**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.

We made the test bench by testing the different functionalities with read enable and write enable using the write enable first, then the read enables. One of the challenges that we have come across was understanding what the Data\_width should be because of the example that was given. We were unsure on the reasoning as to why it was set at 48 bits instead of 16 but we came to the understanding that the example was talking about 3 different sections of 16 bits that equaled up to 48 bits.

Following the lab specifications, we created a FSM wrapper to test our Memory Access Interface. First, we start by resetting everything in memory to be 0. Next, we store the value 2 into the addresses 0, 1, 2, 510, 511, 512, and 513. After all 2’s are stored, we modify the values by adding 69 to them and we store these values into addresses 0, 1, 2, 3, 510, 511, and 512. To ensure correctness, we read from those addresses to verify that the new value 71 is stored into each of them. Finally, we read the value from address 3 and display the value on the seven-segment display.