CMPE 260 Laboratory Exercise 2

Register File

Benjamin Maitland
Performed: February 15th, 2018
Submitted: February 22nd, 2018

Lab Section: 4
Instructor: Sayed Ashraf Mamun
TA: Piers Kwan

Lecture Section: 01
Professor: Richard Cliver

By submitting this report, you attest that you neither have given nor have received any assistance (including writing, collecting data, plotting figures, tables or graphs, or using previous student reports as a reference), and you further acknowledge that giving or receiving such assistance will result in a failing grade for this course.

Your Signature: _____

Table of Contents

1 Abstract

In this exercise, a Register File is constructed out of Multiplexers, Decoders, and Registers. Generics were used to allow for different configurations. The register file contained 8 16 bit registers, with a single input, an input select, a write enable, and two outputs and selections. The register file was tested successfully with behavioral and post route simulations, and then on a physical Field Programmable Gate Array (FPGA).

2 Design Methodology

For this exercise, the goal is to implement a register file. This register file has 8 selectable 16-bit registers, with one being selectable as an input and two being selectable as an output. A block diagram of the register file is shown in Figure 1.

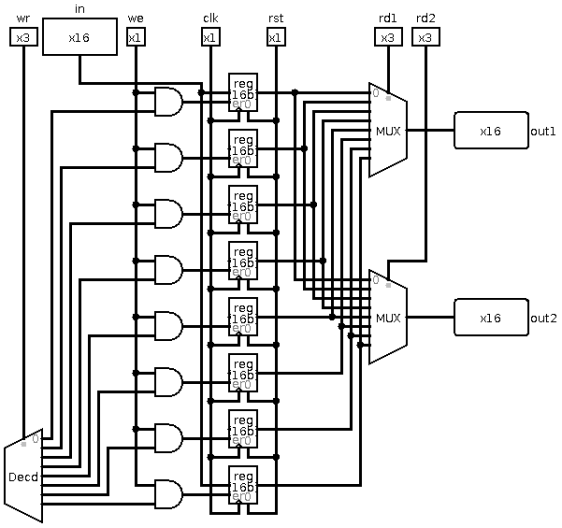

Figure 1: Register File block diagram

The register file was composed of a 3x8 Decoder, two 3x8 Multiplexers, and 8 16-bit Register Modules.

A 3x8 decoder has a 3 bit select input that maps to pull a single bit in the 8-bit output high. A block diagram for the Decoder is shown in Figure 2.

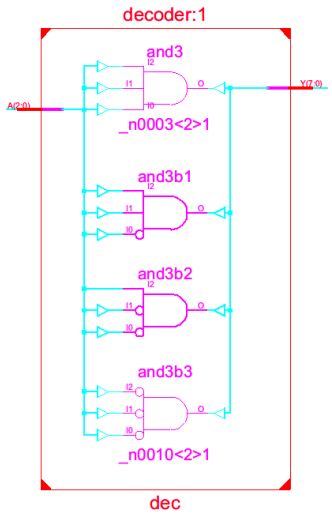

Figure 2: Decoder block diagram

A 3x8 multiplexer has a 3 bit select input that selects one of the 8 input signals to map to the single output signal. A block diagram for the multiplexer is shown in Figure 3.

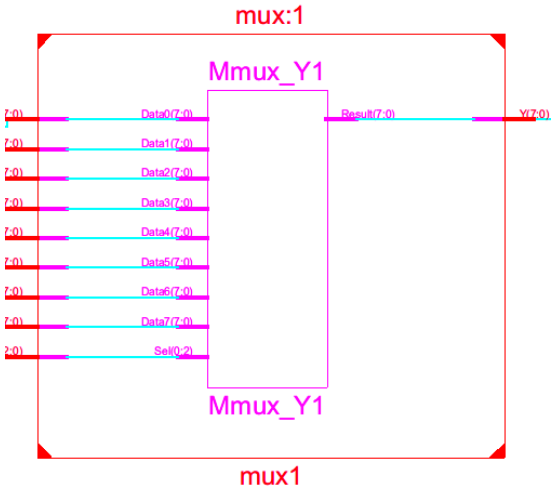

Figure 3: Multiplexer block diagram

A register module has a generic reset value and a generic N bit width. Its inputs are the N bit vector to store, a write enable that enables when pulled high, an asynchronous reset that enables when pulled high, and a clock signal. The output is a N bit vector that is the stored value. A block diagram for the register module is shown in Figure 4.

