EE M16 and CS M51A Winter 2019 Section 2

Logic Design of Digital Systems

Dr. Yutao He

Project #1 - Orientation of Verilog and ISE

Due: Wednesday Februry 6^{th}, 2019

Name: Liu Joshua
Last First

Student ID: 105136031

Discussion: 2B TA: Guanrong Hou

Date: 1/31/19

Result	
Correctness	
Creativity	
Report	
Total Score	

Project #1 Orientation of Verilog and ISE Project Requirement

Dr. Yutao He

1 Objectives

The first project is intended to get you acquainted with Verilog and ISE software by implementing a simple combinational circuit, and familiar with the typical work-flow of using the industry-strength Computer-Aided Design(CAD) tool in designing digital systems.

2 Project Description

In this project, you will use the Xilinx ISE software and Verilog to implement the circuit that is specified in Figure 2.12 on page 32 of the textbook. It consists of two basic steps:

(1) Verilog Coding

You should use the *Text Editor* in ISE to write the Verilog code that describes the function of the circuit and then to synthesize it.

(2) Function (Behavior) Simulation

After the Verilog code passes compilation, simulation is followed to test if your implementation performs the specified function (that is, to verify its correctness). In order to do so, you must apply each combination of input values to the circuit and obtain the corresponding output values. This can be entered as a separate test bench file. *ISim* is a simulator in the ISE serves this purpose. The *Waveform Window* can be used to display the results in the waveform (i.e., a Timing Diagram). If the implementation is not correct, you have to debug it until it works properly.

3 Report Outline

You are required to submit a report that provides complete documentation of your project. As in all technical writing, its purpose is to communicate your work with your colleagues in an efficient and professional way. As a result, it must be clear, concise and complete and must contain the following parts:

(1) *Title Page*

It is provided and you just need to fill in your information in the blanks.

(2) *Project Requirement*

It is this handout.

(3) *The Function of The Circuit*

The function of the circuit has already been specified in the textbook so you do not need to design from the scratch. But you must include in your report its canonical switching expression and the corresponding schematic diagram.

(4) *The Verilog file*

The Verilog file (with extension .v) you write is the implementation of the circuit. You should include it in your report with your name and student ID on it.

(5) *The Simulation Result*

You have to demonstrate that your implementation works as specified by showing the simulation result from Vivado. **Screendump** the Timing Diagram that consists of waveforms of the inputs and the output. Please provide enough information on the timing diagram so that one of your colleagues who doesn't know anything about your project could understand which function you are trying to implement and evaluate it. **A timing diagram without any explanation is subject to penalty**.

(6) *The Summary*

This writeup should be short and state at least one problem you encounter during the implementation and the workaround you come up with, or any other comments you would like to make.

4 ISE Software

The Xilnx ISE Software provides a free version, called *WebPack* for Windows and Linux operating systems. It requires registration for downloading. Its website is: https://www.xilinx.com/products/design-tools/ise-design-suite/ise-webpack.html

5 Project Submission

By February 4, a sumbmission link will be set up on the course website. You should submit one zipped file named 1234567.zip, where 1234567 is your student ID. The zipped file must include the following three files:

1. The pdf file of your report. It must be named with your student ID. As a result, your report should be called *1234567.pdf*;

2. The Verilog file of your circuit implementation. It must be named as: *eem16_proj1.v*;

3. The testbench file. It must be named as *eem16_proj1_tb.v*.

6 Project Deadline

The report is due at midnight (11:59:59pm) February 6 (Wednesday), 2019. Late submission is subject to penalty.

(1) Title Page
(Above)

(2) Project Requirement
(Above)

(3) The Function of The Circuit

The function of the circuit has already been specified in the textbook so you do not need to design from the scratch. But you must include in your report its canonical switching expression and the corresponding schematic diagram.

