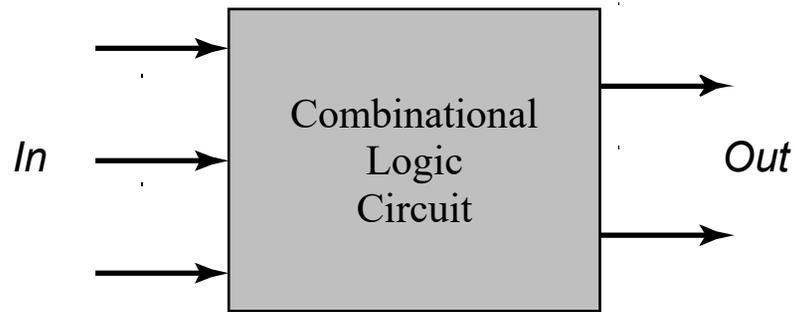
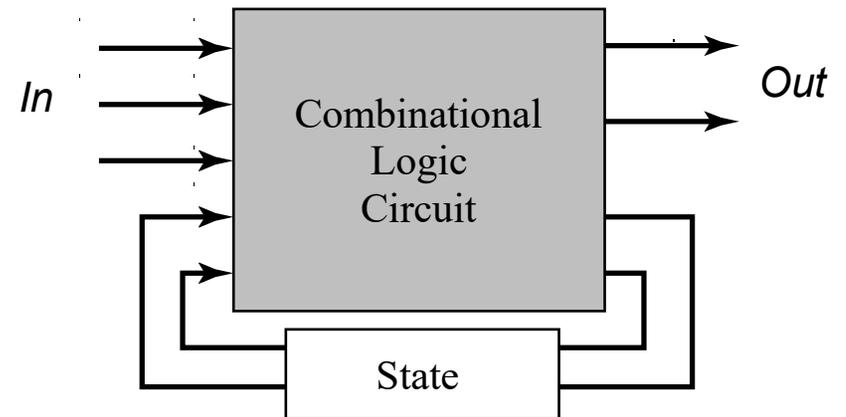


Combinational vs. Sequential Logic



Combinational

$$\text{Output} = f(\text{In})$$



Sequential

$$\text{Output} = f(\text{In}, \text{Previous In})$$

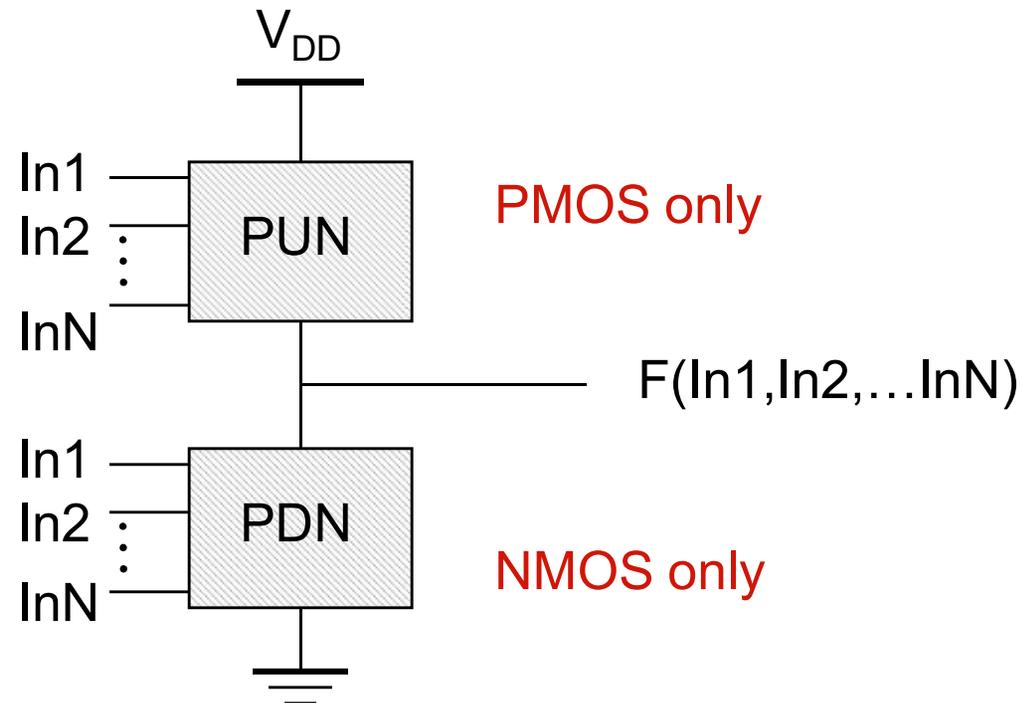
Static CMOS Circuit

At every point in time (except during the switching transients) each **gate output is connected to either V_{DD} or V_{SS}** via a low-resistive path.

The outputs of the gates **assume at all times the value of the Boolean function**, implemented by the circuit (ignoring, once again, the transient effects during switching periods).

This is in contrast to the *dynamic* circuit class, which relies on temporary storage of signal values on the capacitance of high impedance circuit nodes.

