6.3 Experiment 3 (Implementation of a Boolean Expression)

6.3.1 Aim

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

6.3.2 Problem

A circuit takes five inputs denoted as y_1, y_0, x_2, x_1, and x_0. This circuit has only one output z. The output is determined as shown in the following table.

y_1	y_0	z
0	0	PRIME(x_2, x_1, x_0)
0	1	ODD(x_2, x_1, x_0)
1	0	FIBONACCI(x_2, x_1, x_0)
1	1	SQUAREROOT(x_2, x_1, x_0)

We define PRIME, ODD, FIBONACCI, and SQUAREROOT as follows. $x_2x_1x_0$ represents a three bit unsigned number.

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

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

$$FIBONACCI?(x_2, x_1, x_0) = \begin{cases} 1 & x_2x_1x_0 \text{ is an element of Fibonacci Sequence} \\ 0 & \text{otherwise} \end{cases}$$

$$SQUAREROOT?(x_2, x_1, x_0) = \begin{cases} 1 & \text{The square root of } x_2x_1x_0 \text{ is an integer} \\ 0 & \text{otherwise} \end{cases}$$

In the implementation, you are only allowed to use one Decoder, one Multiplexer, two binary logic gates (standard gates like AND, OR, XOR, XNOR, NAND, NOR) and NOT gates (as many as needed). Multiplexer can have at most 8 input (i.e. 8 way mux), and decoder can have at most 3 input (i.e. 3 to 8 decoder).

Note: For example, $x_2x_1x_0$ = 010 (2) occurs in Fibonacci Sequence. Therefore for y_1y_0 = 10, z should be 1.

Note: For example, The square root of $x_2x_1x_0$ = 100 (4) is 2 and it is an integer. Therefore for y_1y_0 = 11, z should be 1.

6.3.3 Preliminary Work

Before the experiment, you should prepare following materials:

1. Fill the truth table.

2. Analyze the expression and make your design using the allowed components to implement the expression. Then, draw the circuit of your design. If you will use other components than the allowed ones or more than the allowed numbers, you will lose point.

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. Secondly, you have to write the gate level verilog code 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 in order to test all possible input combinations.

Then, follow the steps described in Section 3.1 to submit your preliminary work.

6.3.4 Lab Work

1. Load your submitted code given by the assistants and demonstrate that the circuit implements the truth table of the boolean expression assigned to your lab section by using the testbench waveforms you have generated.

2. Do the same procedure as in preliminary work for the problem definition given in the lab section and demonstrate your implementation.