COE3DQ5 – Lab # 4 Report

Group # 33
Fahad Mahmood – 001414984 – mahmof4@mcmaster.ca
Wei Che Kao – 001328256 – kaow@mcmaster.ca

October 16, 2018

Exercise 1:

Firstly, address a is initialize to 0 and will always increment by 1 until it reaches 512. We also initially assign address b to 256 since X[i+256] and W[i+256] starts at address 256. For the first case, I < 256, address a and address b will keep increment until address a is 256. For case I > 256 address a will keep increment until 512 and we will set address b to 0 because X[i-256] and W[i-256] will start at 0 position again. We also assigned

```
// Y[i]

assign write_data_a[0] = read_data_a[0] + read_data_b[1]; // Y[i] when i < 256

assign write_data_a[1] = read_data_a[1] + read_data_b[0]; // Y[i] when i >= 256

// Z[i]

assign write_data_b[0] = read_data_a[1] - read_data_b[0]; // Z[i] when i < 256

assign write_data_b[1] = read_data_a[0] - read_data_b[1]; // Z[i] when i >= 256
```

We also verify the calculation on the ModelSim and both the addition and subtraction work correctly.

Exercise 2:

For exercise 2 we extended our experiment 4 SRAM_BIST.v file. Firstly, for even locations we created S_EVEN_WRITE_CYCLE and S_EVEN_READ_CYCLE state and two even delay states between them using our understanding from the lab. In order to change the address lines in increasing order in even write cycle we incremented our address by 2 decimals until we hit BIST_address == 18'h3FFFE (262142) since we were only checking even locations, then we went into our delay states where we decremented the addresses. In even read cycle, we compared our read data and write data same as our lab to find if there was a mismatch and according to requirements from the lab we decremented our address until we reach BIST_address == 18'd0. Secondly, we used delay 1 and delay 2 states from our lab to compare our read data and write data to find a mismatch. Then, we created S_ODD_WRITE_CYCLE and S_ODD_EVEN_CYCLE and two odd delay states between them. Similar to even states we incremented or decremented our address according to lab requirements. In odd write cycle we decremented addresses until we hit BIST_address == 18'd1 since now we are working with odd cycles. In read cycle we incremented address we hit BIST_address == 18'h3FFFF (262143). Between these states we added two odd delays same as even cycles where we incremented addresses. Lastly, we used delay 3 and delay 4 states from our lab to compare our read data and write data and finished the program by going back to idle state. We verified this to be working using ModelSim.