















































## DeMorgan's Law Converting AND to OR (with some help from NOT) Consider the following gate: — Ā•Ē b B ĀB A·B A·B ΑB 0 0 1 1 1 0

1

1

1

Shows that you can write an expression like "not (A or B)" as "(not A) and (not B)". Similarly, "not (A or B)" can be written as "(not A) and (not B)"

Watch this video http://youtu.be/tKnS3s8fOu4

Therefore, you can implement any truth table using only NAND (or NOR) gates

17

## Summary

MOS transistors are used as switches to implement logic functions.

- n-type: connect to GND, turn on (with 1) to pull down to o - p-type: connect to +2.9V, turn on (with 0) to pull up to 1

Basic gates: NOT, NOR, NAND

Logic functions are usually expressed with AND, OR, and NOT

## DeMorgan's Law

0 1 1 0 0

1 0

1 1 0

0

1 0

0 0

Convert AND to OR (and vice versa) by inverting inputs and output Means we may only need to use NOT and AND to implement any logic circuit

CS210