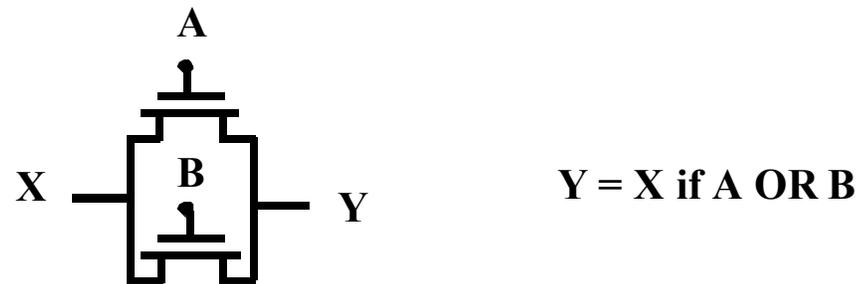
Static Complementary CMOS



PUN and PDN are **dual** logic networks

NMOS Transistors in Series/Parallel Connection

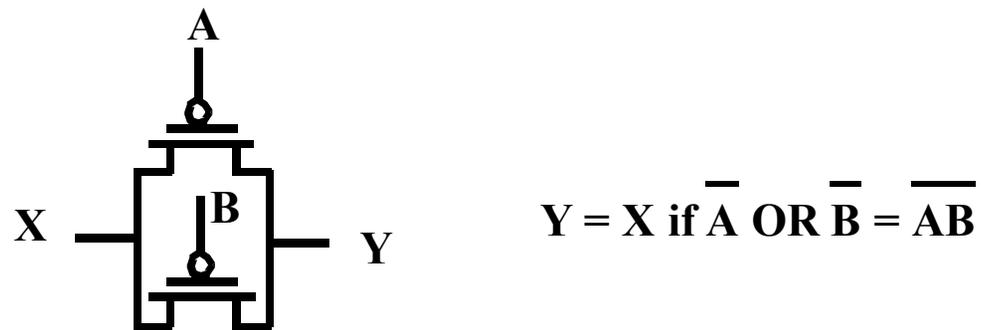
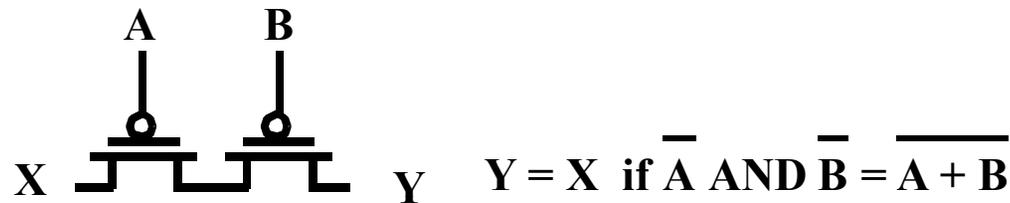
Transistors can be thought as a switch controlled by its gate signal
NMOS switch closes when switch control input is high



NMOS Transistors pass a “strong” 0 but a “weak” 1

PMOS Transistors in Series/Parallel Connection

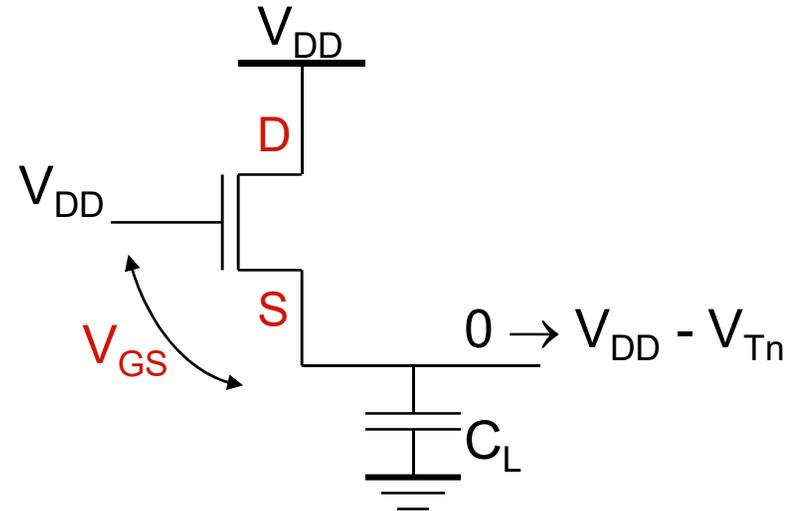
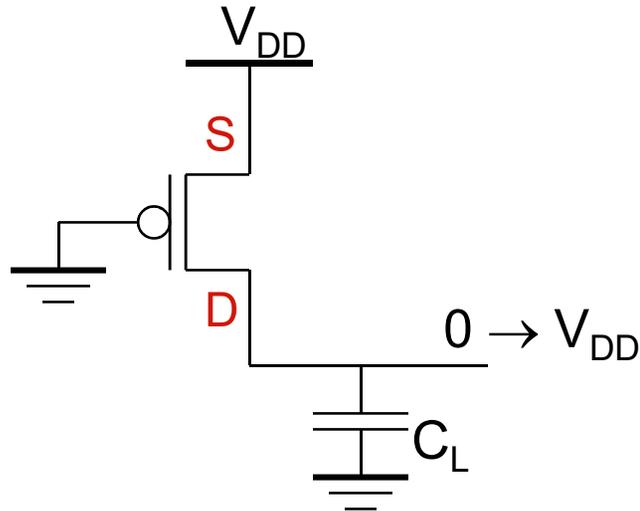
PMOS switch closes when switch control input is low



