System Design Document

for the HDMI Display

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** | Peter M. VanNostrand |  | **Revision History** | | | |
| **Engineer** | Daniel Giovino |  | **Rev** | **Date** | **Session** | **Approved/Grade** |
| **Engineer** | Peter M. VanNostrand |  | Orig | 10/28/19 | Mon. 9AM |  |
| **Teaching Assistant** |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  | **EE478F19**  **LAB 5** |

# Table of Contents

Table of Contents 2

List of Figures 2

List of Tables 2

1. Introduction 3

1.1 Overview 3

1.2 Document Scope 3

1.3 Intended Audience 3

2. System Design Overview 4

2.1 System Block Diagram and Description 4

2.2 PLL Module 4

2.3 Timing Instance Module 5

2.4 State Machine Process 5

2.5 Box Draw Process 6

2.1 HDMI Controller Module 6

1. Testing and Verification 7

1.1 Testbench and Simulation 7

1.2 Objective Verification 7

Glossary 8

2.2 List of Abbreviations 8

2.3 Hardware References 8

# List of Figures

Figure 1: Lab 5 HDMI Block Diagram 4

Figure 3: Clock Divider Circuit 5

# List of Tables

Table I: Finite State Machine 5

Table II: Channel Intensities by State 6

# Introduction

## Overview

This device uses a simple state machine to control the output notes to an audio codec chip. The output notes cycle through 4 different notes, always in the same order, and only starts the sequence when a pushbutton is pressed. This state machine was first designed on paper and then implemented and tested on a Xilinx FPGA using the Xilinx Vivado CAD tool.

## Document Scope

This document has been written to provide information regarding the state machine with audio output circuit developed in Lab 5 of EE478. The inputs, outputs, and operation of this device are described below as well as details regarding implementation on an FPGA.

## 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 HMI system implemented in this lab takes the system clock and a pushbutton signal as inputs and produces the TMDS and TDMSB signals which are sent to the monitor using TMDS.

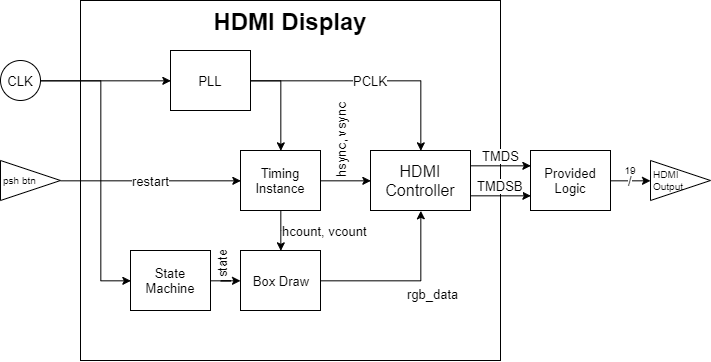


Figure 1: Lab 5 HDMI Block Diagram

The output signal draws a square on the display which changes once per second between the colors red, green, blue, and yellow. This is accomplished using a finite state machine. For appropriate timing a PLL was generated and connected to the provided timing instance. Finally, a box draw process takes the current state and timing information and generates rgb data which it sends to the HDMI controller component provided by the instructor. The HDMI controller then generates the final TMDS and TDMSB output signals.

## PLL Module

This module turns the system clock of frequency 125 MHZ into a clock signal named pclk of frequency 74.25 MHZ. This module was made using the built-in clocking wizard in Vivado. The wizard multiplies and divides the system to get the output frequency of almost exactly 74.25 MHz on pclk. No schematic is given for this module since Vivado makes this circuit and we haven’t yet learned how it does this.

## Timing Instance Module

This module, provided by the instructor, takes characteristics of the screen as inputs and generates the output signals hsync, hcount, vsync, and vcount. These signals are used to represent which pixel on the screen is currently being drawn and to synchronize the drawing of each row and column of pixels so that they align with the edges of the screen. The implementation of this module was handled by the course staff and as such is not in the purview of this report.

