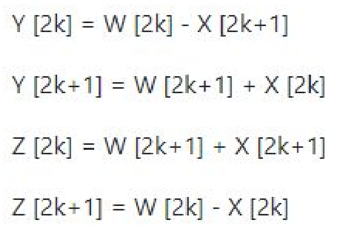
Lab 4 Report - Exercise 1

Frank Shan
Mark Naguib

For exercise 1, we defined write_data_a as Y, write_data_b as Z, read_data_a as W, and read_data_B as X. Thus from the equations given (pic to the left), we would get as follows. (pic to the right)

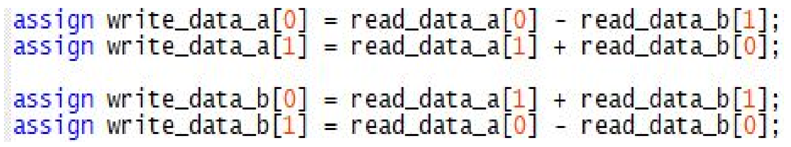


$Y[2k] = W[2k] - X[2k+1]$

$Y[2k+1] = W[2k+1] + X[2k]$

$Z[2k] = W[2k+1] + X[2k+1]$

$Z[2k+1] = W[2k] - X[2k]$



```
assign write_data_a[0] = read_data_a[0] - read_data_b[1];
assign write_data_a[1] = read_data_a[1] + read_data_b[0];

assign write_data_b[0] = read_data_a[1] + read_data_b[1];
assign write_data_b[1] = read_data_a[0] - read_data_b[0];
```

Since the assignment of write_data_a and write data b are outside of the always_ff block, the write data is getting constantly updated, and overwriting the corresponding elements. Since there is only 8 bits in Y, W, X and Z, overflow are ignored if occurred during addition/subtraction.

A new state called S_READ_WRITE2 was created to separate the write enables of the A DP-RAM and the B DP-RAM. It is not allowed to have the enables of both SRAMs to be writing at the same time and same location so the first state enables the write enables of A and disables the write enables of the B DP-RAM. The next state does the opposite by enabling the write enables of the B DP-RAM and disables the write enables of A. The program goes through the two states S_READ_WRITE and S_READ_WRITE2 to get to 511 where it then goes to the S_LAST_WRITE state where everything is set back to 0 and sent to S_IDLE.