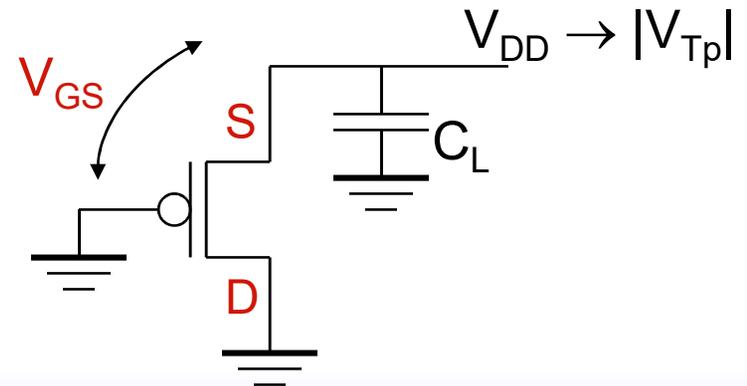
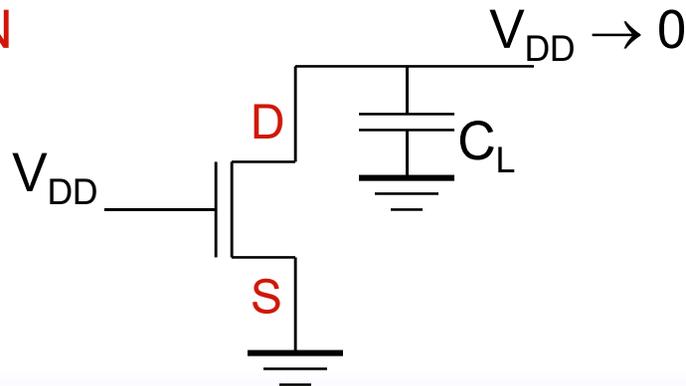
PMOS Transistors pass a “strong” 1 but a “weak” 0

Threshold Drops

PUN



PDN



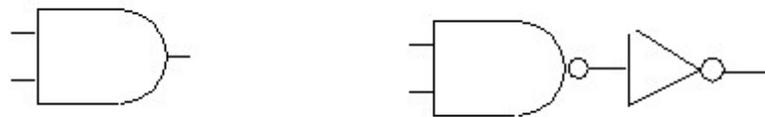
Complementary CMOS Logic Style

- PUP is the DUAL of PDN
(can be shown using DeMorgan's Theorem's)

