Stamatios Morellas

CPR E 281 – Section U

Final Project – 8-bit Booth’s Multiplier

Student ID: 971626680

What to submit?

- A top-level diagram of your design, showing inputs, outputs, components, and the sizes of data signals and the meanings assigned to all of these.

- A list of all block elements that you had to design for this project

- For all/most components, also provide a written description (e.g., the finite state machine, the register file, adder, etc.). This description should include things like state tables, corresponding state assigned tables, truth tables, K-maps, expressions, code, and visual images of any blocks created for this project.

- Description and purpose of any gates used outside of created blocks

- Include several test cases that a user could follow to see the functionality of the circuit

- It is OK for some of these to submit hand written notes/descriptions. In other words, you don't have to type everything.

- All submitted materials should be neatly organized in a professional manner so that your design is easy to follow and understand. Staple together all of your pages or use a binder clip or put all of them a folder. Make sure to write your name, student ID, and lab section letter on the first page.

- To get a grade you must demonstrate your circuit to your TA. Go through at least two of your test cases to demonstrate its functionality. The TA may choose to try other test cases as well.

- Finally, e-mail a ZIP file of the folder that contains your final project to your TA.

Final Project Report

## Overview

I chose to work on option #2 (8-bit Booth’s Multiplier) for my final project. Although I was not able to fully complete the project, I was able to get parts of it to work. The way in which the Booth’s multiplier works is by using an algorithm that multiplies 2 signed binary numbers, in this case each having a length of 8-bits, using two’s complement notation. The implementation that has been used in this case is by using a 2-port read and 1-port write **register file**, a modified **adder/subtractor** circuit, and a **finite** **state** **machine**. The multiplication is done by shifting and adding, so the adder/subtractor circuit needs to be modified accordingly to include selective sign extension capabilities.

## Part A: The Register File

The register file has two read ports, one write port, and the data is stored in four 8-bit registers. It also includes components that will specify which register will read and which register will write. A 2-to-4 decoder is used to write to the register, and there are two 4-to-1 bus multiplexers that contain the result of the two outputs for the register file. The ports of the register file are as follows:

**DATAP**: First output of 8-bit data from the register file.

**DATAQ**: Second output of 8-bit data from the register file.

**RP**: 2-bit Read address that specifies which register will send its output through DATAP.

**RQ**: 2-bit Read address that specifies which register will send its output through DATAQ.

**WA**: 2-bit Write address that indicates which register will update its data.

**LD\_DATA**: 4-bit data that will be written into the register specified by the address in WA.

**WR**: When this is zero, no register will update its data. When this is one, the register specified by WA will update its value with the value in LD\_DATA.

**CLK**: A typical clock connection.

The