Architecture: circuits, assembly vs. machine languages

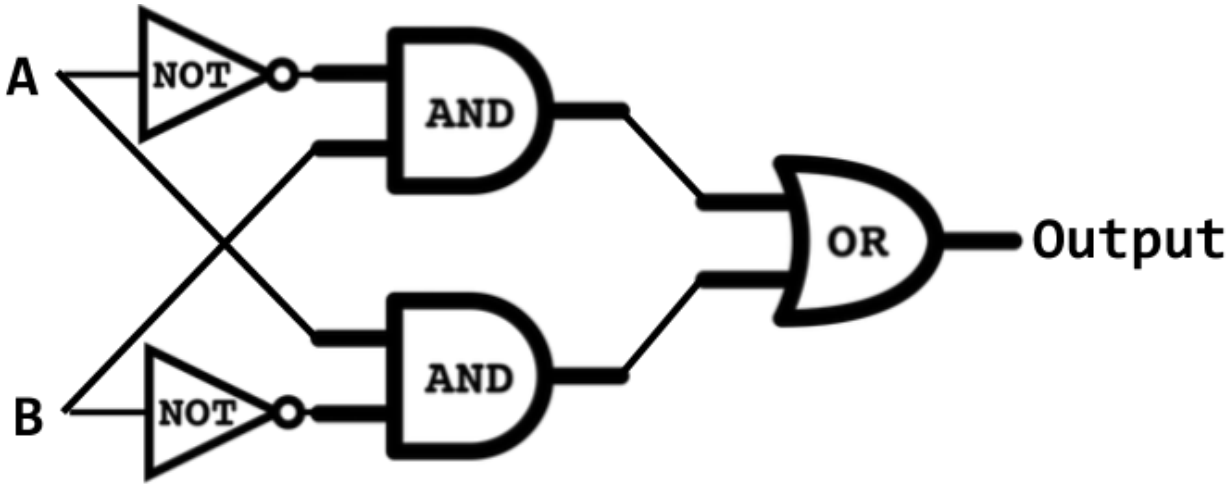
COSC 208, Introduction to Computer Systems, 2022-03-10

1-bit circuits

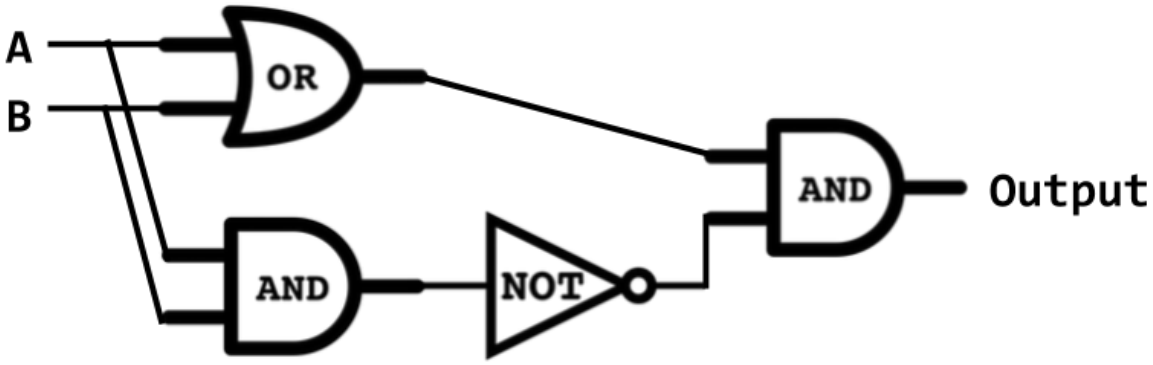
- Connect logic gates to perform a more complex operation
- *Design the truth table: e.g., A != B*

A	B	A != B
0	0	0
0	1	1
1	0	1
1	1	0

- For each row where the output value is 1:
    - Determine how to make each input 1 — e.g., either `A` or `NOT(A)`
    - Conjunct the two subexpressions — e.g., `NOT(A) AND B`
- Create the disjunction of the expressions for each row — e.g., `(NOT(A) AND B) OR (A AND NOT(B))`
- Create a circuit from left to right, starting with the inner-most subexpressions



- *Can we build a circuit that uses fewer gates?*
    - `A XOR B`
    - `(A OR B) AND (NOT (A AND B))`



Q1: *Create a 1-bit circuit for `A <= B` using `AND`, `OR`, `NOT` gates*

```
| A | B | A <= B |
| - | - | ------ |
| 0 | 0 |   1    |
| 0 | 1 |   1    |
| 1 | 0 |   0    |
| 1 | 1 |   1    |
```

```
((NOT A) AND (NOT B)) OR
((NOT A) AND B) OR
(A AND B)
```