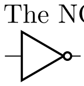
Lecture 3, Sep 13, 2022

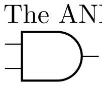
Logic Circuits

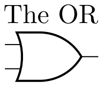
- Transistors can be used as switches
  - Controlled by input x, either connects or disconnects A and B
  - $L(x) = x$
- Two transistors in series forms an AND gate: $L(x_1, x_2) = x_1 \cdot x_2$, or $L(x_1, x_2) = x_1 x_2$
- Two transistors in parallel forms an OR gate: $L(x_1, x_2) = x_1 + x_2$
- A transistor shorting to ground forms a NOT gate: $L(x) = \bar{x}$, or $L(x) = x'$
  - Also referred to the complement of x

Logic Gates

- Using transistors is tedious, so we can represent each of these with gates:
  - The NOT gate:

    
  - The AND gate:

    
  - The OR gate:

    
- Sometimes NOT gates are simplified to just a bubble before the input to a gate
- Example: $S = a\bar{b} + \bar{a}b$

Truth Tables

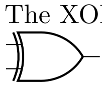
x_1	x_2	AND
0	0	0
0	1	0
1	0	0
1	1	1

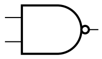
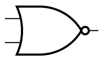
x_1	x_2	OR
0	0	0
0	1	1
1	0	1
1	1	1

- Note AND and OR gates can be extended to an arbitrary number of inputs

Other Gates

- The XOR gate, output is 1 if two inputs are different:

  
  - $L = \bar{x}y + x\bar{y}$
    * When extended to an arbitrary number of inputs, its output is 1 if there are an odd number of 1 inputs

- The NAND gate, output is 0 if both inputs are 1 (i.e. AND + NOT):
  
  - $L = \overline{(xy)}$
    * An AND gate takes 6 transistors, but a NAND gate takes 4 transistors, so this is cheaper to build
    * NAND gates are functionally complete, i.e. you can build any gate with them
- The NOR gate, output is zero if at least one input is 1:
  
  - $L = \overline{(x + y)}$
    * NOR gates are also functionally complete