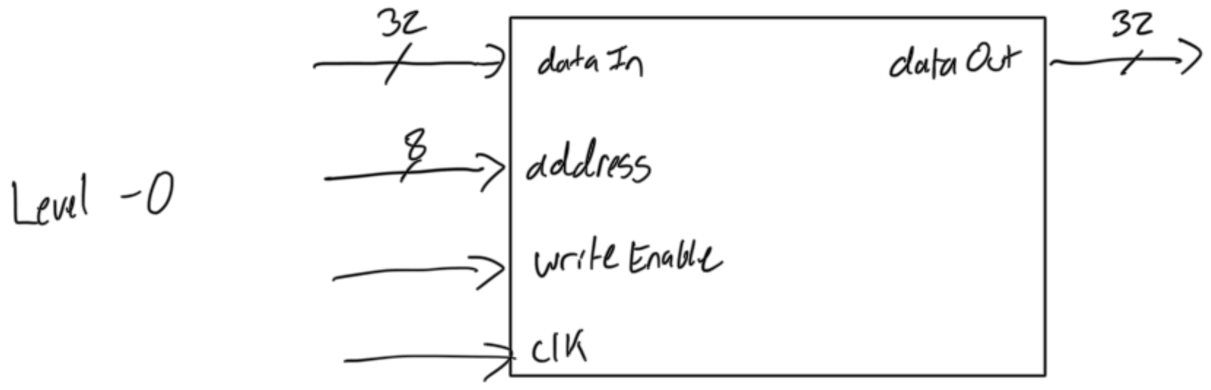
Ethan White

Lab 5 — RAM (256x32)

Objectives
For this lab, I am creating a RAM module (data memory) as part of a CPU to store 256x32 bits.

Introduction
RAM is an essential part of the CPU. A cpu doesn't operate on data which is in storage, instead it operates on data stored in much faster, although smaller, RAM modules. Each time a computer has an operation to perform, the data required for the operation is moved from storage to RAM.

Methodology
The following is a level -0 block diagram of my RAM. It has a 32 bit dataIn input, an 8 bit address input, a writeEnable and clk input, and a 32 bit dataOut output. On the positive clock edge, if writeEnable is high, whatever is on the dataIn bus will be stored in the RAM to the address given by the address input. At any point, the dataOut bus is equal to the data stored in the RAM at whatever address corresponds to the current address input.

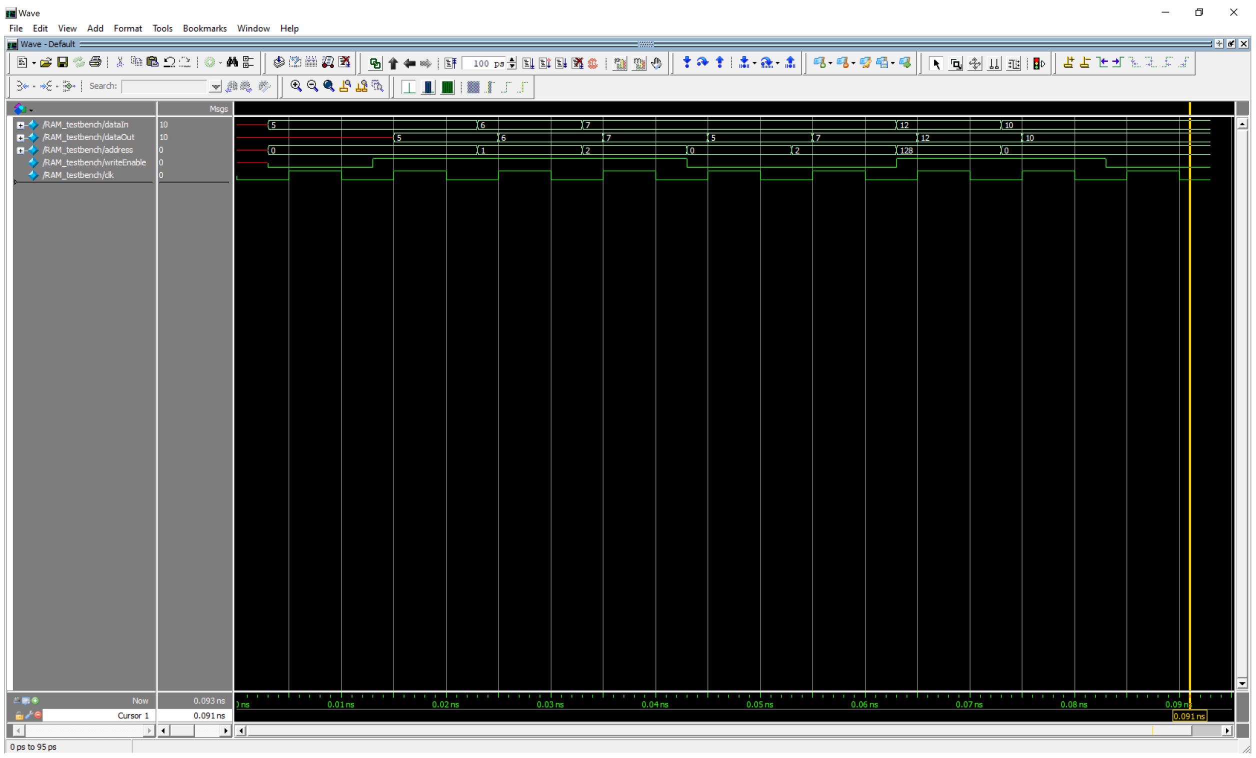


To create this RAM, I used an array of multi-bit registers rather than instantiate many modules or registers. Because of this clever Verilog trick, my RAM does not have a level -1 block diagram consisting of other modules. At the positive clock edge, the RAM module simply checks for write enable before setting the correct address in memory equal to dataIn, and then updates dataOut to the value in the correct address in memory.

Result and analysis
When tested in ModelSim, the RAM functions as expected.

All code, diagrams, and simulation results can be found as files on GitHub: https://github.com/Eth7an/Lab-5.git

A video explanation of the top level design and ModelSim simulation can be found here: https://youtu.be/AqZHj7a1LQA



```
/RAM_testbench/dataIn        10
/RAM_testbench/dataOut       10
/RAM_testbench/address       0
/RAM_testbench/writeEnable   0
/RAM_testbench/clk           0

Now        0.093 ns
Cursor 1   0.091 ns
0 ps to 95 ps
```

Discussion and Conclusion

To conclude, I have built a fully working and tested RAM module. The functionality of all inputs and outputs has been tested and verified, and this module is complete and ready to be instantiated in higher level projects.