System Design Document

for the

Half Adder Module

University at Buffalo, The State University at New York

EE478 Fall 2019

The information enclosed in this document and the source code generated for this laboratory assignment were generated by the named students on this cover page. Detection of plagiarism in a submitted lab or project design document will result in an automatic failing grade in the course by all parties involved.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Written By** | Daniel Giovino |  | **Revision History** | | | |
| **Engineer** | Daniel Giovino |  | **Rev** | **Date** | **Session** | **Approved/Grade** |
| **Engineer** | Peter M. VanNostrand |  | Orig | 09/23/19 | Mon. 9AM |  |
| **Teaching Assistant** |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  | **EE478F19**  **LAB 2** |

# Table of Contents

[Table of Contents 2](#_Toc492225413)

[List of Figures 2](#_Toc492225414)

[List of Tables 2](#_Toc492225415)

[1. Introduction 3](#_Toc492225416)

[1.1 Overview 3](#_Toc492225417)

[1.2 Document Scope 3](#_Toc492225418)

[1.3 Intended Audience 3](#_Toc492225419)

[1.4 Revision History 4](#_Toc492225420)

[2. System Design Overview 5](#_Toc492225421)

[2.1 System Block Diagram and Description 5](#_Toc492225422)

[2.2 Half Adder Module 6](#_Toc492225423)

[3. Testing and Verification 7](#_Toc492225424)

[3.1 Testbench and Simulation 7](#_Toc492225425)

[3.2 Objective Verification 7](#_Toc492225426)

[3.3 Verification Matrix 7](#_Toc492225427)

[Glossary 8](#_Toc492225428)

[List of Abbreviations 8](#_Toc492225429)

[Hardware References 8](#_Toc492225430)

# List of Figures

Figure 1: ALU System Diagram 4

Figure 2: Half Adder Simulation Waveform 6

# List of Tables

**Table 1: Half Adder Truth Table** 6

# Introduction

## Overview

In this section, provide an overview of the lab requirements from a high level. You should discuss the FPGA hardware and the basic functionality of the system. You should not go into detail about the specific design used. For example, if you developed a half adder module that uses switches as inputs and displays the outputs on LEDs, your overview would read (Note that it is expected that this document provide significant detail about the system you designed. Therefore, expect this document to be significantly longer than this template):

A half adder is a basic digital circuit that computes the arithmetic sum of two bits of equal weight and produces two digital outputs representing the two-bit sum. A half adder system was developed on a Xilinx FPGA using the Xilinx ISE WebPack CAD tool. The inputs for the system were provided using slider switches, and the binary outputs were displayed on LEDs located on the board.

## Document Scope

This document has been written to provide information regarding the ALU circuit developed in Lab 2 of EE478. The inputs, outputs, and operation of this device are described below as well as details regarding implementation on an FPGA and testing using a simulated test bench.

## Intended Audience

This document is intended for use by the EE478 TAs and professors for use in grading as well as by the members of group 1L in the Monday 9am lab session.

# System Design Overview

## System Block Diagram and Description

The ALU takes two, 2-bit inputs A and B, which are labeled as A0, A1, B0, and B1 in Figure 1. These inputs are connected to four slider switches on the Xilinx board. It also takes an opcode from BTN0, BTN1, BTN2, and BTN3, which are labeled as such in the following diagram.

