

Lab 3-20

Group Members: Colton Watson, Benjamin Leaptrot, Christian Giauque, Nathan Hummel

The Memory Access Interface

The Memory Access Interface for Lab 3 is using a 2D storage of data components to store the data into memory so that we can use a load or store data opcode for the ALU, we called our project setup in Quartus bram and set up inputs from Data_a and Data_b as well with a address for each component for Data_a and Data_b so we will use a 2D interface setup so that we can implement the address and the data points in memory.

The testbench for the RAM module writes using the first input port zero to thirty-two to memory addresses zero to thirty-two. It then writes using the second input port the value thirty-three to sixty-four to addresses thirty-three to sixty-four. Finally, it reads all addresses from address zero to sixty-four. What occurs is that the monitor outputs zero to sixty-four sequentially.

The demo used for this lab is a simple FSM that goes through five different states: reset state, set values state, modify values state, write back state, and then display state. The reset state assigns registers and outputs to safe values. The set values state assigns arbitrary values to a variety of memory address locations. The modify values state pulls from memory, does an operation on that value, and stores it locally. The write back state writes to memory the operation done in the previous state, effectively simulating a simple fetch, operation, and write back routine. Finally, the display state displays the value written back into memory. In our example, we initially wrote 0x70 into ram address 513, added 0x03, then wrote that back into ram address 513, then displays 0x73 on the seven segment displays.

This can be demonstrated with the image set to the right. The top image is the initial state, reading address 0x0 from memory, which currently holds 0x0. The middle image demonstrates the seven-segment



displays outputting 0x70, the value written to address 513. The last image demonstrates the modify and write-back stage.

The testbench for the demo simply cycles through the states as a simple sanity check to output 0x0073 represented in the signal used for the seven-segment displays. The waveform below has been edited illustrating the state changes, and the expected output.