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

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

- The complementary gate is inverting

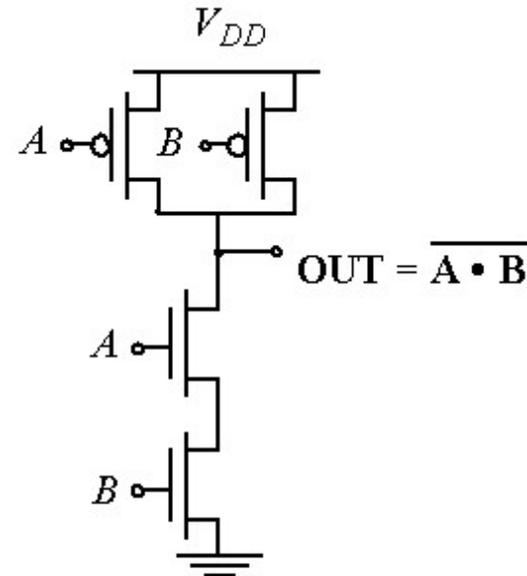


AND = NAND + INV

Example Gate: NAND

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

Truth Table of a 2 input NAND gate

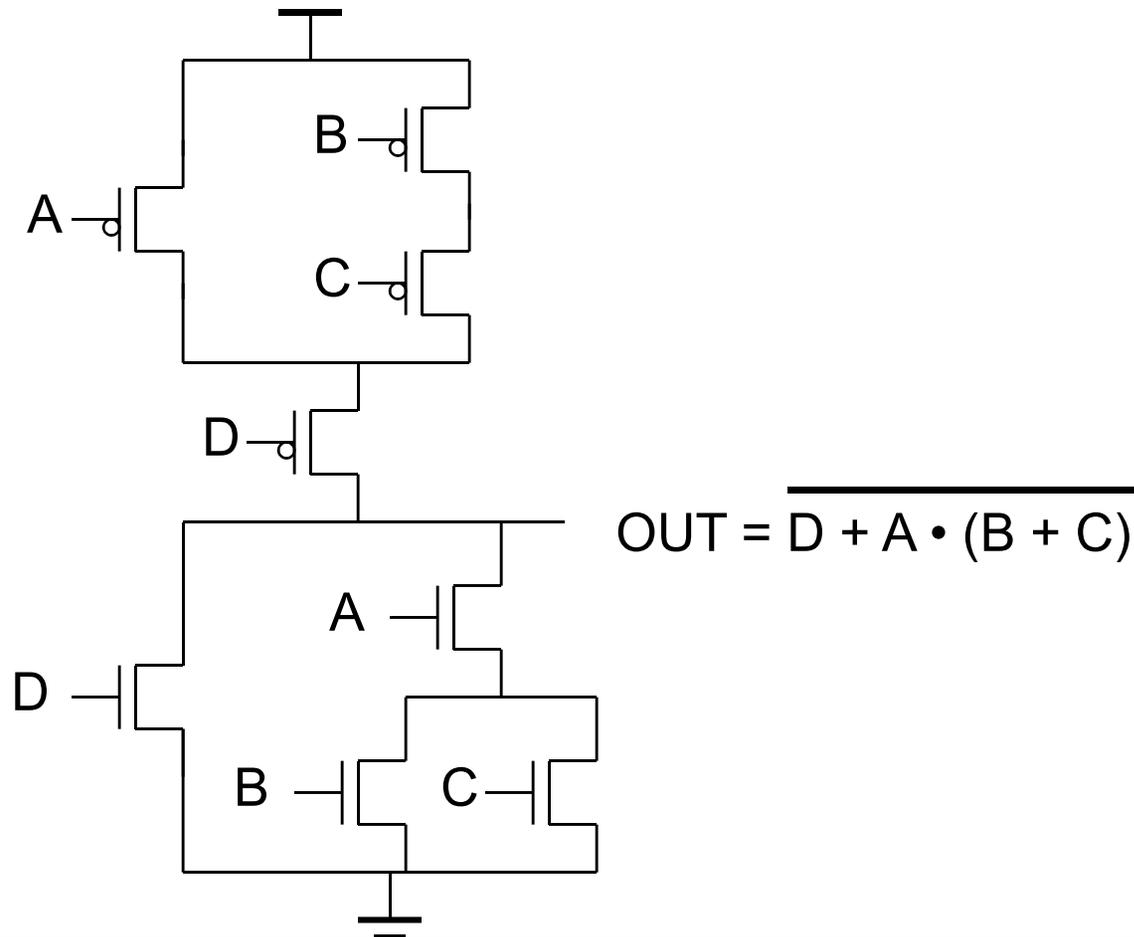


PDN: $G = A B \Rightarrow$ Conduction to GND

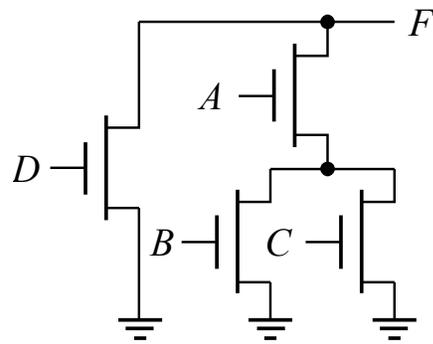
PUN: $F = \overline{A + B} = \overline{AB} \Rightarrow$ Conduction to V_{DD}

$$\overline{G(In_1, In_2, In_3, \dots)} \equiv F(\overline{In_1}, \overline{In_2}, \overline{In_3}, \dots)$$

