# ASIC/FPGA Design Spring 2025 Assignment 5

Professor Shabany

---

## Problem 1

**Combinational Question: Implementation and Simplification with ROBDD and Quine-McCluskey**

Consider the following logic function:

$$f = \sum m(0, 1, 2, 3, 4, 6, 8, 10, 11, 15)$$

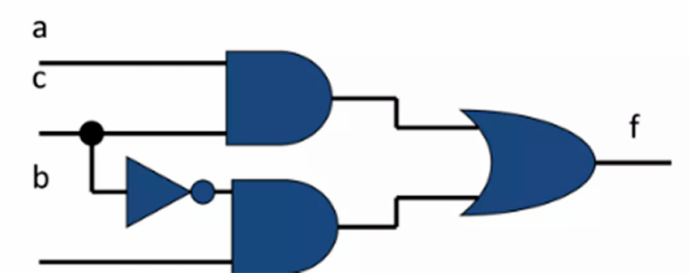
Implement this function in the form of a compressed and optimized ROBDD. Draw the final diagram. Simplify the obtained logic equation using the Quine-McCluskey method. Write a Python or MATLAB code that implements the Quine-McCluskey algorithm. The program should accept input minterms or maxterms and return the simplified expression. The program should take the list of minterms/maxterms and print the minimal logic expression. Test the code with the input of the above function and show that the program output matches your manual analysis.

## Problem 2

**Examination for Combinational Circuit with ROBDD**

A combinational circuit including AND, NOT, and OR gates with inputs a, b, c is given.

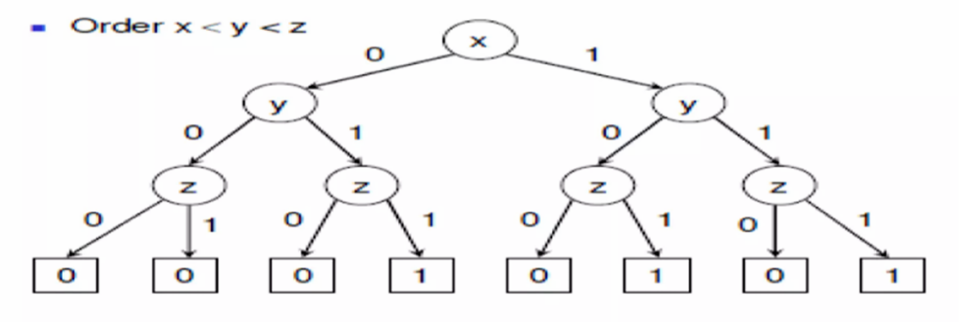
1. For this circuit, consider two variable orders as below and draw the ROBDD for each:
   - First order: (a, b, c)
   - Second order: (b, a, c)
2. Count the number of non-terminal nodes in each ROBDD and specify which order is better from the perspective of implementation cost.
3. Explain how selecting the variable order can be effective in reducing implementation costs in digital circuit designs.




# Problem 3

### Converting OBDD to ROBDD

An OBDD is given with the specified variable order.

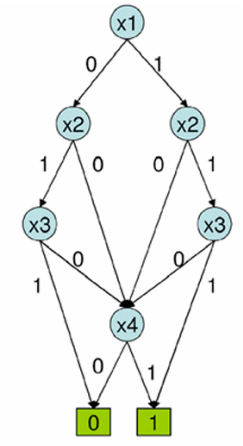



Identify and remove repeated nodes (those with the same input and output). Also, merge paths with the same output as much as possible to reduce the diagram size. Draw the resulting ROBDD and state how many nodes have been saved compared to the initial OBDD.

# Problem 4

### Finding Optimal Variable Order for ROBDD

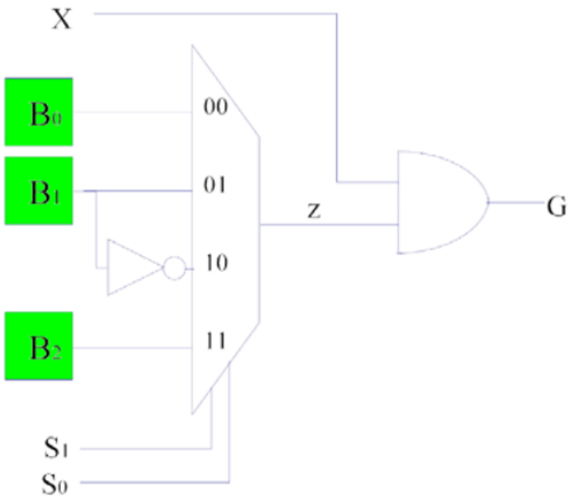
For the given ROBDD, finally examine four meaningful variable orders (by analyzing the function or intuition). For each order, calculate the size of the ROBDD (number of non-terminal nodes). Select the order leading to the smallest graph and explain the reason for its superiority. Only draw the final optimal one; report the size results for the other three in a table or briefly.




# Problem 5

**Mapping Function with SAT Solver**

A configurable logic structure in tree form is at your disposal, and its specified equation is as below.




$$F = (B_0 + S_1 + S_0 + \overline{z})(\overline{B}_0 + S_1 + S_0 + z)(B_1 + S_1 + \overline{S}_0 + \overline{z})(\overline{B}_1 + S_1 + \overline{S}_0 + z)$$
$$(\overline{B}_1 + \overline{S}_1 + S_0 + \overline{z})(B_1 + \overline{S}_1 + S_0 + z)(B_2 + \overline{S}_1 + \overline{S}_0 + \overline{z})(\overline{B}_2 + \overline{S}_1 + \overline{S}_0 + z)$$
$$(\overline{x} + \overline{z} + G)(z + \overline{G})(x + \overline{G})$$

Using a SAT solver, determine which of the target functions below can be mapped in the mentioned structure:

$$f_1 = XS_1S_0 + XS_1\overline{S_0} + X\overline{S_1S_0}$$

$$f_2 = XS_1S_0$$

For each non-mappable function, specify the clauses that cause conflict. In case of possibility of mapping, write the configuration bit values. Briefly explain what limitations or advantages this structure has in mapping functions with specific constraints on inputs.