Final Report

Pipe Maintenance Robot

Chenxin Zhang

Michigan State University

ECE230

April 16, 2024

Dr. Jacob Honer  
College of Engineering   
Michigan State University   
428 S Shaw Ln   
East Lansing, MI 48825

Dear Dr. Honer,

Attached is a copy of Chenxin Zhang’s final report for the Design Project 4 in ECE230: Digital Logic Fundamentals, Spring 2024.

This report describes the process in learning Number System, Logic Circuit, Combinational Logic, Logic Reduction, Arithmetic Circuits, Sequential Logic, State Machine and Datapath Memory.

Based on the Design Projects that in this course, it makes students know how the hardware code working especially the Verilog.

I hope this report meets your requirements, and any questions will gladly be addressed by email.

Sincerely,

**A black background with white text

Description automatically generated*Chenxin Zhang***

**INTRODUCTION:**

Electricity is everywhere in the world, without the electricity almost everything is shutting down. This semester ECE230 Digital Logic Fundamentals teaches the students the logic fundamentals (*Number System, Logic Circuit, Combinational Logic, Logic Reduction, Arithmetic Circuits, Sequential Logic, State Machine and Datapath Memory.)* and how to use the Verilog to build a circuit through the Vivado platform. There are 4 Design Projects in this semester the following will describe what does it teaches.

**Method:**

**1.Number System**

In this chapter, students explore various number systems and methods for converting between them. The chapter introduces the binary number system as the only format acceptable to computers and machines. A key question arises: How does a computer determine whether a number is positive or negative? This is addressed by the concept of signed and unsigned numbers. In signed numbers, the most significant bit determines the sign of the number; a '0' signifies a positive number, whereas a '1' indicates a negative number.

It is well-known that in the decimal system, the sum of 1 and 9 equals 10, leading to an overflow. Similarly, the binary number system also experiences overflow. Overflow is necessary in computer systems because it allows for more efficient computation. Specifically, using the shift operations (left << and right >>) can significantly speed up processing compared to multiplication (\*) or division (/). This efficiency is crucial for enhancing performance in computing environments.

**2.Logic Circuit, Combinational & Sequential Logic**

In modern digital circuits, three fundamental logic gates—AND, OR, and NOT—serve as the building blocks for more complex designs. Engineers optimize these circuits for speed by employing circuit reduction techniques such as Karnaugh maps (K-maps) and De Morgan’s laws. Additionally, certain applications require circuits to maintain an initial state to control subsequent gates, necessitating the use of sequential logic. Examples of devices that employ sequential logic include various types of latches (such as D-latches and reset latches) and flip-flops. These components are critical in the design and functionality of advanced digital systems.

A diagram of a computer system

Description automatically generated**Figure1**: Circuit for Flip-Flop

This figure illustrates the circuit diagram of an edge-triggered flip-flop. Inputs are denoted as ‘d’ and ‘e’, with ‘clk’ representing the clock signal. The flip-flop outputs are labeled as ‘Q’ and ‘Q0’ (Q prime), where ‘Q0’ is the inverse of ‘Q’. The logic flow is controlled by a series of logic gates, which define the state of the flip-flop based on the input and the clock's edge transitions.

A diagram of a computer

Description automatically generated**Figure2**: Circuit for D-latch

The diagram depicted in this figure represents a D-Latch circuit. It accepts inputs ‘d’ and ‘e’ and has a control signal ‘clk’ that determines when the latch is open (transparent) or closed (holding). The outputs are labeled as ‘Q’ and ‘Q’’ (Q prime), indicating the current state and its complement. The D-Latch is designed to store a bit of data when the control signal is active and maintain its state when the control is inactive.

**3.State Machine & Verilog & Xilinx Vivado**

There is total 4 design projects in ECE230 in spring 2024 semester.

In the first design project, engineers learn that Verilog coding on the Vivado platform should consist of two distinct types of code: the module code and the testbench code. The module code is responsible for defining the inputs and outputs, and for specifying the operations the computer should perform on these inputs to produce the outputs. For example, given two inputs, a and b, the output x would be a logical AND of a and b, expressed as x = a & b. The testbench code is used to initialize the values of the inputs at different times. Take an instance the input a is 0 at time 100ns, “#100 a=0;” The primary objective of this project is to enable engineers to fully understand the logical relationships between the inputs and outputs, and the functions of both the module code and the testbench code and simulate that to get waveform from the compiler.

In the second design project, engineers have already mastered combinational logic and circuit reduction. In this phase, they will learn how to define multiple modules within the module code, rather than consolidating all code into a single module. For instance, they may first define the code for an AND gate, and subsequently reuse this module in the remaining code without rewriting it.

In the third design project, the engineers learn to coding a D-Flip Flop in the Verilog code and how to create a simple Finite State Machine (FSM). They design a traffic light state machine which is got four states and using the testbench code define which state should be turn on.

In the last project, the engineers are required to build a much larger FSM to design a Pipe Maintenance Robot, this robot should have several sensors that make the robot to make correct behavior in different situations such as the robot should change facing direction while meet wall in front of it, need cool down while meet fire…

**Result & Conclusions:**

This is the first profession class for the engineer. This course makes them deeply understand digital logic, starting from the basics of binary systems and progressing to complex combinational and sequential circuits. They explored various digital components like logic gates, flip-flops, and multiplexers, which are the building blocks of more complex electronic systems. One of the most enlightening aspects of the course was learning to apply Boolean algebra to simplify logic circuits, thereby enhancing their efficiency and reliability.

In the Design Project, they started from basic Verilog code to complex FSM code for a robot. This enhances their ability to make the code more efficiency and much faster than before.

A screenshot of a computer

Description automatically generated**Figure3**: Waveform of robot based on the JH’s testbench code.

Presented in Figure above is the waveform output from a behavioral simulation, derived from JH's testbench code. The waveform illustrates various signal transitions over time, captured within a digital simulation environment.

A screenshot of a computer

Description automatically generated**Figure4:** Waveform of robot based on CXZ testbench code.

Presented in Figure above is the waveform output from a behavioral simulation, derived from CXZ's testbench code. The waveform illustrates various signal transitions over time, captured within a digital simulation environment.