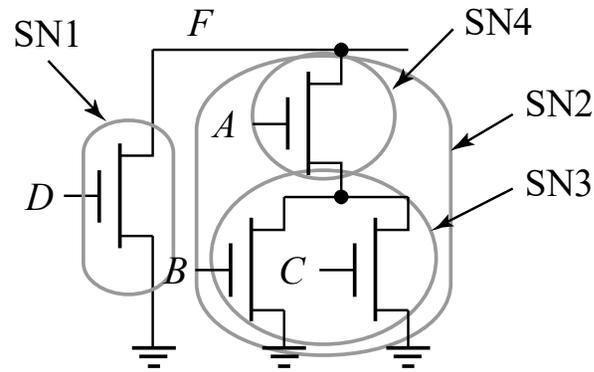
Complex CMOS Gate



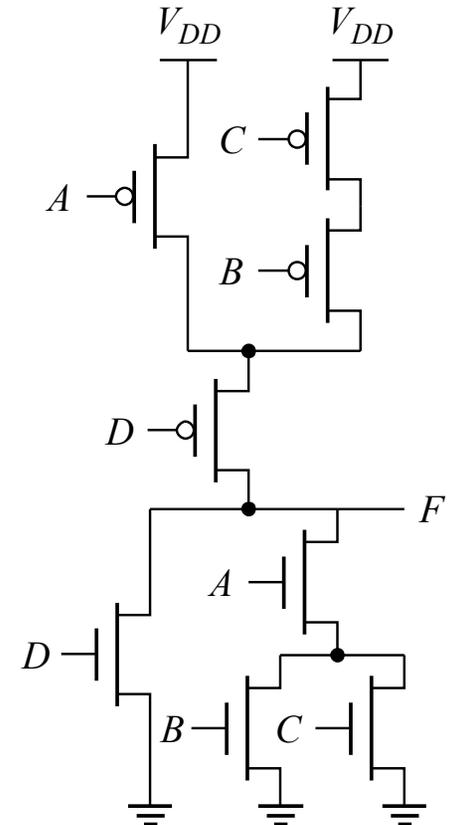
Constructing a Complex Gate



(a) pull-down network

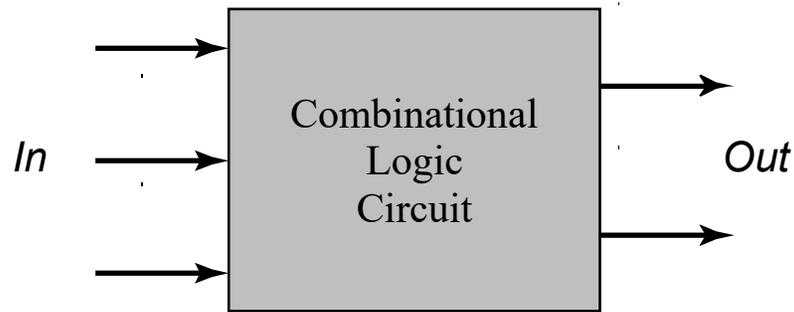


(b) Deriving the pull-up network hierarchically by identifying sub-nets



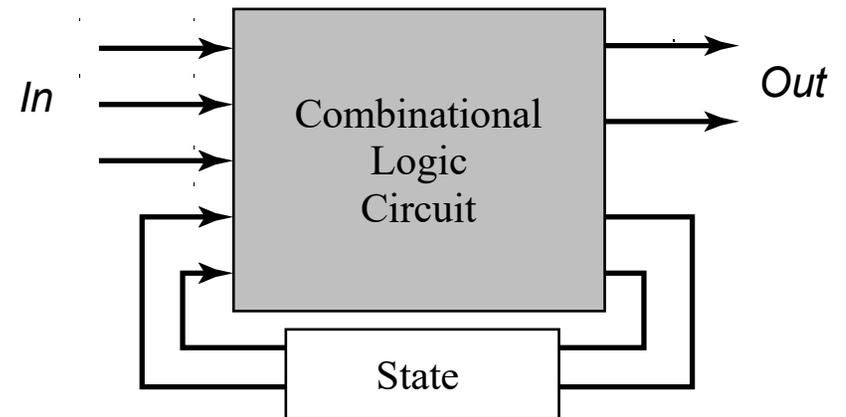
(c) complete gate

Combinational vs. Sequential Logic



Combinational

$$\text{Output} = f(\text{In})$$



Sequential

$$\text{Output} = f(\text{In}, \text{Previous In})$$

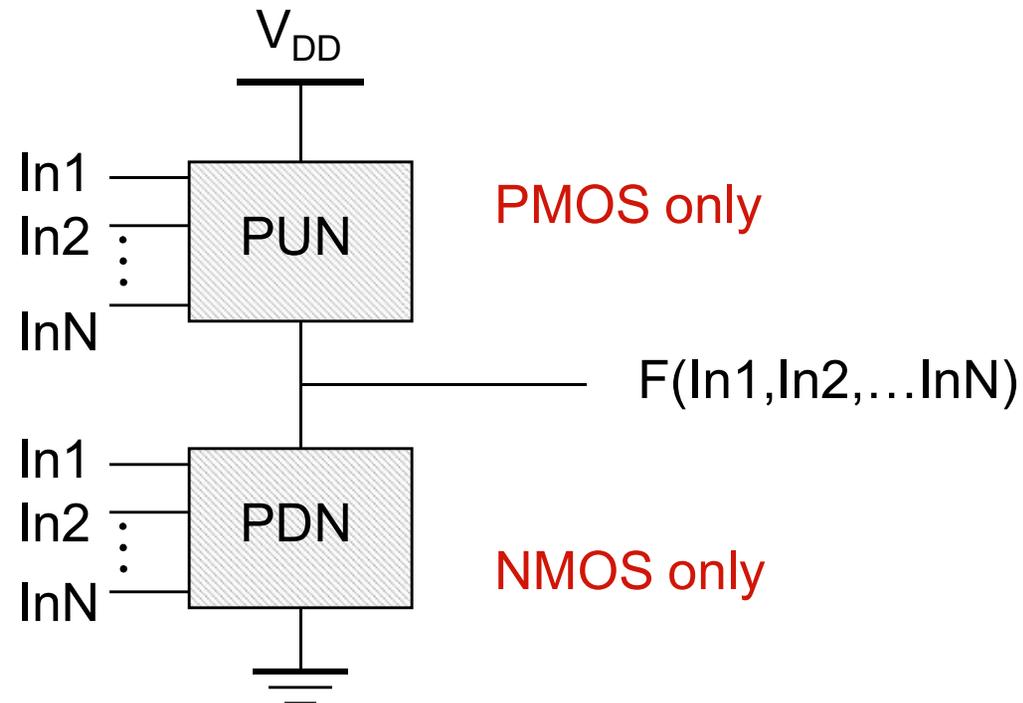
Static CMOS Circuit

At every point in time (except during the switching transients) each **gate output is connected to either V_{DD} or V_{SS}** via a low-resistive path.

The outputs of the gates **assume at all times the value of the Boolean function**, implemented by the circuit (ignoring, once again, the transient effects during switching periods).

This is in contrast to the *dynamic* circuit class, which relies on temporary storage of signal values on the capacitance of high impedance circuit nodes.

