When designing a circuit, we want to optimize the following:
- Area(size) (minimize)
- Speed (maximize) or delay (minimize)
- Power (minimize)
- Can usually optimize one or two, not all 3

Minimize circuit area
- Reduce the number of gates used to implement a circuit
- Reduce the number of inputs to each gate
    - In general a gate with n inputs requires 2n transistors
- Simplify logic expressions to reduce the number of gates by factoring and cancelling terms

Maximize speed
- affected by
    - Levels of logic
    - Gate type
    - Number of inputs (fan-in) to the gate, usually limited to 4-5 input gates
    - Number of outputs a gate connects to (fan-out)
    - Feature size and implementation technology
- Gates take time for the output to change once the inputs change (aka propagation delay)
- Level of logic = max number of gates on any path from input to output
- Speed: NOT < NAND/NOR < AND/OR < XOR/XNOR

Summary
- All digital circuits can be described using AND, OR, and NOT
- Convention: 1=true / 0=false
- A logic circuit takes some inputs and transforms each possible input combination to desired output values

Boolean Algebra
- Larger logic circuits are too complex to analyze with truth tables or schematics
- Instead, using Boolean algebra to manipulate/simplify the equation
    - Axioms = basis/assumptions used
        - Digital signals/ binary variables: only 1 or 0
        - 3 fundamental logic operations: AND, OR, NOT
    - Theorems = statements derived from axioms

Single variable theorems
- Provide some simplifications for expressions containing:
    - A single variable
    - A single variable and a constant
- Each theorem has a dual
- Each theorem can be proved by writing a truth table for both sides

Waveform diagrams show digital circuit operation over time
- Vertical axis: each signal can be HIGH (1) or LOW (0)
- Horizontal axis: time

Combinational vs Sequential Logic

- Combinational
    - Outputs = f(current inputs)
    - Memory less circuits - outputs only depend on inputs now
- Sequential
    - Outputs = f(current inputs + past inputs)
    - Stateful (having "memory" or storage) - remembering inputs in the past through "**register**"
    - Usually have a controlling signal (aka the "clock") that indicates when the device should "**sample and hold**" the current input

Pulses and Clock Period

- Registers need a a clock pulse edge to trigger
- We can generate pulses at **specific** times that we know the data we want has arrived (at some irregular interval)
- Other registers in our hardware should trigger at a **periodic** interval
    - Alternating HIGH/LOW voltage pulse train
    - 1 cycle is usually measured from rising/positive edge to rising/positive edge
    - Clock frequency (F) = # of cycles per second
    - $F = 1/T$
- The clock period of a digital circuit is set based on the **slowest** delay path from **output** of a register to the **input** of the next

Summary:

- Combinational logic corresponds to operations that manipulate number
- Sequential logic corresponds to variables that can store values for further processing