Computer Organization

Functional Completeness

Disjunctive Normal Form

- A Boolean expression that is a sum of products of literals is said to be in **disjunctive normal form**
- A disjunctive normal form (DNF) expression always takes a similar form: $\mathbf{c_1 + c_2 + ... + c_m}$ where each $\mathbf{c_j}$ **for** $\mathbf{j \in \{1, ..., m\}}$ is a **product of literals**
  - Each c_j is called a **term**

Disjunctive Normal Form – cont.

$$\underbrace{\bar{x}y\bar{z}} + \underbrace{xy} + \underbrace{\bar{w}} + \underbrace{y\bar{z}w}$$

- This expression is a sum of four terms
- Each term is a product of literals
- Complements are only applied to individual literals
- No addition operations are applied within terms

Conjunctive Normal Form

- A Boolean expression that is a product of sums of literals is said to be in **conjunctive normal form**
- A conjunctive normal form (CNF) expression always takes a similar form: $\mathbf{d_1 \bullet d_2 \bullet \ldots. \bullet d_m}$ where each $\mathbf{d_j}$ **for** $\mathbf{j \in \{1, \ldots, m\}}$ is a **sum of literals**
    - Each d_j is called a **clause**

Conjunctive Normal Form – cont.

$$\underbrace{(\overline{x} + y + \overline{z})}\underbrace{(x + \overline{y})}\underbrace{(w)}\underbrace{(y + \overline{z} + w)}$$

- This expression is a product of four clauses
- Each term is a sum of literals
- Complements are only applied to individual literals
- No multiplication operations are applied within clauses

Functional Completeness

- A set of operations is **functionally complete** if any Boolean function can be expressed using only operations from the set
- The set **{addition, multiplication, complement}** is functionally complete since any Boolean function can be expressed via disjunctive or conjunctive normal form which only use addition, multiplication, and complement operations

Functional Completeness – cont.

- **Is it possible that only one type of operation is sufficient to compute any Boolean function?**
  - In other words, is it possible for a single Boolean operation to be functionally complete by itself?

- Neither addition, multiplication, nor complement are functionally complete, *but* we can utilize some Boolean trickery to create a functionally complete operation…

De Morgan’s Law

- While none of the aforementioned operations might seem functionally complete on their own, we can easily use NAND and NOR to create a functionally complete system

- Two expressions can be multiplied using only addition and complement by applying De Morgan's law

