

Boolean logic in CMOS



Representations of Boolean logic

- Truth table
- Boolean equation
- Circuit element (gate)



Truth table

- Brute force I/O specification
- Grows exponentially with number of inputs



Boolean algebra

■ Identities

$$x + 0 = x$$

$$x + 1 = 1$$

$$x + x = x$$

$$x + x' = 1$$

$$x'' = x$$

$$x * 1 = x$$

$$x * 0 = 0$$

$$x * x = x$$

$$x * x' = 0$$



Boolean algebra

■ Commutativity

$$x + y = y + x$$

$$x * y = y * x$$

■ Associativity

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

$$x * (y * z) = (x * y) * z$$



Boolean algebra

■ Distributive

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

$$\begin{aligned}x + (y * z) &= (x+y) * (x+z) \\ &= x + xy + xz + yz \\ &= x(1+y) + xz + yz \\ &= x + xz + yz \\ &= x(1+z) + yz \\ &= x + yz\end{aligned}$$

■ De Morgan

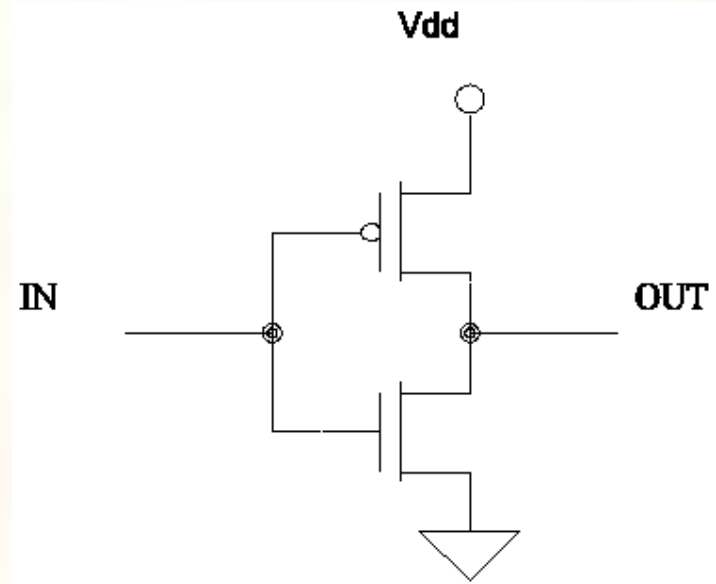
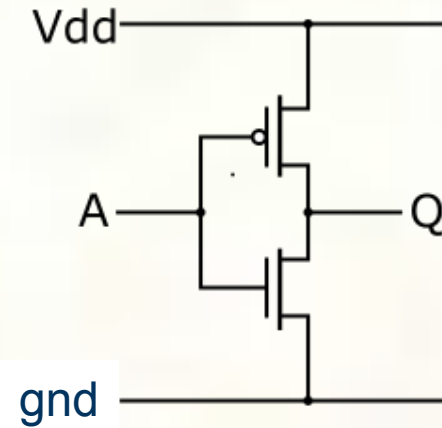
$$(x + y)' = x' * y'$$