The function of the circuit we need to implement is a two-level AND-OR gate, where the canonical switching expression is:

$$\mathrm{z} = \sum m(1,2,6)$$

Minterms expressed as $\mathrm{F(A, B, C)}$:

$$\begin{aligned} \mathrm{F(A, B, C)} &= \sum m(1,2,6) \\ &= \mathrm{m1 + m2 + m6} \qquad \leftarrow \text{m-notation} \\ &= \mathrm{A'B'C + A'BC' + ABC'} \end{aligned}$$

In maxterms, $\mathrm{F(A, B, C)}$:

$$\begin{aligned} \mathrm{F(A, B, C)} &= \prod M(0,3,4,5,7) \\ &= \mathrm{M0 \cdot M3 \cdot M4 \cdot M5 \cdot M7} \qquad \leftarrow \text{M-notation} \\ &= \mathrm{(A' + B' + C') \cdot (A' + B + C) \cdot (A + B' + C') \cdot (A + B' + C) \cdot} \\ &\quad \mathrm{(A + B + C)} \end{aligned}$$

Below is a truth table for the circuit:

Dec	A	B	C	F(A, B, C) = z
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

This is the schematic we are modeling, taking from the textbook:

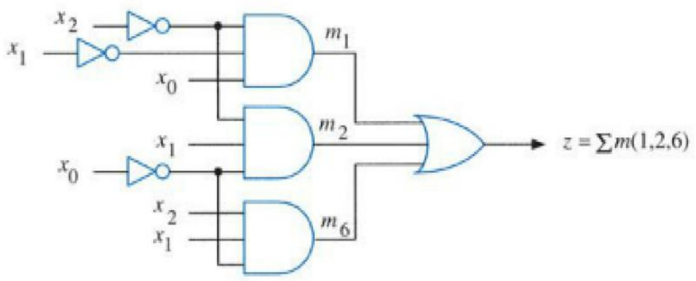
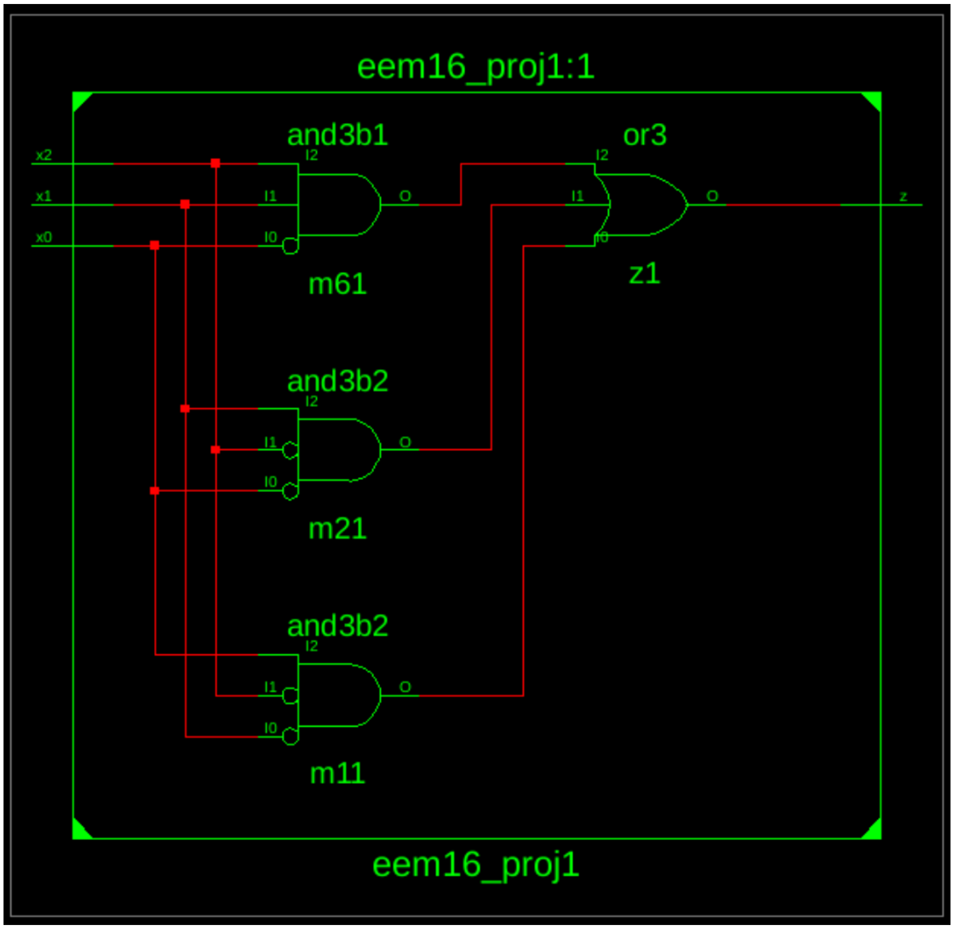


Figure 2.12 Gate network corresponding to $E(x_2, x_1, x_0) = \sum m(1, 2, 6)$.