## State Machine Process

This process has inputs of the system clock signal and the pclk signal generated by the PLL. Using these it updates a state signal which determines which color is displayed on the screen. The state signal and therefore the displayed color changes once per second to red, green, blue, or yellow. This is done by implementing a clock divider as shown in Figure 2. This clock divider counts to the value 74,250,000 which is the number of pclk cycles in one second and then changes to the next state as shown by Table I and resets the counter to zero. The process only executes when sys\_clk changes and only updates the counter on the rising edge of pclk.

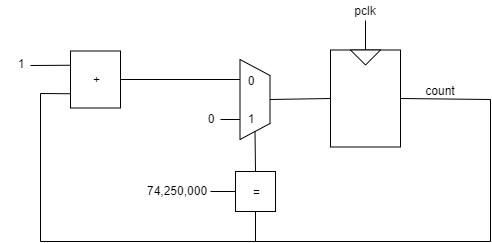


Figure 2: Clock Divider Circuit

Table I: Finite State Machine

|  |  |
| --- | --- |
| Current State | Next State |
| sRED | sGREEN |
| sGREEN | sBLUE |
| sBLUE | sYELLOW |
| sYellow | sRED |

The truth table for the finite state machine is implemented using a case statement. When the counter reaches the terminal count it is reset to zero and then the state is updated to the next state in the progression.

## Box Draw Process

This process takes the hcount, vcount, and state as inputs and generates updates three output signals called red\_data, green\_data, and blue\_data which correspond to the red, green, and blue components of the output signal respectively. These components control the color of the current pixel and as such this process triggers once for every pixel in the output image as the hcount and vcount change to “scan” across the screen.

To determine which pixels should be colored to draw a square on the screen a set of four constants are used, these are h\_min, h\_max, v\_min, and v\_max which represent pixel coordinates of the edges of the square. The pixels are then colored under the following conditions

This essentially drives all pixels outside the areas of the square to be black, with all pixels inside the square being colored according to the current state. The exact value of each channel for each state is shown in Table II

Table II: Channel Intensities by State

|  |  |  |  |
| --- | --- | --- | --- |
| State | Red Data | Green Data | Blue Data |
| sRED | 0xFF | 0x00 | 0x00 |
| sGREEN | 0x00 | 0xFF | 0x00 |
| sBLUE | 0x00 | 0x00 | 0xFF |
| sYELLOW | 0xFF | 0xFF | 0x00 |

The conditional value of channel data is achieved using a case statement with the current state being used as the key

## HDMI Controller Module

This module, provided by the instructor, takes the color channel information generated by the box draw process as well as the hcount and vcount signals generated by the timing instance and generates two outputs called TMDS and TMDSB. These signals are the TMDS encoded signals which are sent to the display. The module also takes in the system clock and a reset signal to ensure that it stays synchronized with the rest of the system. The implementation of this module was handled by the course staff and as such is not in the purview of this report.

# Testing and Verification

To ensure that our circuit functioned as intended we performed testing. The way we tested this code was by synthesized our design and loading it onto an FPGA where we performed testing of the physical circuit. A testbench, which is a good programming practice, was not used in this lab.

## Testbench and Simulation

A testbench was not used in this lab, even though it is a good programming practice. A testbench could have been used to test each module of this design individually.

## Objective Verification

Once we synthesized our design and loaded it onto the FPGA we tested the circuit connecting a display to the “HDMI OUT” port of our FPGA. We then expected to see a large square in the center of the screen which changed color every four seconds. As this was observed result we concluded that our circuit was operating as intended.

# Glossary

This section includes helpful supplemental information to aid in understanding the content of this report.

## List of Abbreviations

Below are selected abbreviations used in this document

* VHDL: VHSIC Hardware Description Language
* FPGA: Field Programmable Gate Array
* CAD: Computer Aided Design
* PLL: Phase-Locked Loop

## Hardware References

* Zybo Z7 Manual ([DigitalInc](https://reference.digilentinc.com/_media/reference/programmable-logic/zybo-z7/zybo-z7_rm.pdf))
* Xilinx Vivado CAD tool ([Xilinx](https://www.xilinx.com/products/design-tools/vivado.html))