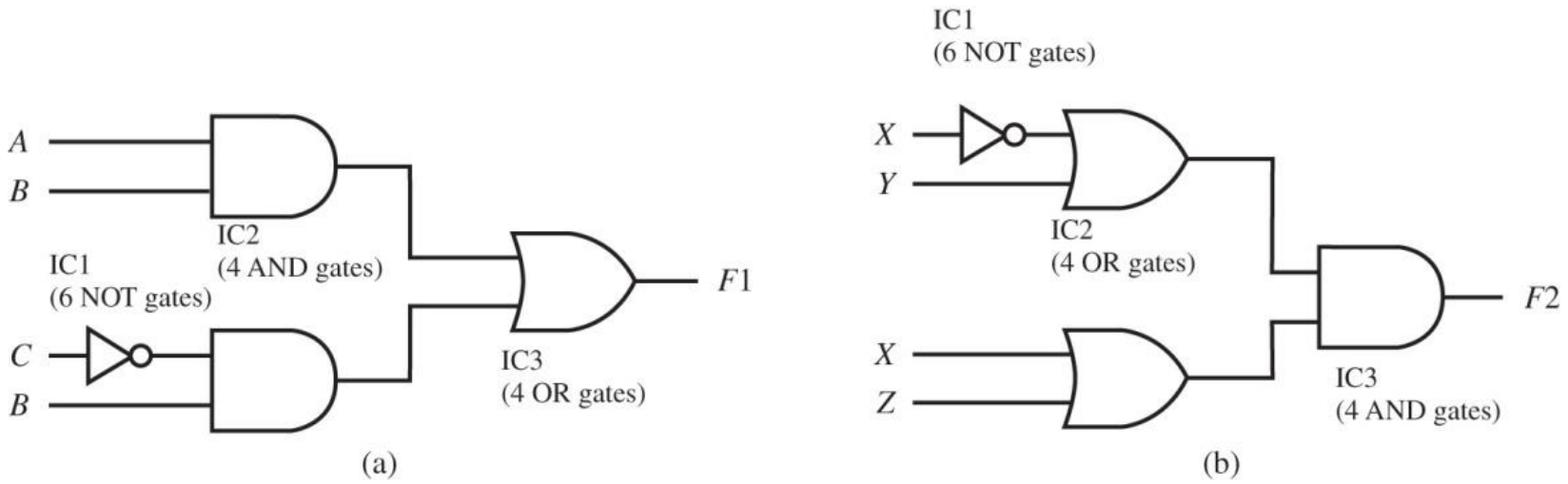
# Analysing IC Logic Circuits

# Analyzing IC Logic Circuits

Circuit Analysis: the process of obtaining a Boolean function for a schematic or a circuit diagram.

- AND/OR format comes from SOP (SOM)
- OR/AND format comes from POS (POM)






F1 = AB + C'B or F1' = (A' + B')(C + B')

F2 = (X' + Y)(X + Z) or F2' = XY' + X'Z'