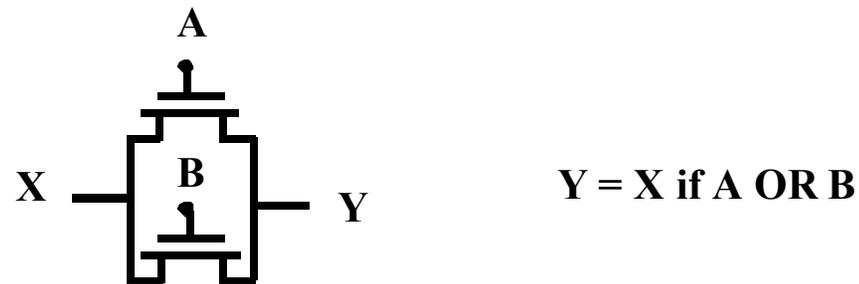
Static Complementary CMOS



PUN and PDN are **dual** logic networks

NMOS Transistors in Series/Parallel Connection

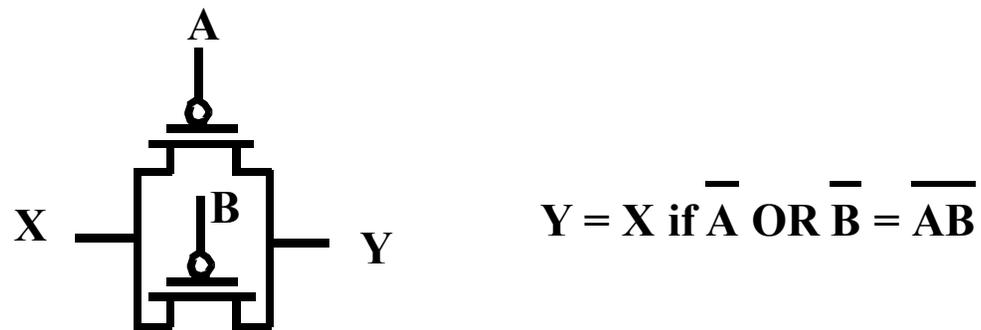
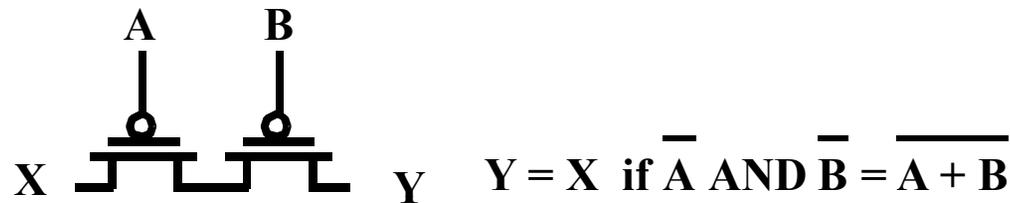
Transistors can be thought as a switch controlled by its gate signal
NMOS switch closes when switch control input is high



NMOS Transistors pass a “strong” 0 but a “weak” 1

PMOS Transistors in Series/Parallel Connection

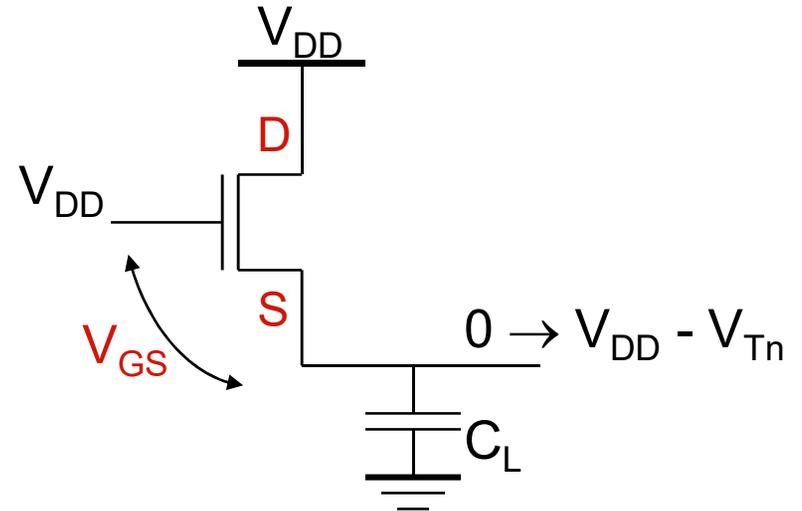
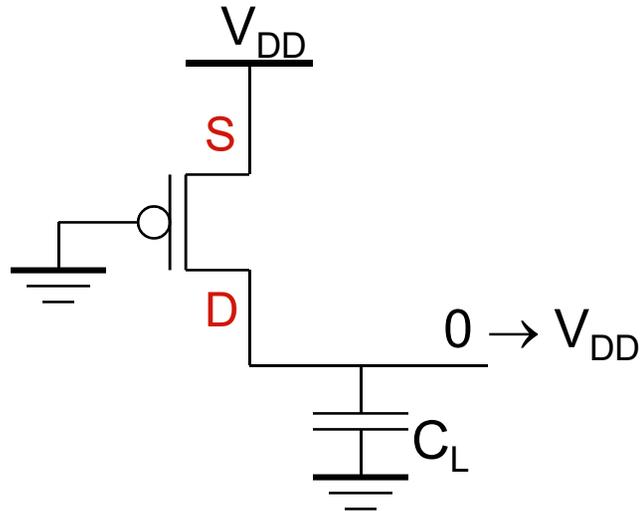
PMOS switch closes when switch control input is low



