0.1 Experiment 2 (Implementation of a Boolean Expression)

0.1.1 Aim

In this experiment, your knowledge to minimize the number of gates by using decoder and multiplexer gate.

0.1.2 Problem

A circuit takes five inputs denoted as x_4, x_3, x_2, x_1, and x_0. This circuit has only one output y. The output is determined as shown in the following table.

x_4	x_3	y
0	0	PRIME?(x_2, x_1, x_0)
0	1	EVEN?(x_2, x_1, x_0)
1	0	MAJ?(x_2, x_1, x_0)
1	1	EQ?(x_2, x_1, x_0)

We define PRIME, EVEN, MAJ, and EQ as follows. $x_2x_1x_0$ represents a three bit unsigned number.

$$PRIME?(x_2, x_1, x_0) = \begin{cases} 1 & x_2x_1x_0 \text{ is prime} \\ 0 & \text{otherwise} \end{cases}$$

$$EVEN?(x_2, x_1, x_0) = \begin{cases} 1 & x_2x_1x_0 \text{ is even} \\ 0 & \text{otherwise} \end{cases}$$

$$MAJ?(x_2, x_1, x_0) = \begin{cases} 1 & \text{Most of } x_2, x_1, x_0 \text{ are } 1 \\ 0 & \text{otherwise} \end{cases}$$

$$EQ?(x_2, x_1, x_0) = \begin{cases} 1 & x_2 = x_1 = x_0 \\ 0 & \text{otherwise} \end{cases}$$

You are only allowed to use **one 2x4 Decoder, one 8x1 Multiplexer, at most 2 binary logic gates (AND, OR, XOR, XNOR, NAND, NOR) and infinite number of NOT gates** to complete the design of this circuit.

0.1.3 Preliminary Work

Before the experiment, you should prepare following materials:

1. Fill the truth table.
2. Draw the circuit of your design.
3. Write a verilog code for the circuit drawn in the previous step. Verilog code should have two components. First, you have to write behavioural level verilog code which implements the functionality of multiplexer and/or decoder. Then, you have to write the gate level verilog code (source.v) for your circuitry which uses the implemented components and additional **built-in** gates (i.e. *AND*, *OR*, *NAND*, *NOR*, *XOR*, or *XNOR*).

4. Write the verilog code for the testbench (testbench.v) in order to test all possible input combinations.

Then, submit the your code, and your report under the name `<StudentID1>_<StudentID2>_PRE2.zip` through Moodle.