# Designing Circuits in NAND/NAND and NOR/NOR Form

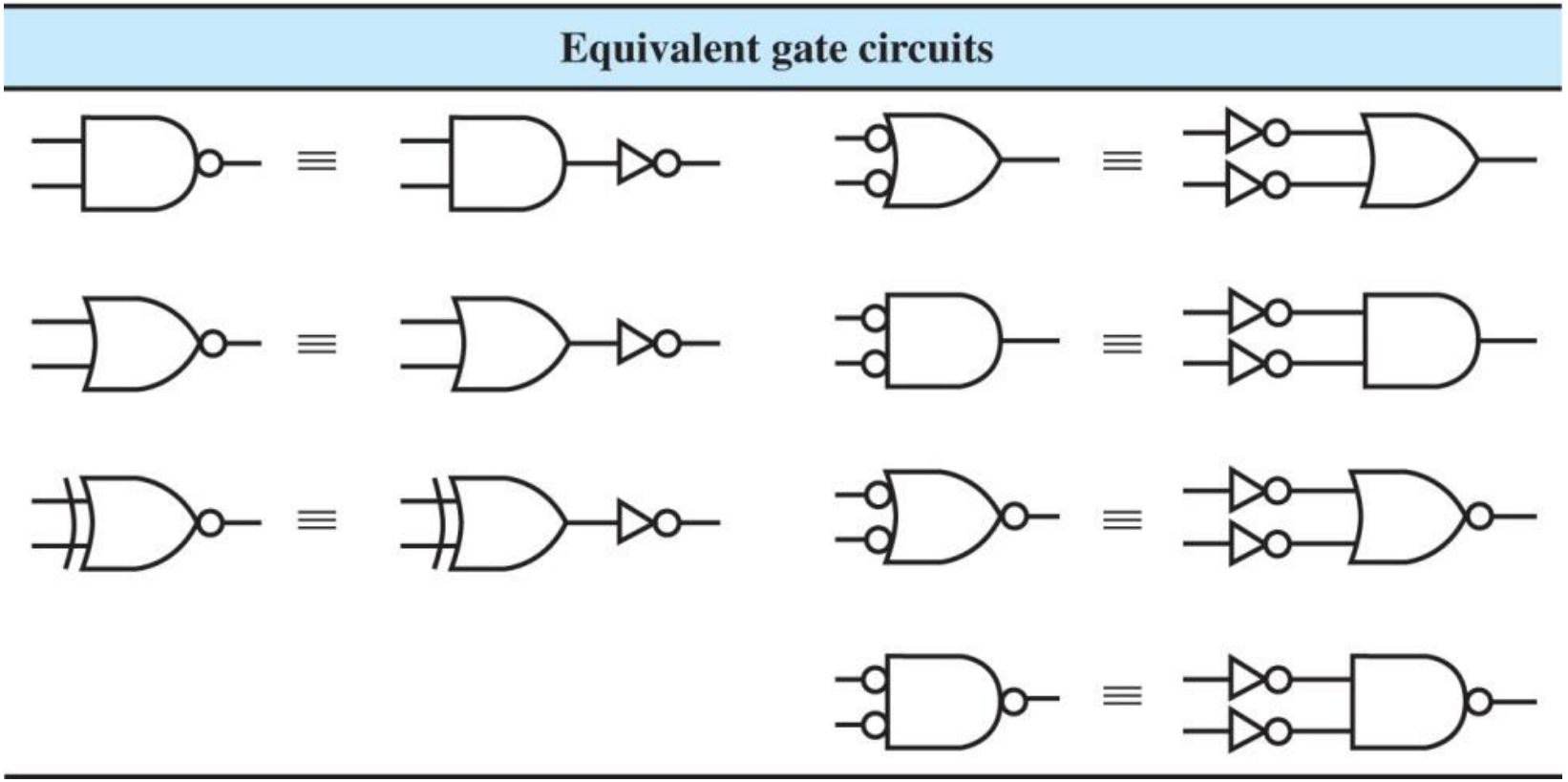
# DeMorgan Rules

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

- Add bubbles to all inputs
- Add bubbles to all outputs
- Change ANDs to ORs
- Change ORs to ANDs
- Two bubbles result in no bubble; Double Negation Theorem

# Equivalent Gate Circuits





# Designing Circuits with NAND/NAND and NOR/NOR Form

Why NAND and NOR gates:

- NAND gates are generally faster than AND gates, and NOR gates are generally faster then OR gates in the same logic family.
- NAND gates and NOR gates are available with a larger variety of fan-ins to choose from than AND gates or NOR gates.
- Fewer IC packages are required to design circuits that use NAND gates or NOR gates.

# 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 variables and their complements are available as circuit inputs. Thus, we do not count inverters which are connected directly to input variables when determining the number of levels in a circuit.

# AND-OR to NAND-NAND Transformation

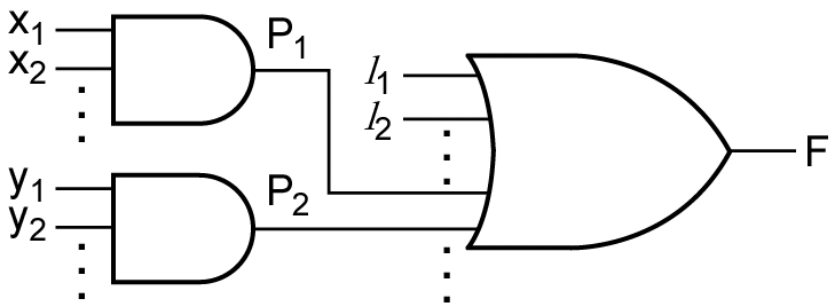
## Design of Minimum Two-Level NAND-NAND Circuits

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

1. Find a minimum *sum-of-products* expression for F.
2. Draw the corresponding two-level AND-OR circuit.
3. Replace all gates with NAND gates leaving the gate interconnection unchanged. If the output gate has any single literals as inputs, complement these literals.

