**Project 1**

**Introduction to the Lab Environment**

**EE 371 AC**

**January 08, 2016**

|  |  |
| --- | --- |
| **NAME** | **STUDENT ID** |
| Denny Ly | 1231185 |
| Minhhue H. Khuu | 1329349 |
| Ruchira Kulkarni | 1234324 |

Table of Contents:

[1. Abstract 3](#_Toc440040017)

[2. Introduction 3](#_Toc440040018)

[3. Discussion of the Project 3](#_Toc440040019)

[3.1 Design Specification 3](#_Toc440040020)

[3.2 Design Procedure 3](#_Toc440040021)

[3.3 System Description 3](#_Toc440040022)

[3.4 Software Implementation 3](#_Toc440040023)

[3.5 Hardware Implementation 3](#_Toc440040024)

[4. Testing 4](#_Toc440040025)

[4.1 Test Plan 4](#_Toc440040026)

[4.2 Test Specification 4](#_Toc440040027)

[4.3 Test Cases 4](#_Toc440040028)

[5. Presentation, Discussion, and Analysis of the results 4](#_Toc440040029)

[6. Analysis 4](#_Toc440040030)

[6.1 Analysis of any errors 4](#_Toc440040031)

[6.2 Analysis of possible errors 4](#_Toc440040032)

[6.3 Feedback upon the analysis 4](#_Toc440040033)

[7. Summary and Conclusion 4](#_Toc440040034)

[8. Appendices 4](#_Toc440040035)

# Abstract

The goal of this lab was to understand different developing environments such as iVerilog and gtkwave and Signal Tap Tool in the Quartus II IDE for developing Verilog code. We achieved this by making four 4-bit down counters, namely the ripple-down counter, a four stage synchronous down counter and a Johnson down counter. We also familiarized ourselves with the basics of C programming by writing a simple currency conversion calculator.

# Introduction

In this lab, our main focus was to analyze the process of model design and testing process by implementing different down counters in Verilog. To understand this process,

The tools we used in this lab are:

* iVerilog
* gtkwave
* Quartus II Signal Tap
* Cyclone V FPGA
* Command line tool
* GCC C Compiler
* CodeBlocks

# Discussion of the Project

## Design Specification

1. **Four Bit Ripple Down Counter**

A ripple counter is an asynchronous counter where only the first flip-flop is clocked by an external clock. All subsequent flip-flops are clocked by the output of the preceding flip-flop. The ripple counter ranges from 0000 to 1111. On reset, the counter starts at 0000, then goes to 1111, 1110, 1101, … till 0000 again.

According to the design specifications given to us, we made a gate or structural model with active low reset. Gate level or structural design implies interconnecting logical gates with wires (such as and, nor, or gates) only. The advantages of this design is that it is has a lot of implementation details.

In our design, we drew out the required bit patter results for the ripple counter and formed a Boolean equation from it. We used this Boolean equation as a reference for designing and constructing or circuit at the gate level in Verilog. This table, equations and circuit is shown in the design procedure for ripple down counter. We also used the code snippet provided to us for the D flip flop in the spec.

1. **Synchronous Down Counter (Dataflow Model)**

In a synchronous down counter, all the output bits change state simultaneously. We designed the counter using the data flow model as mentioned in the spec with an active low reset. In a dataflow design, a module is implemented by specifying the movement of the data. This allowed us to use assign statements instead of using AND gates in our module for the counter. We also used the code snippet provided to us for the D flip flop in the spec.

In our design, we drew out the bit pattern which ranged from 0000 to 1111. On reset, the counter starts at 0000, then goes to 1111, 1110, 1101, … till 0000 again. We also made use of K-maps to determine the new state value using combinational logic and present values.

1. **Synchronous Johnson Counter**

The Johnson Counter is a shift register with feedback with inverted output Q of the last flip-flop is now connected back to the input D of the first flip-flop.

We designed the Johnson Counter using a behavioral model with reset on active low. In behavioral modelling, the underlying hardware is abstracted away based upon an algorithmic description. The Johnson 4-bit counter only takes 8 different values. At reset the counter is set to 0000. It follows the pattern of 0000, 1000, 1100, 1110, 1111, 0111, 0011, 0001.

1. **Synchronous Down Counter (Schematic Entry)**

## Design Procedure

1. **Four Bit Ripple Down Counter**

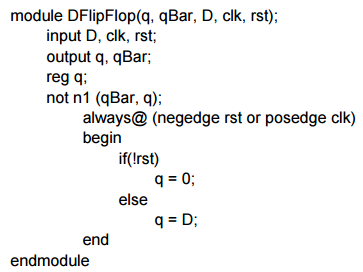
**Truth Table for Down Counter starting at reset**

|  |  |  |  |
| --- | --- | --- | --- |
| **Q[3]** | **Q[2]** | **Q[1]** | **Q[0]** |
| **0** | **0** | **0** | **0** |
| **1** | **1** | **1** | **1** |
| **1** | **1** | **1** | **0** |
| **1** | **1** | **0** | **1** |
| **1** | **1** | **0** | **0** |
| **1** | **0** | **1** | **1** |
| **1** | **0** | **1** | **0** |
| **1** | **0** | **0** | **1** |
| **1** | **0** | **0** | **0** |
| **0** | **1** | **1** | **1** |
| **0** | **1** | **1** | **0** |
| **0** | **1** | **0** | **1** |
| **0** | **1** | **0** | **0** |
| **0** | **0** | **1** | **1** |
| **0** | **0** | **1** | **0** |
| **0** | **0** | **0** | **1** |
| **0** | **0** | **0** | **0** |

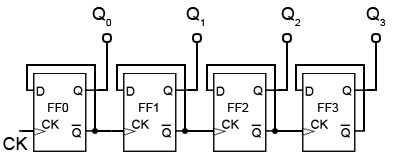
Table (1)

**Boolean Equation:**

**Code**



**Ripple Down Counter Diagram:**



1. **Synchronous Down Counter (Dataflow Model)**

**Truth Table**

Truth table that we used for the synchronous counter is the same one as the Table (1).

