1. More on gates
    a. Functionally complete sets – a set of gates that can implement any Boolean function
        i. Less gates, easier fabrication
        ii. AND, OR, NOT
            1. XOR takes five gates: $A \oplus B = A * \sim B + \sim A * B$
        iii. AND, NOT
            1. OR requires four gates to implement DeMorgan's law
            2. $A + B = \sim(\sim A * \sim B)$
        iv. OR, NOT
            1. AND requires four gates to implement DeMorgan's law
            2. $A * B = \sim(\sim A + \sim B)$
        v. NAND - $\uparrow$ is the NAND operator.
            1. $\sim A = A \uparrow A$
            2. $A + B = (A \uparrow A) \uparrow (B \uparrow B)$
            3. $A * B = (A \uparrow B) \uparrow (A \uparrow B)$, this requires only 2 NANDS because $(A \uparrow B)$ is used twice
            4. $A \oplus B = (A \uparrow (A \uparrow B)) \uparrow (B \uparrow (A \uparrow B))$, just 4 NANDS needed because $(A \uparrow B)$ is used twice
        vi. NOR - $\downarrow$ is the NOR operator.
            1. $\sim A = A \downarrow A$
            2. $A * B = (A \downarrow A) \downarrow (B \downarrow B)$,
            3. $A \oplus B = ((A \downarrow A) \downarrow (B \downarrow B)) \downarrow (A \downarrow B)$
            4. $A + B = (A \downarrow B) \downarrow (A \downarrow B)$, just 2 NOR gates needed
    b. Implement using transistors

2. Truth tables
    a. Boolean function with *n* variables has 2^n rows
    b. Entire set resembles counting upwards in binary, e.g. 000, 001, 010, 011… with 3 variables
    c. Minterms
        i. Product term in which each of the *n* variables appears once (in either complemented or uncomplemented term)
        ii. Minterm results in a 1 for output of a single cell expression, 0s for all other rows in truth table
        iii. Boolean function can be represented by sum of all minterms for which function is true
            1. Sum of products form (SOP)
    d. Maxterms
        i. Like minterms, variables can only appear once in complemented or uncomplemented form
        ii. Maxterm results in a 0 for the output of a single cell expression, 1s for all other rows in truth table
        iii. The complement of the corresponding row's minterm
        iv. Boolean function can be represented as product of all maxterms for which function is false
            1. Product of sums form (POS)
    e. SOP easier to work with, more natural, but sometimes POS can lead to simpler logic
    f. Term indices correspond to binary concatenation of row variable's truth values
        i. Minterms represented with lower case m, e.g. m_2 for inputs 010
        ii. Maxterms represented with upper case M, e.g. M_5 for inputs 101
    g. Example: 3-variable Boolean function true when A is true, B is true, and C is false
        i. Minterm $m_6 = AB\overline{C}$ (110), maxterm $M_6 = \overline{A} + \overline{B} + C$
        ii. $m_0 = \overline{ABC}$ (000), and $m_7 = ABC$ (111)

3. Synthesizing using Gates
    a. Boolean function of three variables
    b. True when either, but not both, of the first two variables is true
    c. Truth table below:

Index	A	B	C	f(A, B, C)	Minterm	Maxterm
0	0	0	0	0	$m_0 = \overline{A}\,\overline{B}\,\overline{C}$	$M_0 = A + B + C$
1	0	0	1	0	$m_1 = \overline{A}\,\overline{B}C$	$M_1 = A + B + \overline{C}$
2	0	1	0	1	$m_2 = \overline{A}B\overline{C}$	$M_2 = A + \overline{B} + C$
3	0	1	1	1	$m_3 = \overline{A}BC$	$M_3 = A + \overline{B} + \overline{C}$
4	1	0	0	1	$m_4 = A\overline{B}\,\overline{C}$	$M_4 = \overline{A} + B + C$
5	1	0	1	1	$m_5 = A\overline{B}C$	$M_5 = \overline{A} + B + \overline{C}$
6	1	1	0	0	$m_6 = AB\overline{C}$	$M_6 = \overline{A} + \overline{B} + C$
7	1	1	1	0	$m_7 = ABC$	$M_7 = \overline{A} + \overline{B} + \overline{C}$

    d. Sum of products
        i. f = $\overline{A}B\overline{C}$ + $\overline{A}BC$ + $A\overline{B}\,\overline{C}$ + $A\overline{B}C$ = $m_2 + m_3 + m_4 + m_5$
        ii. Can simplify using equivalence laws to reduce number of gates
            1. f = $\overline{A}B\overline{C}$ + $\overline{A}BC$ + $A\overline{B}\,\overline{C}$ + $A\overline{B}C$
            2. = $\overline{A}B(\overline{C} + C)$ + $A\overline{B}(\overline{C} + C)$ by distributive law
            3. = $\overline{A}B$ + $A\overline{B}$ by OR complement law
    e. Product of sums
        i. f = $(A + B + C)$ * $(A + B + \overline{C})$ * $(\overline{A} + \overline{B} + C)$ * $(\overline{A} + \overline{B} + \overline{C})$ = M_0 * M_1 * M_6 * M_7
        ii. Can also simplify using laws of equivalence
            1. f = $(A + B + C)$ * $(A + B + \overline{C})$ * $(\overline{A} + \overline{B} + C)$ * $(\overline{A} + \overline{B} + \overline{C})$
            2. = $\left((A + B)(C + \overline{C})\right) * ((\overline{A} + \overline{B})(C + \overline{C}))$ by distributive law
            3. = (A + B) * $(\overline{A} + \overline{B})$ by OR complement law