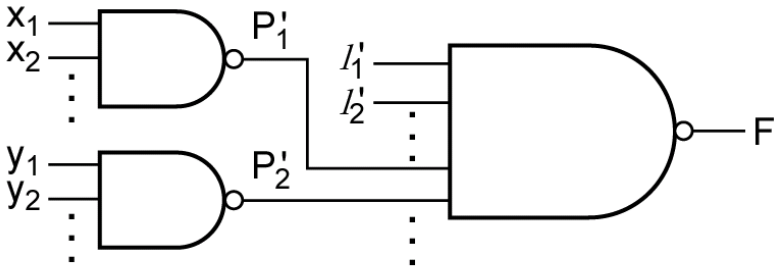
# AND-OR to NAND-NAND Transformation

## Design of Minimum Two-Level NAND-NAND Circuit




(a) Before transformation

$$F = l_1 + l_2 + \cdots + P_1 + P_2 + \cdots$$




(b) After transformation

$$F = (l_1' l_2' \cdots P_1' P_2' \cdots)'$$

# AND-OR to NAND-NAND Transformation

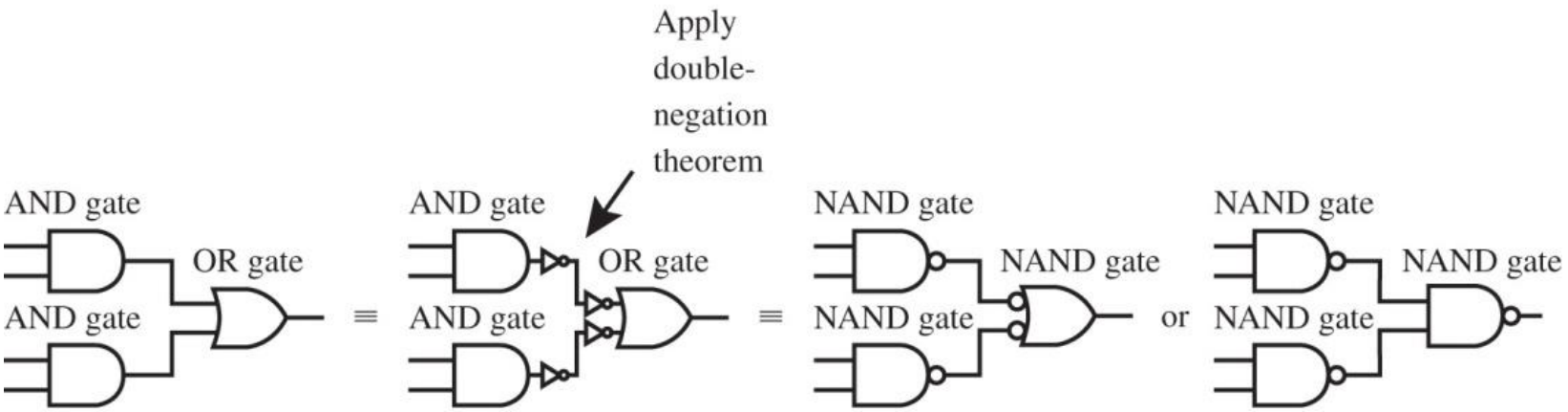
## Design of Minimum Two-Level NAND-NAND Circuits

*Procedure for designing a minimum two-level NAND-NAND circuit (another algorithm):*

1. Find a minimum *sum-of-products* expression for F.
2. Draw NAND gate for each product term with at least two variables. Use for inputs the actual variables.
3. Draw single NAND gate with inputs from outputs of previous gates.
4. A term with a single literal requires and inverter.