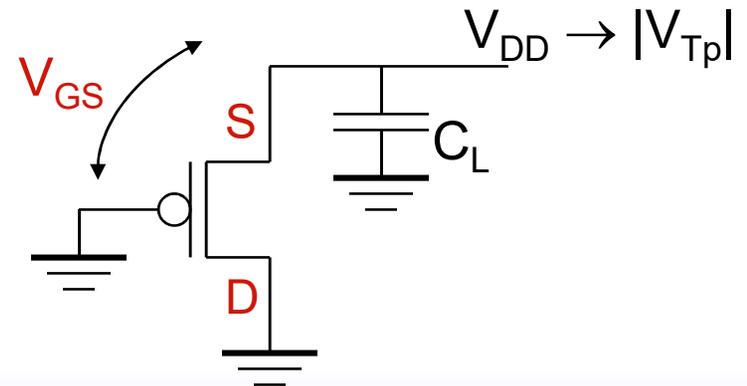
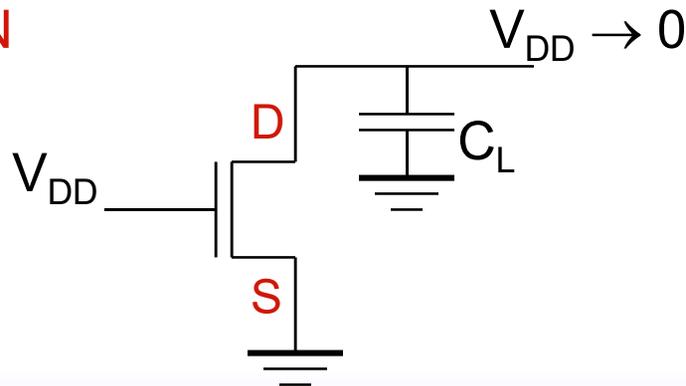
PMOS Transistors pass a “strong” 1 but a “weak” 0

Threshold Drops

PUN



PDN



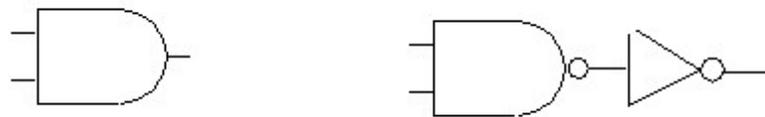
Complementary CMOS Logic Style

- PUP is the DUAL of PDN
(can be shown using DeMorgan's Theorem's)