(Figure 1) Ercegovac, M. (1999) *Introduction to Digital Systems*. New York

This is the schematic for the circuit coded in Verilog:



(Figure 2)

In summary, when the values of the input of x_2, x_1, and x_0 represent the decimal values 1, 2, or 6, then the output will be 1. Otherwise, the output z will be 0.

(4) The Verilog File

eem16_proj1.v

```
`timescale 1ns / 1ps
// Name: Joshua Liu
// Date: 1/31/19
// ID: 105136031
// Project: Project #1 Orientation of Verilog and ISE
// Class: EE M16 Logic Design of Digital Systems
// Discussion: 2B

module eem16_proj1(input x0, input x1, input x2, output z
    );
         // We create wires to link to gates in order to create a 2-level network
         // Wires notx0, notx1, and notx2 connect to an AND gate
         // Wires m1, m2, m6 connect to the OR gate as represented in the textbook
         wire notx0, notx1, notx2, m1, m2, m6;

// NOT gates
assign notx0 = !x0;
assign notx1 = !x1;
assign notx2 = !x2;

// AND gates
assign m1 = notx2 && notx1 && x0;
assign m2 = notx2 && x1 && notx0;
assign m6 = x2 && x1 && notx0;

// OR gates, output
assign z = m1 || m2 || m6;

endmodule
```

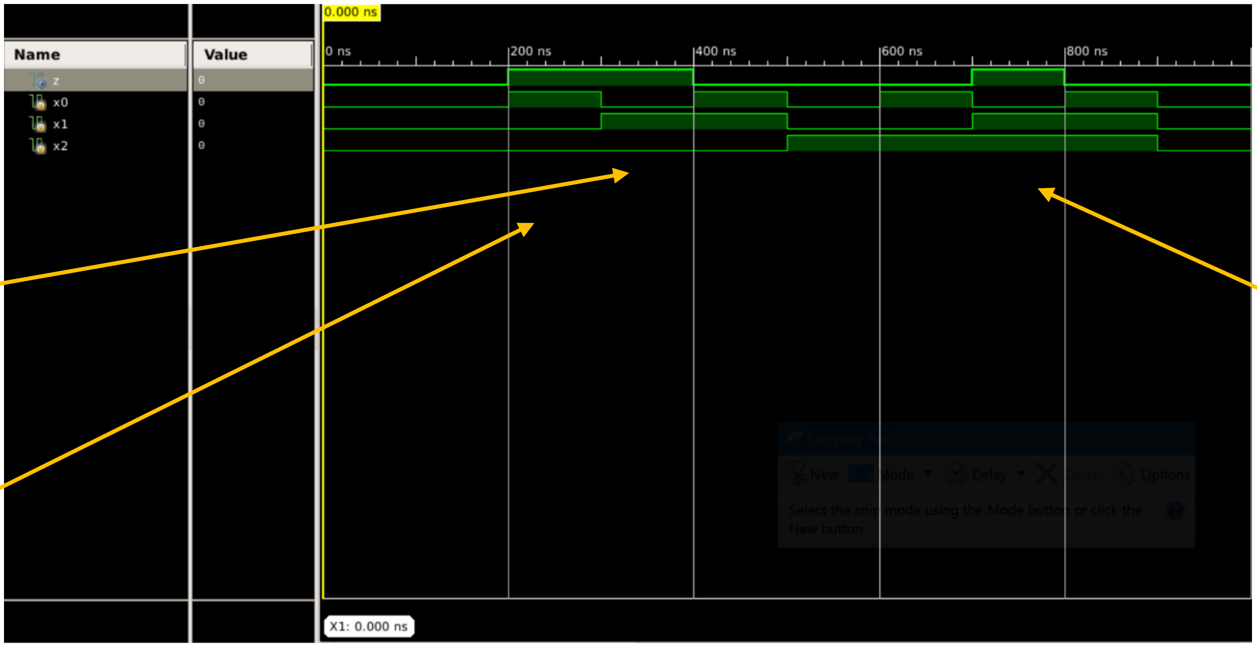
eem16_proj1_tb.v

```
`timescale 1ns / 1ps
// Name: Joshua Liu
// Date: 1/31/19
// ID: 105136031
// Project: Project #1 Orientation of Verilog and ISE
// Class: EE M16 Logic Design of Digital Systems
// Discussion: 2B

module simulation;

        // Inputs
        reg x0;
        reg x1;
        reg x2;

        // Outputs
        wire z;

        // Instantiate the Unit Under Test (UUT)
        eem16_proj1 uut (
                .x0(x0),
                .x1(x1),
                .x2(x2),
                .z(z)
        );

        initial begin
                // Initialize Inputs
                x0 = 0;
                x1 = 0;
                x2 = 0;
        // Wait 100 ns for global reset to finish
        #100;

        // Add stimulus here

        // After the first 100ns, every 100ns after is a period of time
        // when the input value increases(from 0 to 7), and ends with
        // a time interval of 100ns where all inputs are zero

        //000   0
        x2 = 0; x1 = 0; x0 = 0;
        #100; // Duration set to 100ns

        //001   1
        x2 = 0; x1 = 0; x0 = 1;
        #100;

        //010   2
        x2 = 0; x1 = 1; x0 = 0;
        #100;

        //011   3
        x2 = 0; x1 = 1; x0 = 1;
        #100;

        //100   4
        x2 = 1; x1 = 0; x0 = 0;
        #100;

        //101   5
        x2 = 1; x1 = 0; x0 = 1;
        #100;

        //110   6
        x2 = 1; x1 = 1; x0 = 0;
        #100;

        //111   7
        x2 = 1; x1 = 1; x0 = 1;
        #100;

        // Set all input to zero
        x2 = 0; x1 = 0; x0 = 0;
        #100;

        $finish;

    end

endmodule
```