$$(x * y)' = x' + y'$$



CMOS gates - NOT

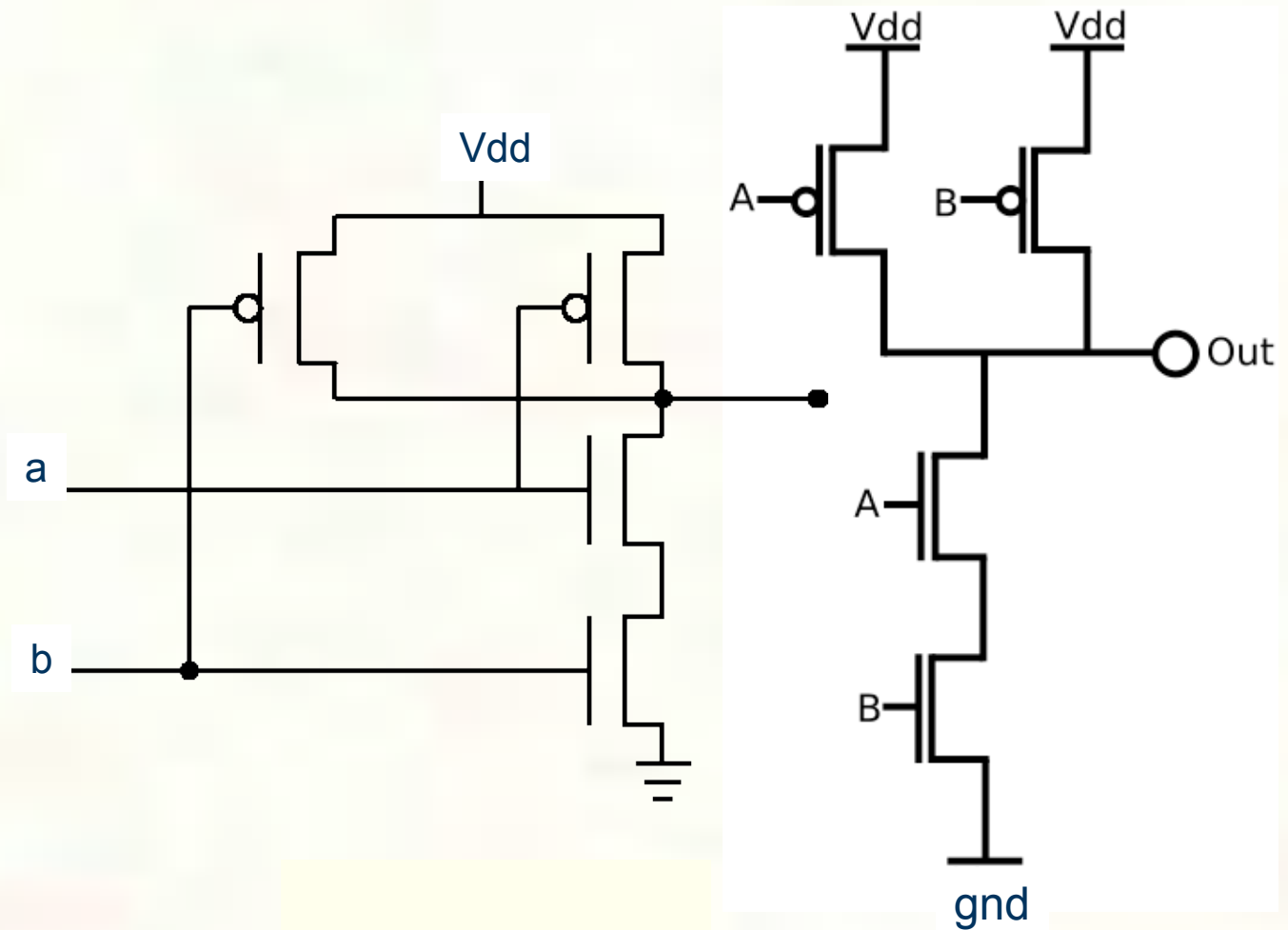
In	Out
0	1
1	0





CMOS gates - NAND

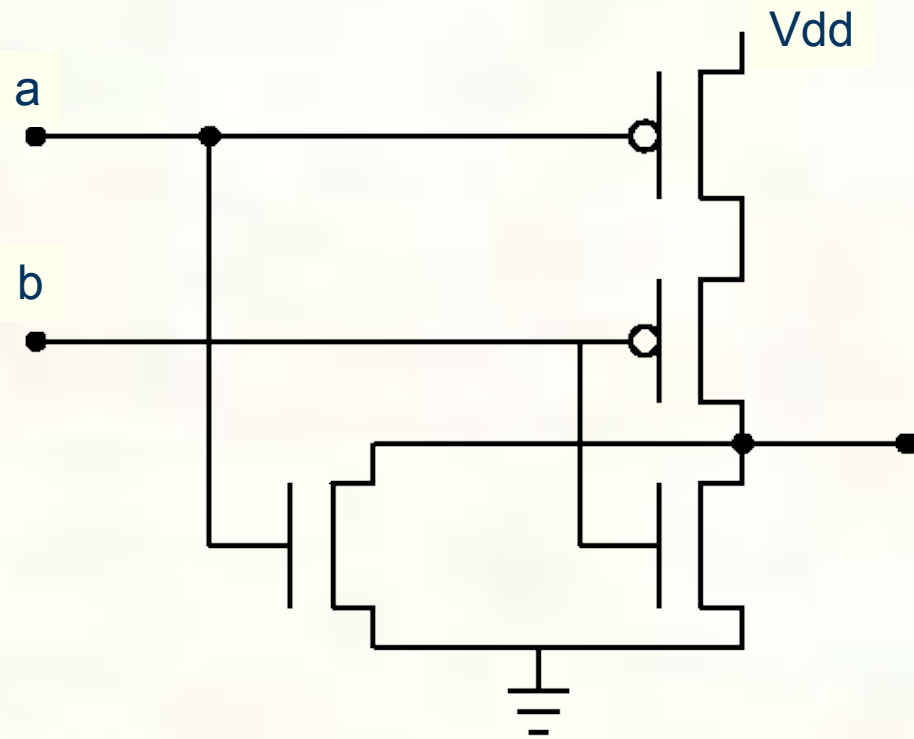
A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0