# AND/OR to NAND/NAND






**Step 1**
(First draw circuit in SOP form, i.e., AND/OR form)

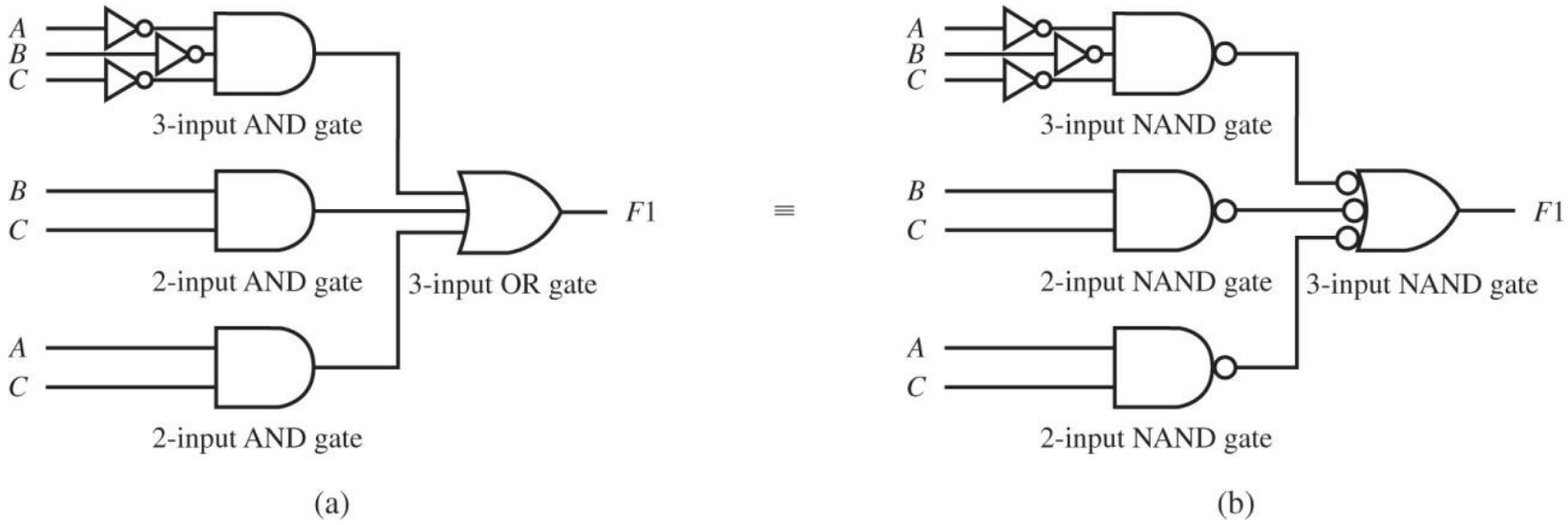
**Step 2**
(Apply DNT)

**Step 3**
(Use NAND gates)

# AND/OR to NAND/NAND

F1 = A'B'C' + BC + AC'





# VHDL Design for function F1

F1 = A'B'C' + BC + AC'



```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity comb4 is port (
     A, B, C : in std_logic;
     F1 : out std_logic
     );
end comb4;

architecture Boolean_function of comb4 is
begin
     F1 <= (not A and not B and not C) or (B and C) or (A and C);
end Boolean_function;
```

# OR-AND to NOR-NOR Transformation

## Design of Minimum Two-Level NOR-NOR Circuits

*Procedure for designing a minimum two-level NOR-NOR circuit:*

1. Find a minimum *product-of-sums* expression for F.
2. Draw the corresponding two-level OR-AND circuit.
3. Replace all gates with NOR gates leaving the gate interconnection unchanged. If the output gate has any single literals as inputs, complement these literals.

# OR-AND to NOR-NOR Transformation

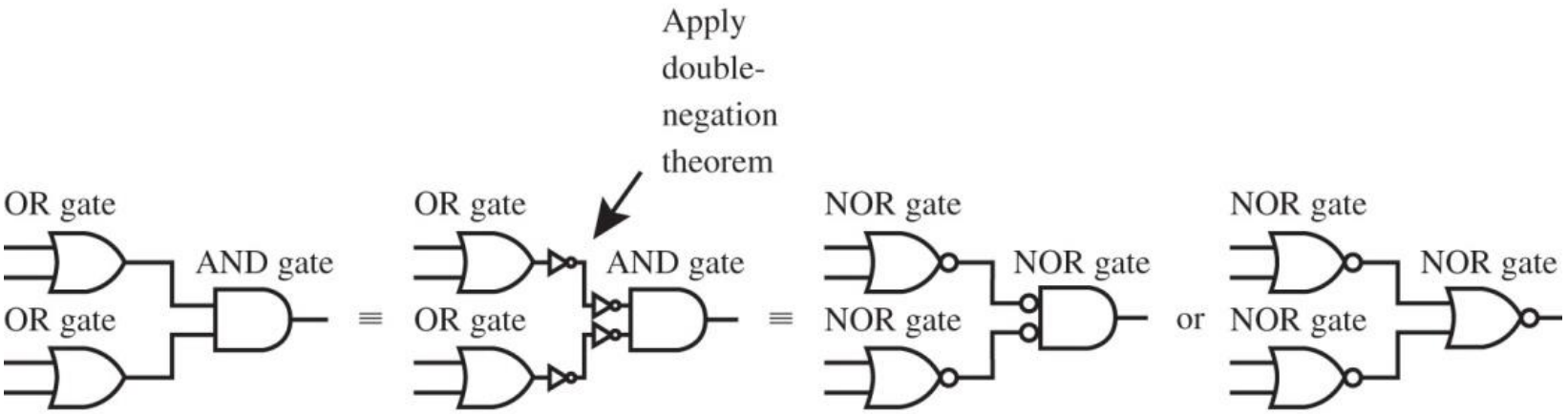
## Design of Minimum Two-Level NOR-NOR Circuits