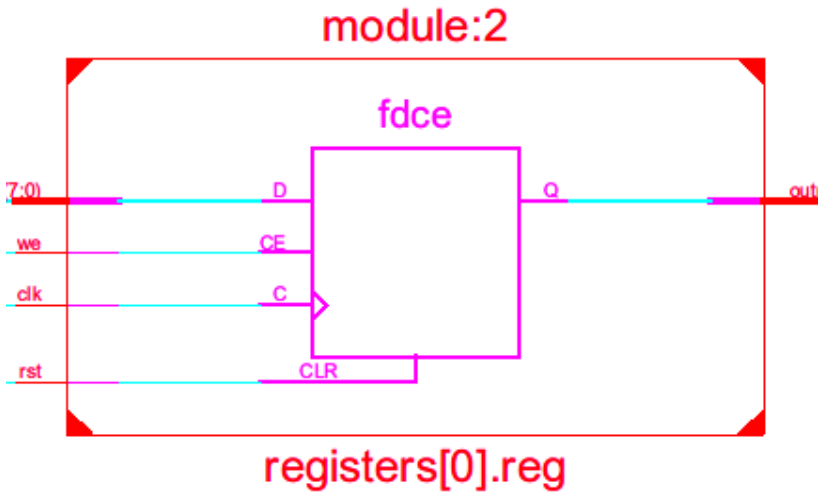

Figure 4: Register Module block diagram

In the register file, the decoder is connected to the write select input. The outputs of the decoder are each ANDed with the file's write enable bit, and that result is then passed to the write enable bit of the respective registers. The input to each of the registers is connected to the file's input. The output of each vector is connected to two multiplexers, each of which is connected to a register select and it's respective output.

After the register file was designed, it was implemented in VHDL. A VHDL testbench gave the register file a series of inputs and compared the outputs to expected ones. This testbench was then used to do a behavioral simulation in GHDL. This simulation was successful.

After the code was successfully simulated, the code was synthesized in Xilinx ISE Project Navigator 14.7. This allowed for a post-route simulation in ModelSim. This simulation was successful.

After synthesis and post-route testing, inputs and outputs were mapped to the switches on the Nexys 3 FPGA Development Board for manual testing. This was done using Plan Ahead. Switches were mapped to the write select, register selects, and the input. Buttons were mapped to enable and reset. LEDs were mapped to the outputs.

3 Results

The results of the behavioral simulation are shown in Figure 5.

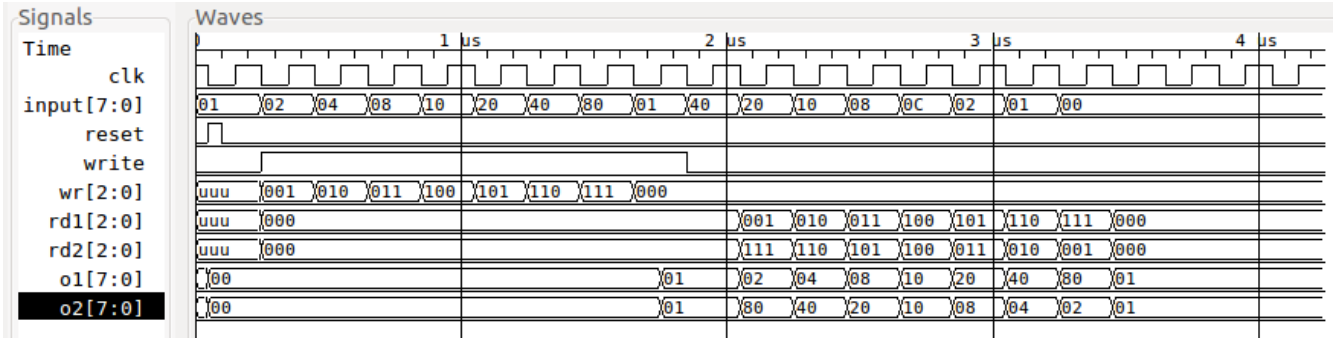

Figure 5: Behavioral simulation

After the FPGA was programmed, it worked successfully.

After synthesis, 33 slices were used, with 36 LUTs and 64 slice registers used.

4 Conclusion

This exercise was successful. A Register File was constructed out of Multiplexers, Decoders, and Registers. Generics were used to allow for different configurations. The register file contained 8 16 bit registers, with a single input, an input select, a write enable, and two outputs and selections. The register file was tested successfully with behavioral and post route simulations, and then on a physical Field Programmable Gate Array (FPGA).