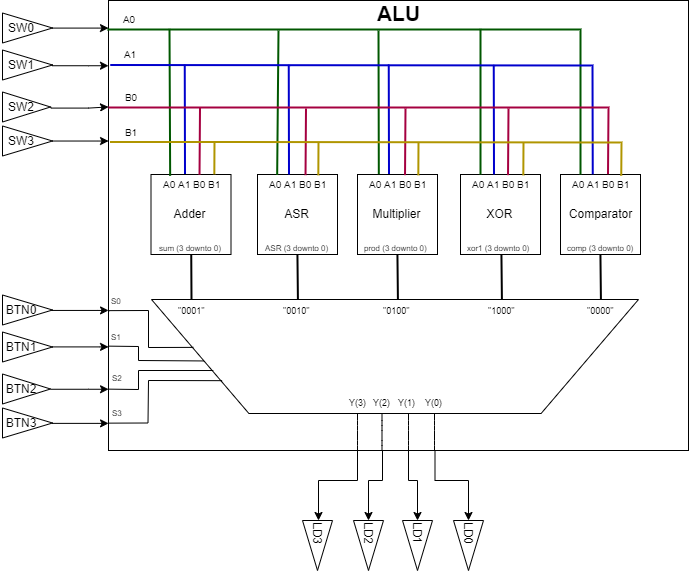


Figure 1: ALU System Diagram

The input bits from vectors A and B (each containing 2 bits) are fed to each of the following modules: Adder, Arithmetic Shift Right (ASR), Multiplier, XOR, Comparator (See following sections for details on each). Each of these modules output a 4-bit signal which is sent to a MUX which uses the opcode to determine which operation it should output to the final output of the system, which is displayed on the LEDs labeled LD0, LD1, LD2, and LD3.

## Half Adder Module

Every module must have a section like this. These sections describe the detailed operation of each module, in terms of how the module outputs are derived from their inputs. Diagrams and truth tables might be necessary in these sections. For instance, a baud clock module might include a waveform timing diagram. For example:

This module implements the functionality of a basic half adder digital circuit. The half adder receives bits and and computes their two-bit arithmetic sum. The sum is stored on the and output signals, in which the binary number represents the two bit sum. The truth table of the half adder is given by:

Table 1: Half Adder Truth Table

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

Basic analysis gives that the sum is computed as a logical exclusive OR of the inputs, and the carry out function is a logical AND of the inputs.

# Testing and Verification

In this section, you will discuss the verification that your design meets the requirements. Generally, this will involve a discussion of your simulations and testbenches, as well as discussion of your physical verification on the board.

## Testbench and Simulation

You should detail the testbench setup, simulator settings, runtime, and include screenshots of the waveforms. For more complex systems, there may be testbenches for individual modules as well as the entire system. You should include waveforms from each of these testbenches. The waveforms you include should show compliance with the requirements. For example:

For the Half Adder module design, a simulation was run at the top level of the design. The and inputs were driven to all four possible combinations, and the and outputs were visualized using ISim for proper performance and compliance to Table 1 and the functionality of a Half Adder.

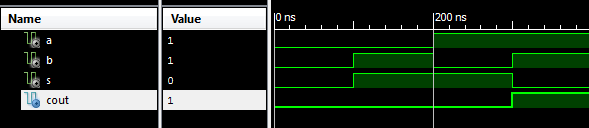


Figure 2: Half Adder Simulation Waveform

## Objective Verification

In this section, describe how you verified the functionality of the design once it was programmed onto the Atlys board. No images are required in this section, but they can be included if helpful. For example:

The functionality of the Half Adder module was verified in deployment on the Digilent Atlys development board. Two slider switches were used to provide stimulus for the and inputs, and the correct behavior was observed on the LEDs representing the and signals.

# Glossary

This section includes helpful supplemental information. The subsections will depend on the lab, but may include references to online datasheets for hardware modules, supplemental information about communication interfaces, etc.

## List of Abbreviations

Engineering documents are often full of abbreviations and acronyms. Design documents generally include a list of these abbreviations alongside their full text. For example

* VHDL: VHSIC Hardware Description Language
* VHISC: Very High Speed Integrated Circuit
* FPGA: Field Programmable Gate Array
* LED: Light Emitting Diode
* CAD: Computer Aided Design

## Hardware References

* Xilinx Spartan 6 FPGA <https://www.xilinx.com/support/documentation/data_sheets/ds160.pdf>
* Digilent Atlys development board <https://reference.digilentinc.com/atlys/atlys/refmanual>
* Xilinx ISE WebPack CAD tool, etc.