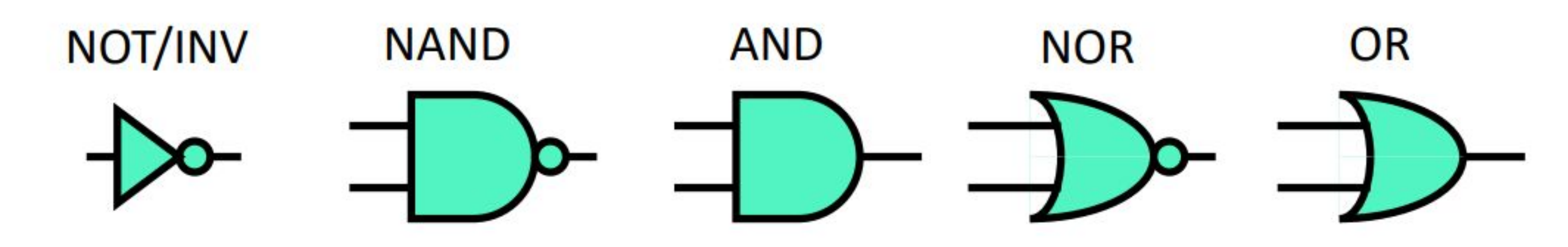
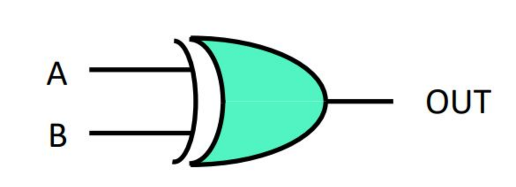
Recitation woooooooooo

•••

10/9

Logic Gates

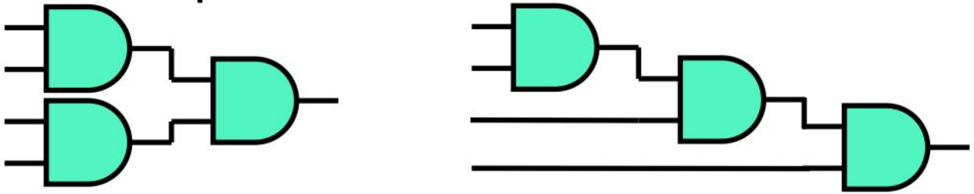




Boolean algebra rules on gates

- Boolean rules still apply to gate-level circuits

- Associative Example:



- De Morgan’s law Example:

- **NOTE**: There are actual differences between these circuits, but “logically” they are the same. More on this later in lecture.

PLAs

- Simplest way to turn truth tables into gate level systems
- Make a truth table, follow easy conversion process
- Downsides: not practical for large inputs, not simplest
    - In hardware will be slower

Practice for truth table -> PLA

- Given a 4 bit number, make a PLA that outputs 1 when the input is a multiple of 7
- Can this be simplified? If so, how?

Muxes

- Way of selecting an output from multiple inputs
- N select bits chooses from 2^N inputs
- Can choose from "groups" of bits by having mini muxes choose the same bits from a group - switch n-bits with n muxes with the same select bits
- Practice question from lecture: how might you implement shifting for a 4 bit number using muxes?

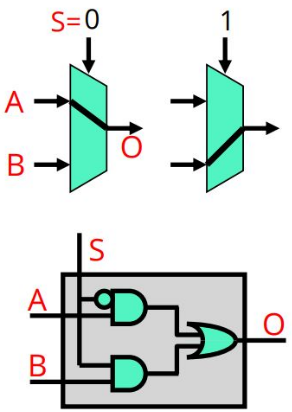
The Multiplexer

Note: selector bits map all "0" to he top input, and increment each input "down"

If you don't want to follow this ordering, label your MUX in the HW

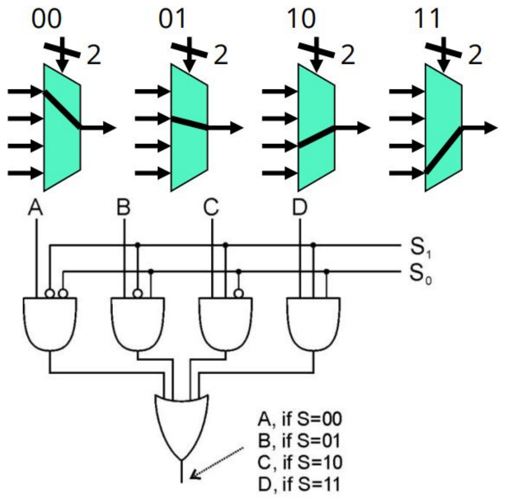
- **Selector/Chooser of signals**
- Shorthand: "Mux"

2-to-1 Mux



Input "S" *selects* A or B to attach to "O" *output*
Acts like an "IF/ELSE" statement

4-to-1 Mux



Simplifying boolean logic

DO NOT MEMORIZE THIS. Understanding WHY these things are true takes less time and is more effective. (Except DeMorgan's law that one is epic)

Name	AND	OR
Identity	1 & A = A	1 | A = 1
Null	0 & A = 0	0 | A = A
Idempotent	A & A = A	A | A = A
Inverse	A & ~A = 0	A | ~A = 1
Commutative	A & B = B & A	A | B = B | A
Associative	(A & B) & C = A & (B & C)	(A | B) | C = A | (B | C)
Distributive	A | (B & C) = (A | B) & (A | C)	A * (B | C) = (A & B) | (A & C)
Absorption	A & (A | B) = A	A | (A & B) = A
De Morgan's	~(A & B) = ~A | ~B	~(A | B) = ~A & ~B