Group 1

Our top level testbench file is very straightforward. First there is a clock that alternates high-low/low-high every 5-time units. This is the only input to the highest module (besides a PC counter reset).

We partitioned our 2^{16} word ram into two separate register banks. One register bank is dedicated to the instruction sets, while the is dedicated to data. The instruction register bank will continuously take the program counter as the address, then will output the next instruction (since it only contains in instructions). The register bank for data is accessed from the memory control module, where it will be read or written to if the opcode reflects a ldr or str cycle respectively. **Please note** for the purpose of the provided instructions, the instructions and data should be "muddied", as in data and instructions should be considered as one. To achieve this, both the data registers and instruction registers should be loaded with the instruction set, the uploaded files reflect this decision.

Once the new instruction is taken, it is first decoded by the register bank. The register bank will decode the destination address and prepare the result found later to be placed in the correct register. Next the register bank module will decode the two source registers and output their values using a MUX. They will accept the contents of the registers, as well as the instruction. The ALU will then execute based on correct operating principles. The exact execution of the ALU is beyond the scope of this briefing. In essence, it only activates the correct sub-module based on the opcode and if it satisfied the condition based on the flag value.

The ALU will then feed the output value to the memory control module. The memory control only interjects if the opcode indicates it is a LDR or STR cycle. If it is a LDR cycle, the memory control will read from the RAM based on the address outputted from a register in the register bank. It will take the value outputted by the RAM and feed this value to the register bank instead of the ALU value. The register bank will place this value in the "highlighted" register mentioned earlier. In the case it is a STR cycle, the memory control will send the correct data to the correct address in the RAM where the RAM will be written to the new correct value. If it is neither a LDR nor STR cycle, the memory control will only route the ALU output to the register bank where it will update the "highlighted value".

Once this whole instruction fetch – decode – execute cycle is complete, the clock will change, and the program counter will execute. This will fetch the new instruction where the entire cycle will restart.