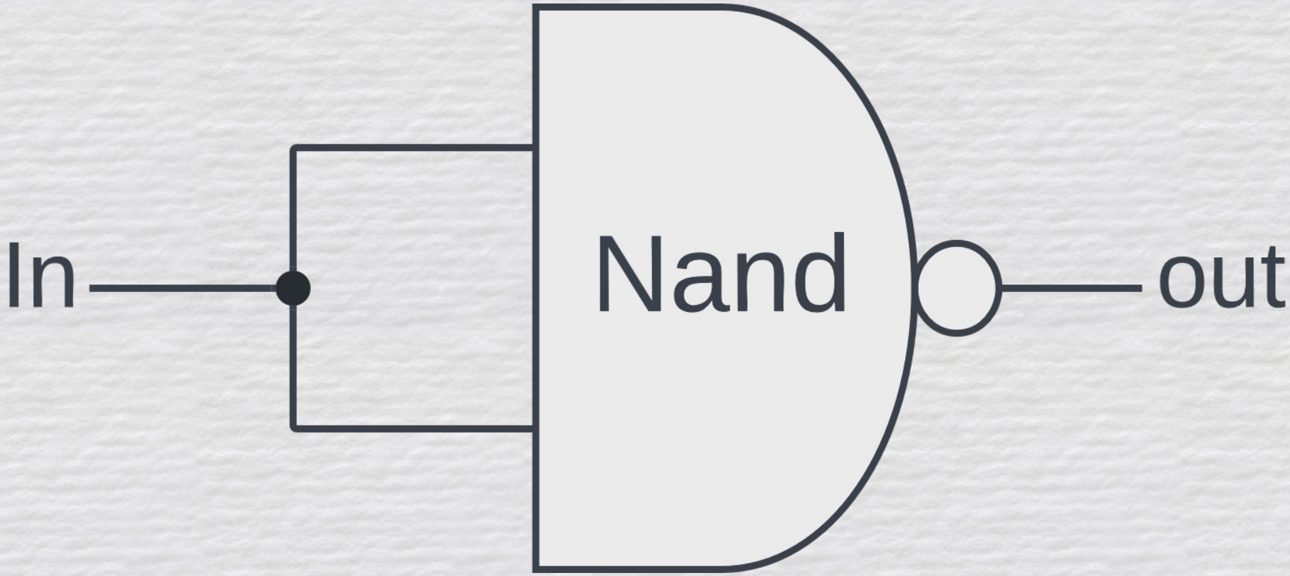
$$xy = \overline{\bar{x} + \bar{y}}$$

Creating Gates with NAND

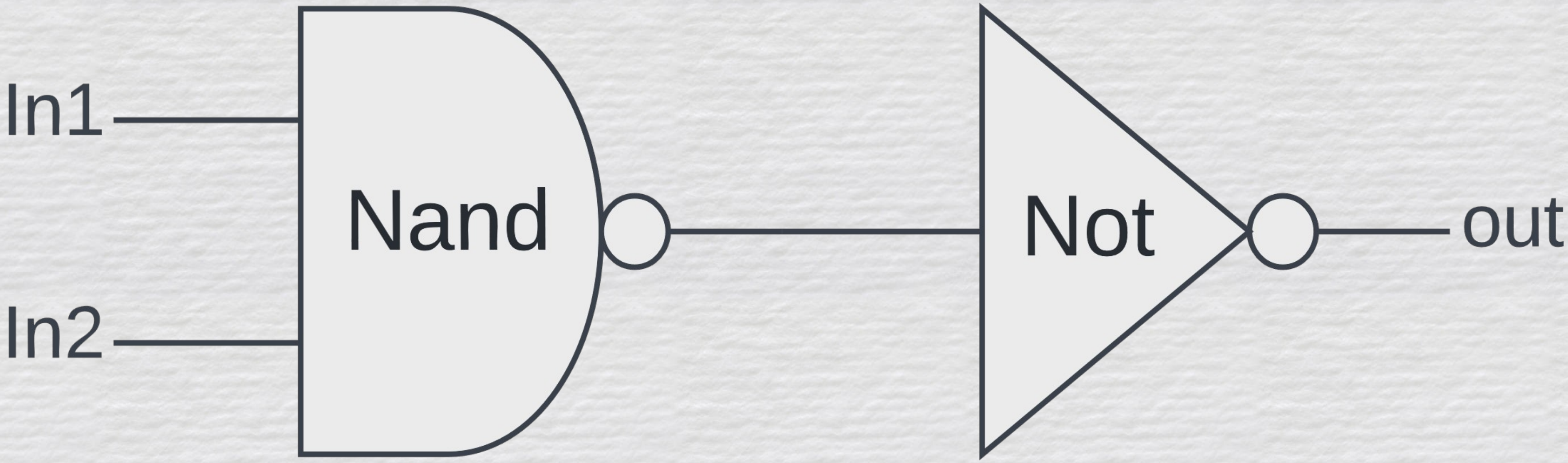
- Since we know that NAND and NOR are functionally complete, we should also know that it is possible to create any possible Boolean expression using just NAND gates

- The easiest way to show this is to create AND, OR, and NOT gates using just NAND gates

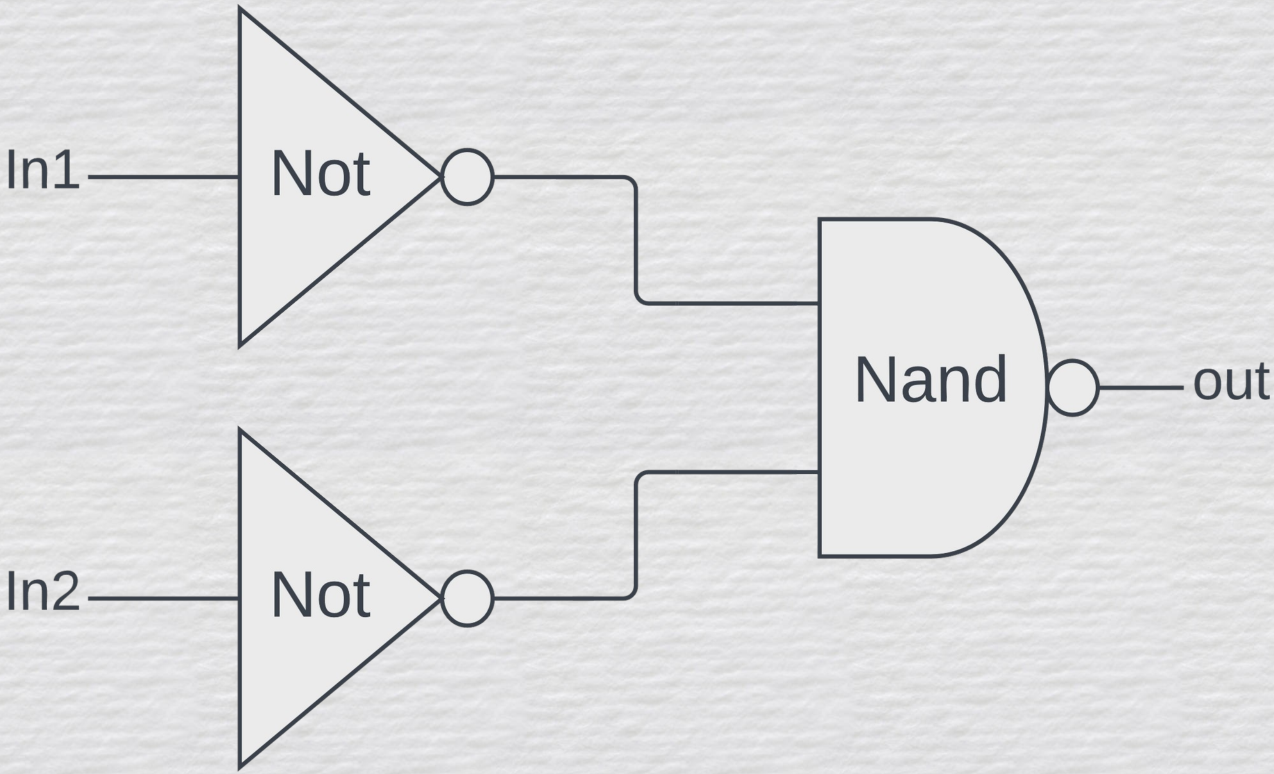
Creating Gates with NAND – NOT Gate



Creating Gates with NAND – AND Gate



Creating Gates with NAND – OR Gate



Creating Gates with NAND – XOR Gate