CENG 450

Post-lab Summary

February 18th, 2014

Lab Session 3

Lab session 3 focused on the integration of the register unit and the ALU into a usable system. This system consisted of several busses which represented barriers at each stage of the pipeline. Before beginning to program any of these buffers, we first had to lay out the entire pipeline process in order to know what data needed to be carried where. We ended up with 4 buffers: Fetch/Decode, Decode/Execute, Execute/Memory and Memory/Writeback.

The Fetch/Decode buffer will receive input from an external source, such as memory, and receive an instruction. The instruction Byte is then split up into its component parts. For the a-type instructions we are working on implementing, this means, two registers and an opcode. The data is then passed to the Decode/Execute buffer each on its own bus.

The Decode/Execute buffer receives input from the previous buffer and sends the information to the ALU. In addition, this buffer also sets the writeback and memory flags. If the WB flag is set, the information coming out of the ALU or memory will be written back to the register. The MEM flag indicates that a memory operation will be performed. These flags are set using *if* statements comparing the opcode.

After the ALU executes its instructions, the result is passed to the Execute/Memory buffer, along with the MEM and WB flags from the previous buffer. This buffer contains no logic and exists simply to store information for the clock cycle.

Finally, The Memory/Writeback buffer receives the result from the instruction and passes the data through to either the Register unit for writeback, or to an external port. The Writeback register is passed through each buffer so that it travels with the instruction. The writeback can be performed at the same time as the Register unit is reading, as long as there is no conflict with the particular register.

Each of these buffers load the current value on their inputs to the corresponding output pins on the clock’s rising edge. Input of each register is received asynchronously.

This still needs to be tested, however at this time we have no way of accessing memory, or avoiding hazards in the pipeline. The test will have to be programmed to take this into account.