CMOS gates - NOR

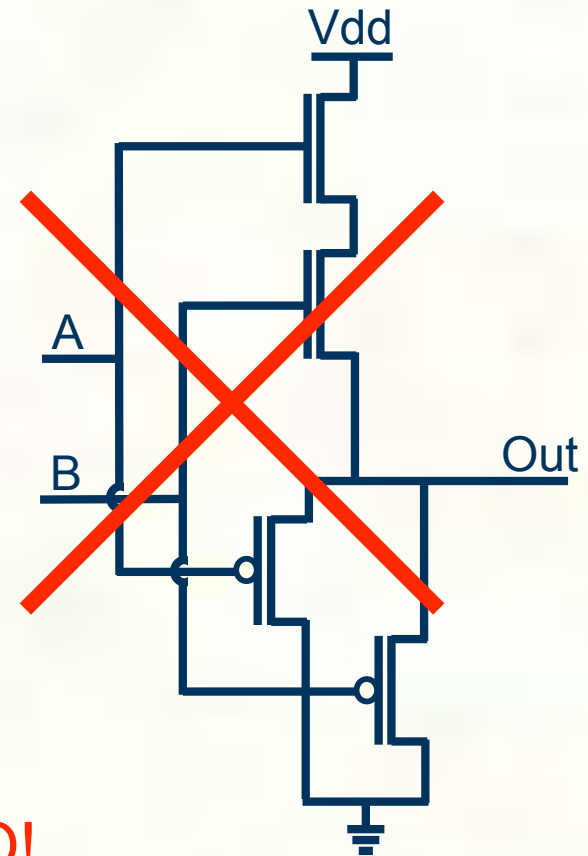
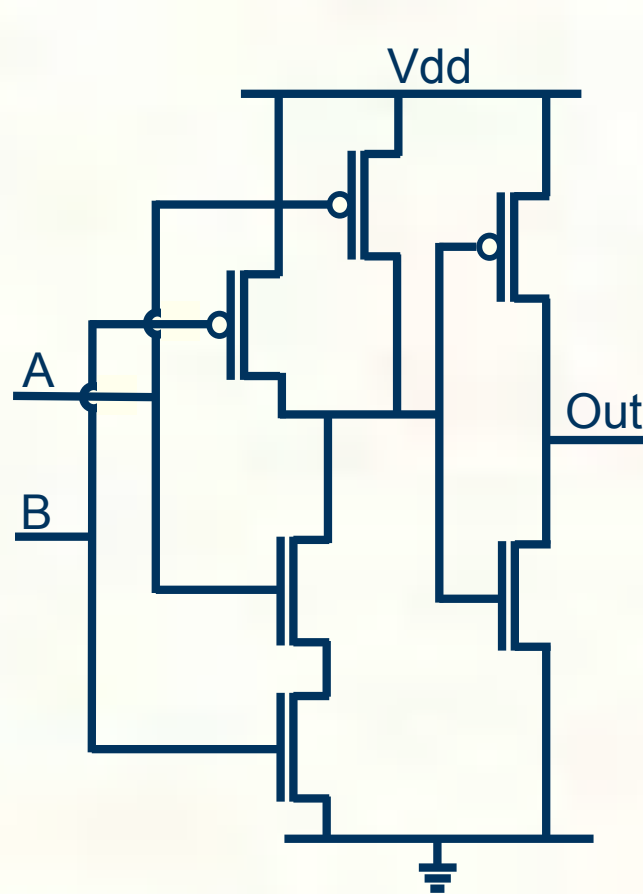
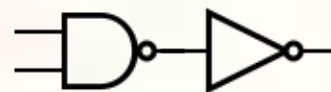
A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0





CMOS gates - AND

A	B	Out
0	0	0
0	1	0
1	0	0
1	1	1



NO!

Logically correct, but violates n to n and p to p rule, passes weak values



CMOS gates - OR

A	B	Out
0	0	0
0	1	1
1	0	1
1	1	1