*Procedure for designing a minimum two-level NOR-NOR circuit (another algorithm):*

1. Find a minimum *product-of-sums* expression for F.
2. Draw NOR gate for each sum term with at least two variables. Use for inputs the actual variables.
3. Draw single NOR gate with inputs from outputs of previous gates.
4. A term with a single literal requires and inverter.

# OR/AND to NOR/NOR






**Step 1**
(First draw circuit in POS form, i.e., OR/AND form)

**Step 2**
(Apply DNT)

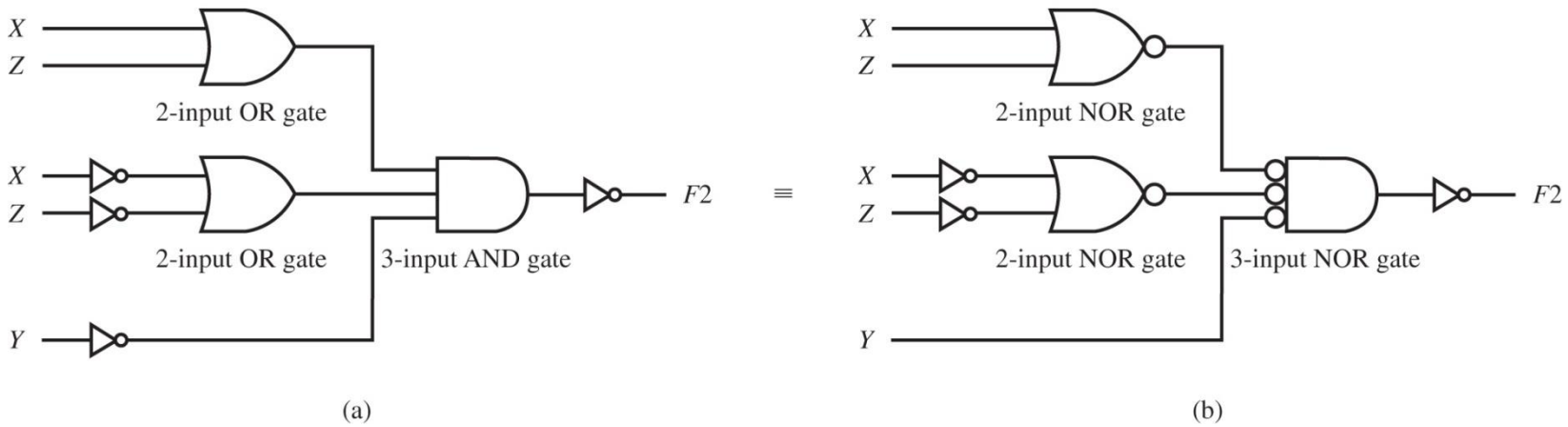
**Step 3**
(Use NOR gates)

# OR/AND to NOR/NOR

F2 = X'Z' + XZ + Y

F2' = (X + Y)(X' + Z')Y'





# VHDL Design for function F2

$F2 = X'Z' + XZ + Y$

$F2' = (X + Y)(X' + Z')Y'$



```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity comb5 is po.5rt (
     X, Y, Z : in std_logic;
     F2 : out std_logic
     );
end comb5;

architecture Boolean_function of comb5 is
begin
     F2 <= not ((X or Z) and (not X or not Z) and not Y);
               --POS form for F2
     --F2 <= (not X and not Z) or (X and Z) or Y;
               --This is an alternate description for F2
               --i.e., an SOP form for F2
end Boolean_function;
```

# Q&A