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

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

- The complementary gate is inverting

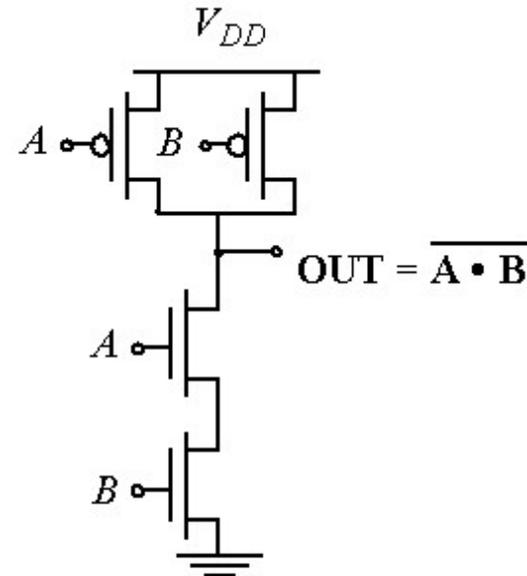


AND = NAND + INV

Example Gate: NAND

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

Truth Table of a 2 input NAND gate

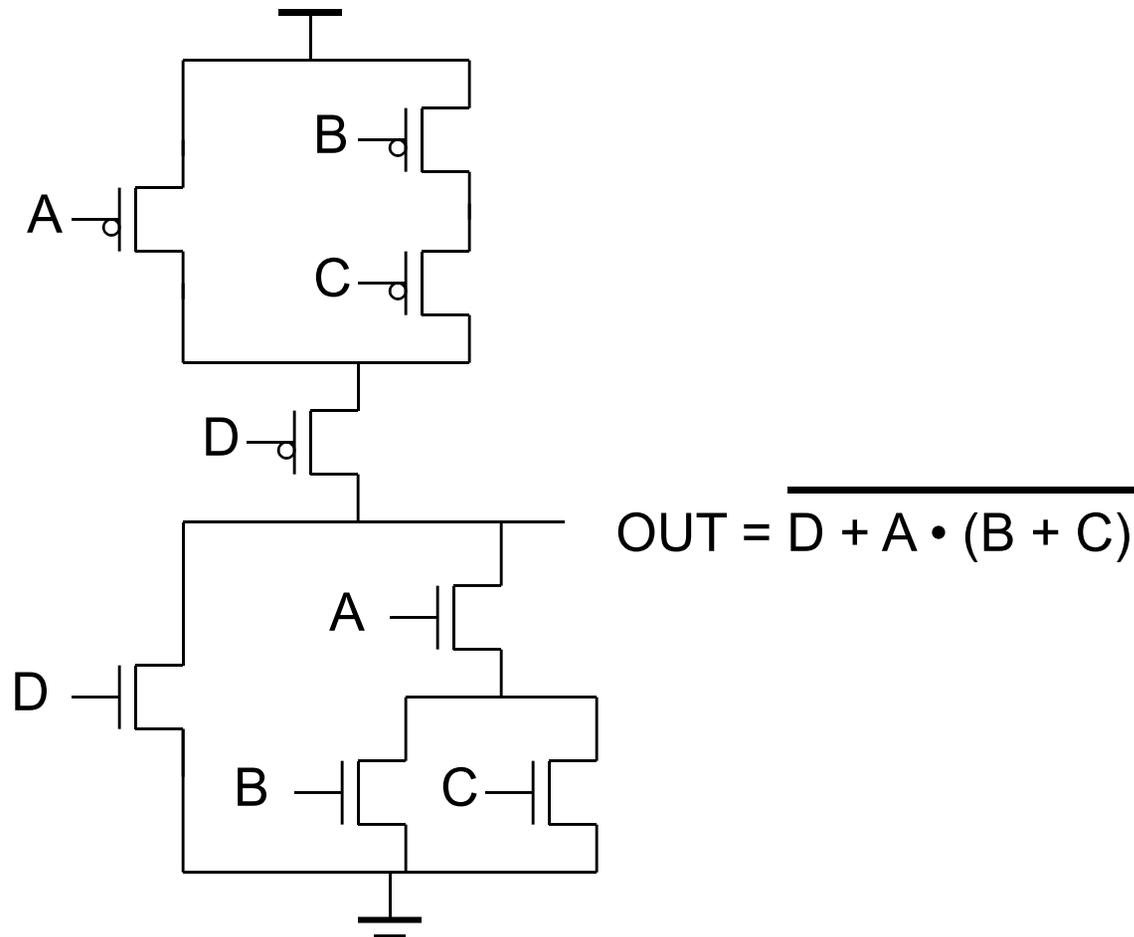


PDN: $G = A B \Rightarrow$ Conduction to GND

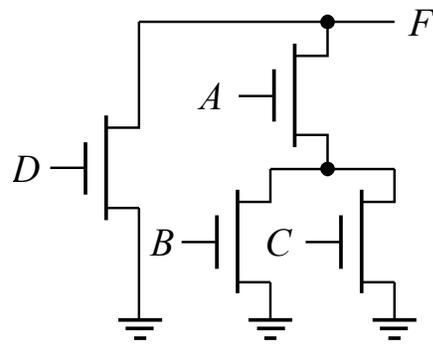
PUN: $F = \overline{A + B} = \overline{AB} \Rightarrow$ Conduction to V_{DD}

$$\overline{G(In_1, In_2, In_3, \dots)} \equiv F(\overline{In_1}, \overline{In_2}, \overline{In_3}, \dots)$$

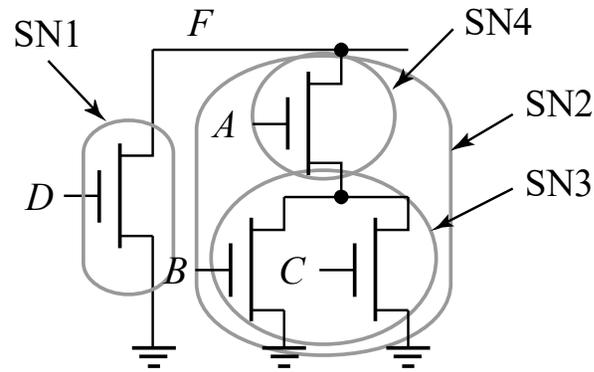
Complex CMOS Gate



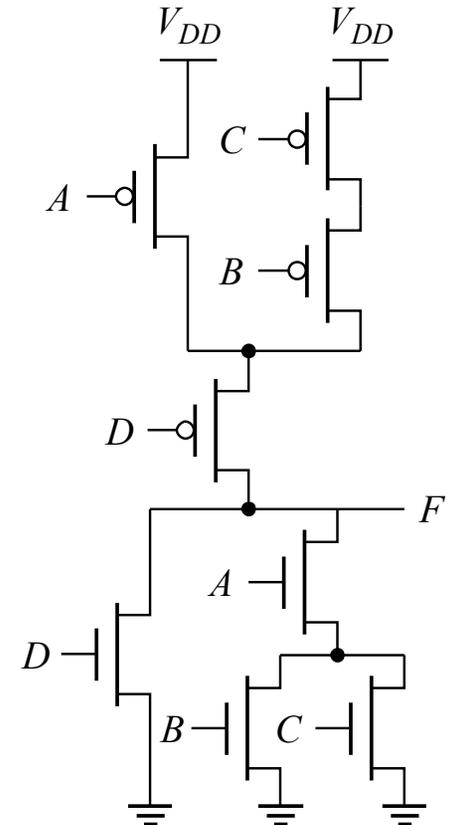
Constructing a Complex Gate



(a) pull-down network



(b) Deriving the pull-up network hierarchically by identifying sub-nets



(c) complete gate