**Sequential Logic Design with Counters in Verilog Design Report**

Benjamin Betancourt

Lab Dates: February 10th – February 18th, 2025

Date of Submission: February 17th, 2025

Table of Contents

**Introduction3**

**Implementation3-5**

**Results5-7**

**Conclusion7**

***Introduction:***

This project is a good introduction to the storage of values and state management. The requirement of a load case and store case for the GCD calculation introduced the requirement for previous state storage. This project also required the implementation of a GCD algorithm in c as well, so it offered a good comparison between traditional programming languages and hardware description languages. The developed algorithm was used in the implementation of the Verilog step as it allowed for expectations of what the code looks like to be drawn. The Verilog step made use of sequential blocks such as the always block and case statements to compute the result. Lastly, registers were used heavily to store previous states in the case that the load button is not pressed.

**GCD Algorithm:**

The steps of the provided algorithm are as follows:

1. Start with two positive numbers, A and B
2. If min(A, B) == 0, jump to step 5
3. Subtract the smaller number from the larger number
4. Jump to step 2
5. The maximum of A and B is the GCD

The running of the algorithm can be seen below with each step and intermediary calculation included:

Algorithm given A=55, B=121

**GCD Implementation in C:**

My implementation in C using the algorithm and its output shown above is as follows:

#include <stdio.h>

int max**(**int a**,** int b**)** **{**

**if** **(**a **>=** b**)** **{**

**return** a**;**

**}**

**else** **{**

**return** b**;**

**}**

**}**

int min**(**int a**,** int b**)** **{**

**if** **(**a **<=** b**)** **{**

**return** a**;**

**}**

**else** **{**

**return** b**;**

**}**

**}**

int GCD**(**int a**,** int b**)** **{**

**if** **(**min**(**a**,** b**)** **==** 0**)** **{**

**return** max**(**a**,** b**);**

**}**

**while** **(**min**(**a**,** b**)** **!=** 0**)** **{**

**if** **(**a **-** b **>=** 0**)** **{**

a **=** a **-** b**;**

**}**

**else** **{**

b **=** b **-** a**;**

**}**

**}**

**return** max**(**a**,** b**);**

**}**

void main**(**void**)** **{**

printf**(**"GCD: %d\n"**,** GCD**(**165**,** 363**));**

printf**(**"GCD: %d\n"**,** GCD**(**48**,** 180**));**

printf**(**"GCD: %d\n"**,** GCD**(**27**,** 36**));**

**}**

A black background with white text

AI-generated content may be incorrect.

GCD C program output