**Boolean Equations**

By observation, the Boolean equations are as follows:

D[0] =

Q [0]

Q [1]

|  |  |
| --- | --- |
| 1 | 0 |
| 0 | 1 |

D[1] =

Q [1]

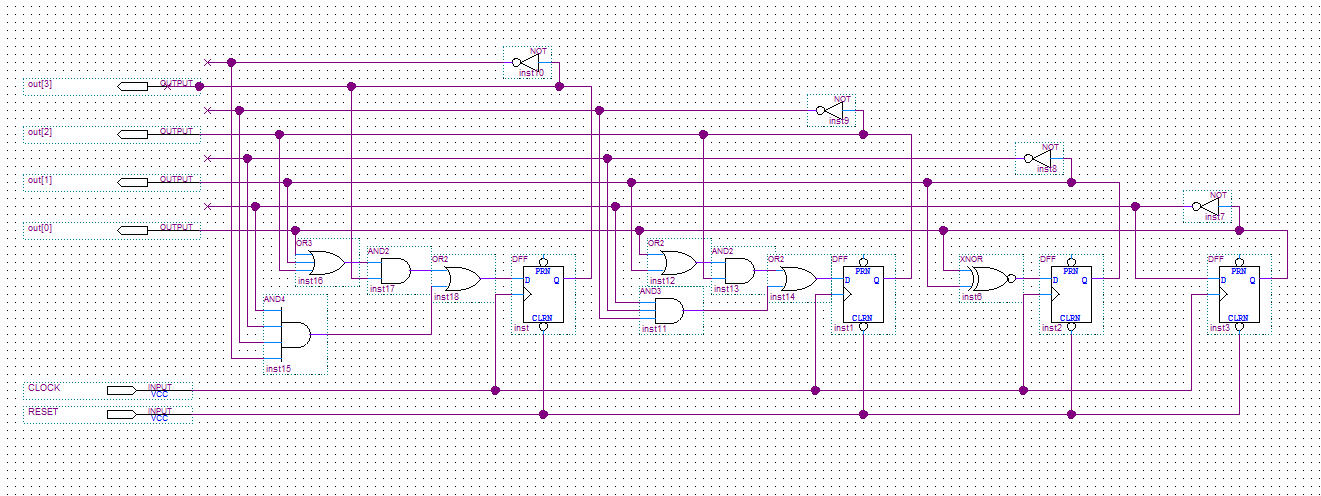
|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |

D[2] =

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 |

D[3] =

**Diagram**

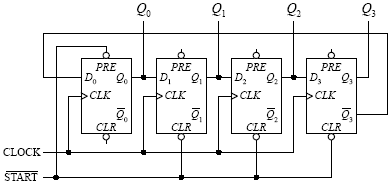


1. **Synchronous Johnson Counter**

**Truth Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **0** | **0** | **0** | **0** |
| **1** | **0** | **0** | **0** |
| **1** | **1** | **0** | **0** |
| **1** | **1** | **1** | **0** |
| **1** | **1** | **1** | **1** |
| **0** | **1** | **1** | **1** |
| **0** | **0** | **1** | **1** |
| **0** | **0** | **0** | **1** |

**Diagram**



1. **Synchronous Down Counter (Schematic Entry)**

## System Description

DENNY LY

## Software Implementation

Please refer to the code submitted via canvas dropbox.

## Hardware Implementation

We used the DE1\_SOC FPGA board.

# Testing

## Test Plan

MINH

## Test Specification

MINH

## Test Cases

MINH

# Presentation, Discussion, and Analysis of the results

# Analysis

## Analysis of any errors

Software and tool problems. Iverilog not running because it is not included in windows path variable.

Problem wiring modules.

## Analysis of possible errors

One possible error is by selecting a clock speed that is too fast. This causes the gate delays to have a more noticeable effect.

Verilog syntax errors.

## Feedback upon the analysis

# Summary and Conclusion

# Appendices

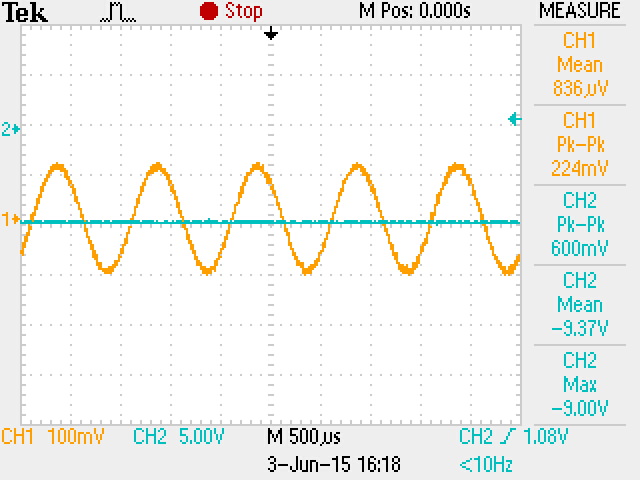


Figure **1.** Actual 0 W Output Waveform.