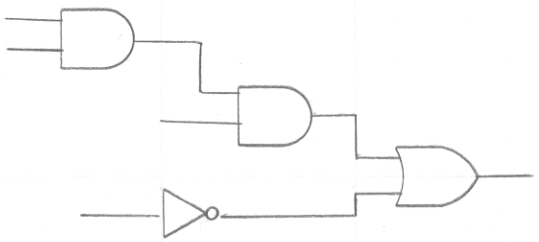
7.1 Multi-Level Gate Circuits

Level of a circuit ≡ maximum number of gates in series between input and output

Sum-of-Products: 2 level circuit (AND-OR)

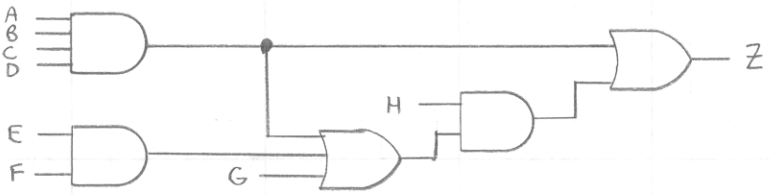
Product-of-Sums: 2 level circuit (OR-AND)

Ex:



3 level circuit

Ex: $Z = (ABCD + EF + G)H + ABCD$



4 level circuit

Why does level matter?

1. Multi-level may have fewer gates
2. Limited number of inputs per gate (ie., limited fanin).
3. Limited number of loads (gates) connected to the output of a gate (limited fanout).
4. Delay: a change in the input variable value takes time to induce a change in the output value (propagation delay). A circuit can't run faster than its slowest path.

Ex: Suppose each gate has a propagation delay of 100ps. The slowest delay for the circuit for Z above takes 400ps.

7.2 NAND and NOR Gates

A logic gate (or gates) is/are <u>functionally complete</u> if they are able to generate <u>any</u> Boolean function.

Operators for minterm expansions are functionally complete (AND, OR, NOT)
Operators for maxterm expansions are functionally complete (OR, AND, NOT)

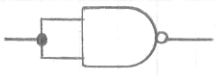
AND is <u>not</u> functionally complete $(A \cdot A \not\Rightarrow A')$
OR " " " " $(A + A \not\Rightarrow A')$
} can't generate inverse.

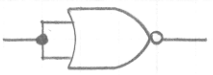
NOT (inverter) is <u>not</u> functionally complete - can't generate two input operations.

NAND, NOR } <u>are</u> functionally complete.

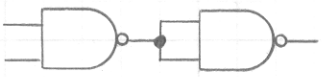
Difficult to prove directly, but easier to say that we can generate <u>any</u> function by its minterm/maxterm expansion using AND, OR, NOT, and then implement these using NAND / NOR:

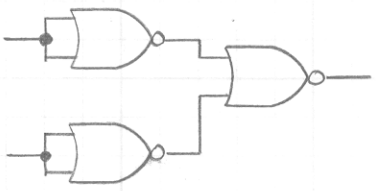
NOT:





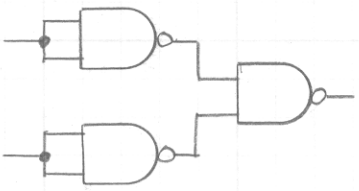
AND:





(De Morgan)

OR:

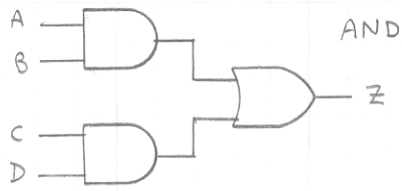


(De Morgan)

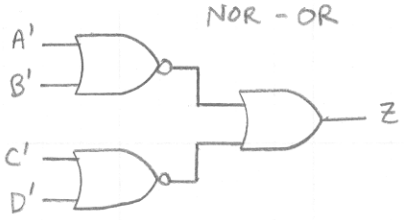
∴ NAND / NOR are functionally complete.

Moving Between Different Types of Two Level Circuits

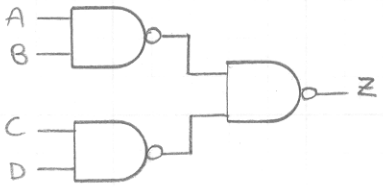
Starting with Sum-of-Products:



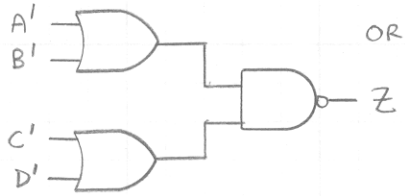
AND-OR



NOR-OR

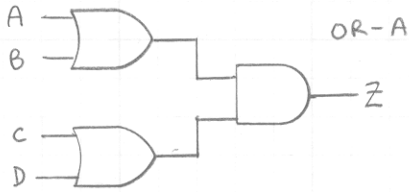


NAND-NAND

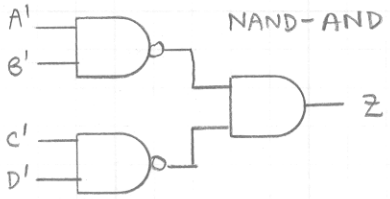


OR-NAND

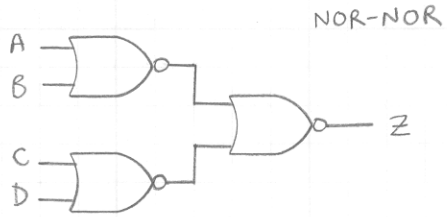
Starting with Product-of-Sums:



