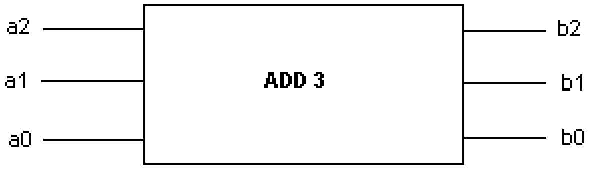
Homework 2 Due: Friday, November 11, 2022

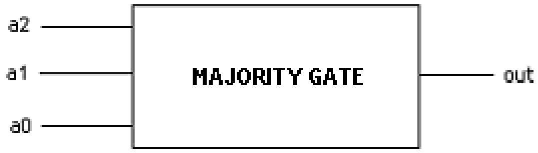
- This homework is due on Friday, November 11, 2022.
- Hand in your completed handwritten homework to me during my lab on Friday from 9:30 am to 12:30 pm.
- Late submission is NOT allowed.
- Copied homework will be awarded 0 marks.

1. The **'ADD 3'** block takes a 3-bit input and gives a 3-bit output. The value at the output is greater than the input by 3. If the output is too big to be represented, then all the output bits are set to 1.



   a. Construct the truth table for the above block.
   b. Simplify b_0, b_1, and b_2 using K-Maps.
   c. Draw the circuit diagram for the above block using AND, OR, and NOT gates.

2. The **'MAJORITY GATE'** block takes a 3-bit input and produces a 1-bit output. The value at the output is equal to 1 if the majority of the inputs are 1s. The output is 0 otherwise.



   a. Build the truth table for the above block.
   b. Simplify **out** using K-Map.
   c. Draw the circuit diagram for the above block using NAND gates only.

3. There is a committee with three people. Each person votes either YES or NO for a proposal. If at least two people vote YES, the proposal is passed. Design a circuit that determines whether a proposal passes.

4. Sometimes the lights are controlled by more than one switch, like in the stairs. The downstairs switch is variable x, and the upstairs switch is the variable y. The lights should get ON when both x and y are ON or both are OFF. Draw the logic circuit of Light (x, y).

5. Design a logic circuit that accepts a 3-bit number and generates an output binary number equal to the square of the input number.

6. Design a circuit that compares two 2-bit numbers, A and B, to check if they are equal. The circuit has one output x, so that $x = 1$ if $A = B$, and $x = 0$ if $A \neq B$.

7. Design a combinational circuit whose input is a 4-bit number and whose output is the 2's complement of the input number.

8. You as a Computer Systems Engineer is given a project to construct a security system for a sensitive military zone. The conditions are as follows:

    a. If it is night & door is opened or laser light is disturbed, alarm should go on.
    b. In day time military is on duty, so there is no risk.

    Develop the logic circuit to meet the above requirements.

9. Design a logic circuit to produce a HIGH output only if the input, represented by a 4-bit binary number, is greater than twelve or less than three. First develop the truth table and then draw the logic diagram.

10. Construct the logic circuit to meet the following requirements:

    A battery-powered lamp in a room is to be operated from two switches, one at the back door and one at the front door. The lamp is to be ON if the front switch is ON and the back switch is OFF, or if the front switch is OFF and the back switch is ON. The lamp is to be OFF if both switches are OFF or if both switches are ON. Let a HIGH output represents the ON condition and a LOW output represents the OFF condition.

11. Design a logic circuit that produces a 1 only when the number of 1's in a set of three input variables A, B, and C is even (i.e. an even parity checker).

12. Obtain the simplified expressions in sum-of-products for the following Boolean functions:

- **a.** $F(x, y, z) = \sum_m (2, 3, 6, 7)$
- **b.** $F(A, B, C, D) = \sum_m (7, 13, 14, 15)$
- **c.** $F(A, B, C, D) = \sum_m (4, 6, 7, 15)$
- **d.** $F(w, x, y, z) = \sum_m (2, 3, 12, 13, 14, 15)$
- **e.** $F(x, y, z) = xy + x'y'z' + x'yz'$
- **f.** $F(A, B, C) = A'B + BC' + B'C'$
- **g.** $F(a, b, c) = a'b' + bc + a'bc'$
- **h.** $F(x, y, z) = xy'z + xyz' + x'yz + xyz$

13. Obtain the simplified expressions in product-of-sums for the following Boolean functions:

- **a.** $F(x, y, z) = \prod_M (0, 1, 4, 5)$
- **b.** $F(A, B, C, D) = \prod_M (0, 1, 2, 3, 4, 10, 11)$
- **c.** $F(w, x, y, z) = \prod_M (1, 3, 5, 7, 13, 15)$

14. Obtain the simplified expressions in (1) sum-of-products, and (2) product-of-sums:

- **a.** $F(x, y, z) = x'z' + y'z' + yz' + xyz$
- **b.** $F(A, B, C, D) = (A + B' + D)(A' + B + D)(C + D) (C' + D')$
- **c.** $F(A, B, C, D) = (A' + B' + D')(A + B' + C')(A' + B + D')(B + C' + D')$
- **d.** $F(A, B, C, D) = (A' + B' + D)(A' + D')(A + B + D')(A + B' + C + D)$
- **e.** $F(v, w, x, y) = w'y + vw' + vw'x + v'w + v'w'y'$

15. Simplify the Boolean function F using the don't-care conditions d, in (1) sum-of-products, and (2) product-of-sums:

- **a.** $F(A, B, C, D) = A'B'D' + A'CD + A'BC$
  $d = A'BC'D + ACD + AB'D'$
- **b.** $F(w, x, y, z) = w'(x'y + x'y' + xyz) + x'z'(y + w)$
  $d = w'x(y'z + yz') + wyz$
- **c.** $F (A, B, C, D) = B'C'D' + BCD' + ABCD'$
  $d = B'CD' + A'BC'D$
- **d.** $F(x, y, z) = y' + x'z'$
  $d = yz + xy$