(5) The Simulation Result

You have to demonstrate that your implementation works as specified by showing the simulation result from Vivado. Screendump the Timing Diagram that consists of waveforms of the inputs and the output. Please provide enough information on the timing diagram so that one of your colleagues who doesn't know anything about your project could understand which function you are trying to implement and evaluate it. A timing diagram without any explanation is subject to penalty.

Running my program on iSim, we can see the waveform below:



Input is 1.
$z = 1$.
This arrow points at the time interval 200ns-300ns when $x_0 = 1$, $x_1 = 0$, and $x_2 = 0$.

Input is 2,
$z = 1$.
This arrow points at the time interval 300ns-400ns when $x_0 = 0$, $x_1 = 1$, and $x_2 = 0$.

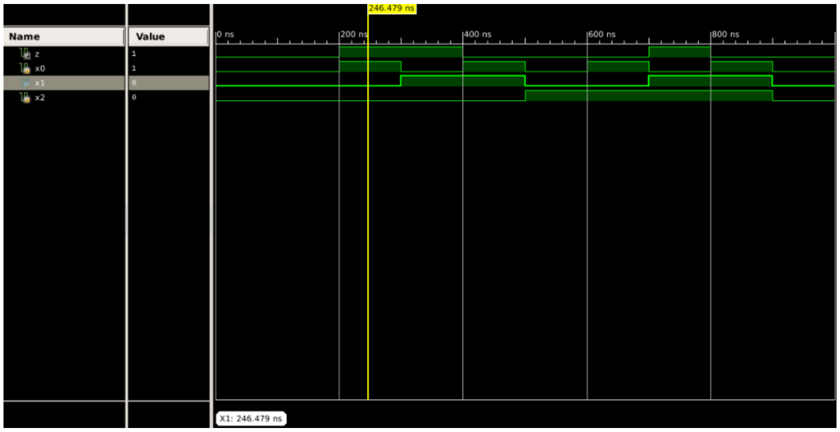
Input is 6,
$z = 1$.
This arrow points at the time interval 700ns-800ns when $x_0 = 0$, $x_1 = 1$, and $x_2 = 1$.