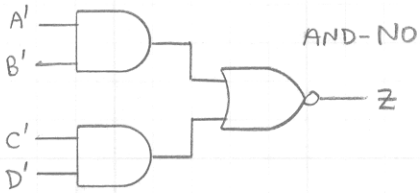
OR-AND



NAND-AND

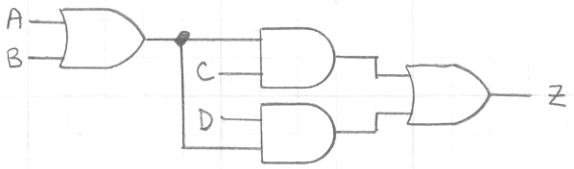


NOR-NOR

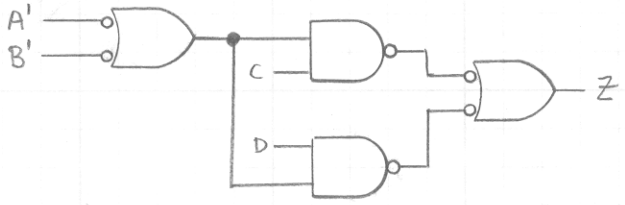


AND-NOR

Ex: Multi-level NAND



OR-AND-OR form



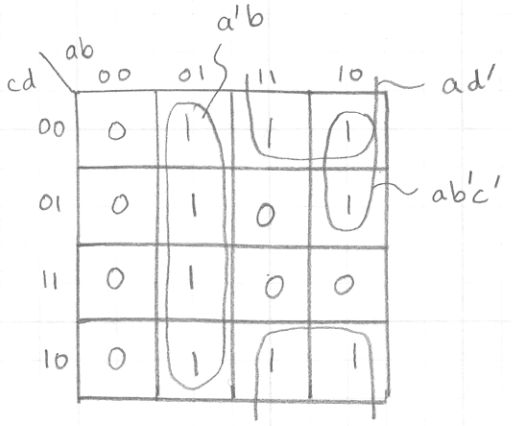
convert using DeMorgan's Law and $X'' = X$ identity

Ex: Find minimum 3 level NOR circuit that implements

$$f = a'b' + abd + acd$$

cd \ ab	00	01	11	10
00	1	0	0	0
01	1	0	1	0
11	1	0	1	1
10	1	0	0	0

f



f'

Want NOR-NOR-NOR circuit,

1) Find AND-OR-AND circuit → find OR-AND circuit then factor using $(X+Y)(X+Z) = X+YZ$

To find OR-AND (Product-of-Sums), use K-map for f':

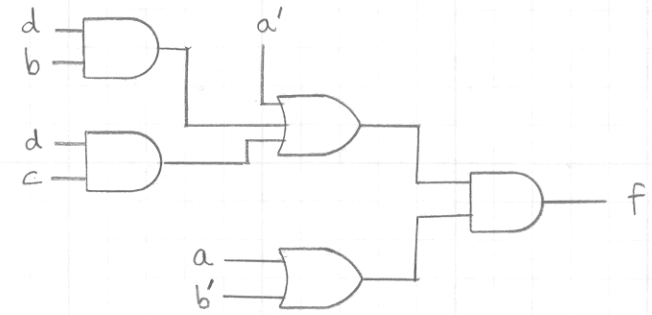
$$f' = a'b + ad' + ab'c'$$

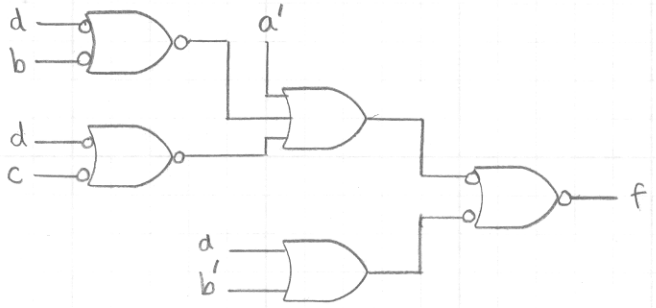
$$f = f'' = (a+b')(a'+d)(a'+\underbrace{b+c}_{Z=b+c})$$

$X = a'$ $Y = d$

$$f = (a+b')(a'+d(b+c)) \quad \text{4 levels}$$

$$= (a+b')(a'+db+dc) \quad \text{3 levels}$$





Replace AND with NOR equivalents (De Morgan's Laws)