(Figure 3)

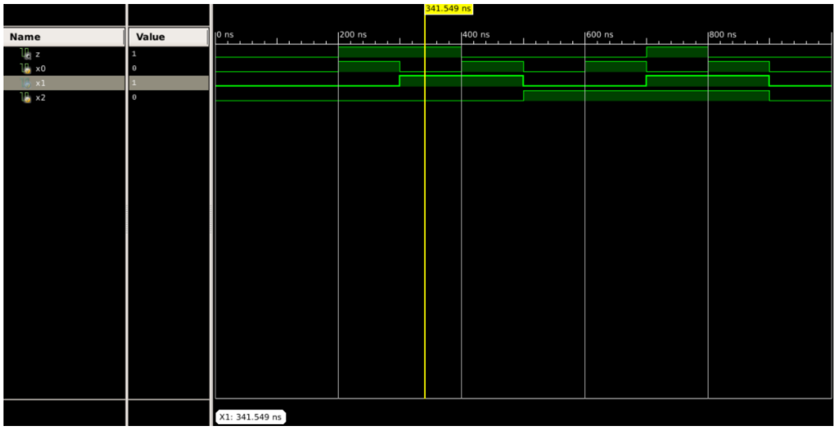
I programmed the timing diagram(Figure 3) to display an input of all 0's for the first 100ns as well as the last 100ns. Then, for the 800ns in between, each interval of 100ns represents an input value, incrementing from 0 to 7.

This timing diagram shows the function we are supposed to model, where when the inputs are either 1, 2, or 6, the output z is 1. However, when the inputs are not 1, 2, or 6, the output z is 0.

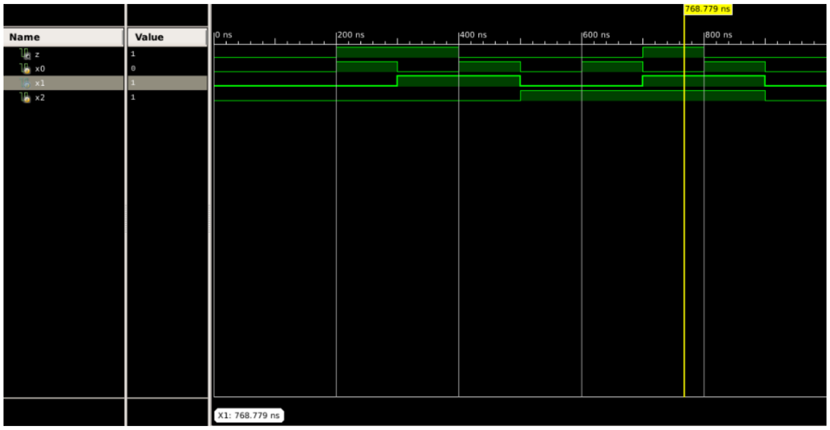
Below are screenshots of the waveform at the input values of 1, 2, and 6. We can see that the output z is 1 only at these points.



(Figure 4) We can see that when $x_2 = 0$, $x_1 = 0$, and $x_0 = 1$, $z = 1$.



(Figure 5) We can see that when $x_2 = 0$, $x_1 = 1$, and $x_0 = 0$, $z = 1$.



(Figure 6) We can see that when $x_2 = 1$, $x_1 = 1$, and $x_0 = 0$, $z = 1$.

In Figures 4, 5, and 6, $z = 0$ only at the input values 1, 2, and 6, which is exactly the circuit we are trying to implement.

(6) The Summary

This writeup should be short and state at least one problem you encounter during the implementation and the workaround you come up with, or any other comments you would like to make.

In this project, we were given a circuit from the textbook to implement in Verilog, which I am completely new to. The purpose of the project was to get us to know the basics of Verilog and to get familiar with Xilinx ISE’s text editor to write code. We needed to implement a two-level AND-OR gate network where $\mathrm{F(A, B, C)} = \mathrm{z} = \sum m(1,2,6)$. Using a waveform diagram, we checked out implementation of the circuit and ensured that for some input value, the output was what we desired.

While working on the project, I ran into many problems that I had to encounter. First, setting up the virtual machine to run Xilinx’s ISE was difficult. After downloading and installing it, I could open the virtual machine, but could not connect to the network or figure out a way to transfer files to and from the host machine, from the virtual instance. After hours of googling and watching videos of possible fixes, I was able to find the solution. After getting the program to work as I desired, I had to learn Verilog, which was not very easy since there seems to be so many ways to implement circuits. After hours of reading and watching tutorials, I was able to implement the circuit for the project and run the waveform program to test the circuit. I thought the whole process was rewarding since I learned so much about virtual machines and the program in general.