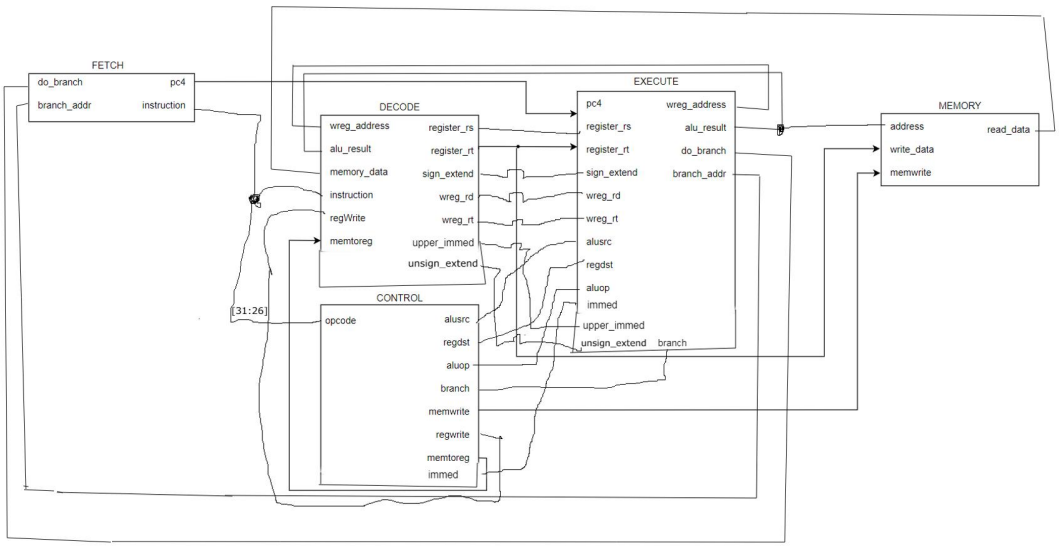

Figure 1. Datapath diagram. I have added lines directly from mips.v and added a few new signals (immed, unsigned_extend, upper_immed) that will be used to implement the ori and lui operations. The branch on the execute block is an input but had to be moved to the bottom of the block due to lack of space.

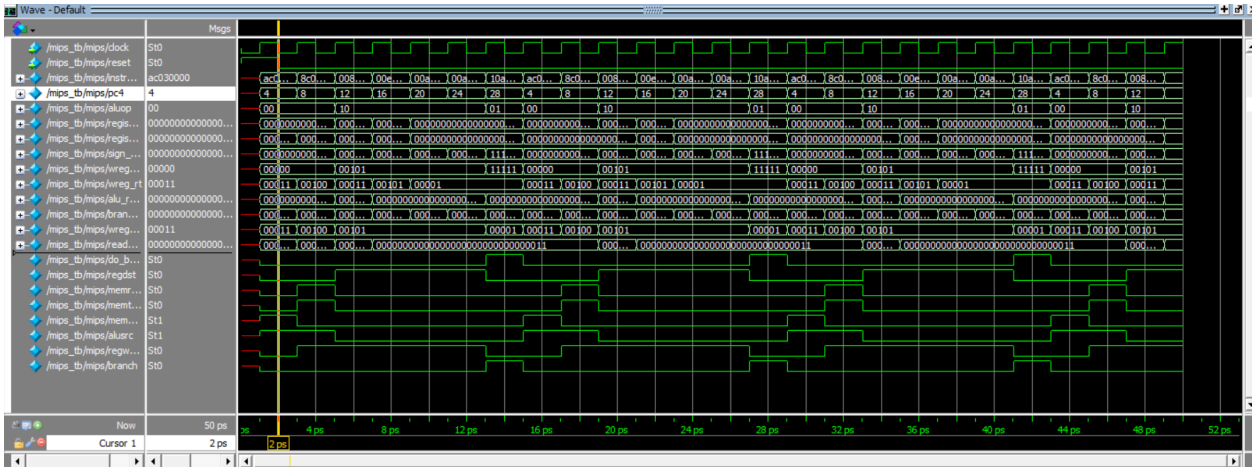

Figure 2. Waveform for the first run of mips_tb.v. This is the waveform resulting from running init.do and compile.do (equivalent to running the makefile).

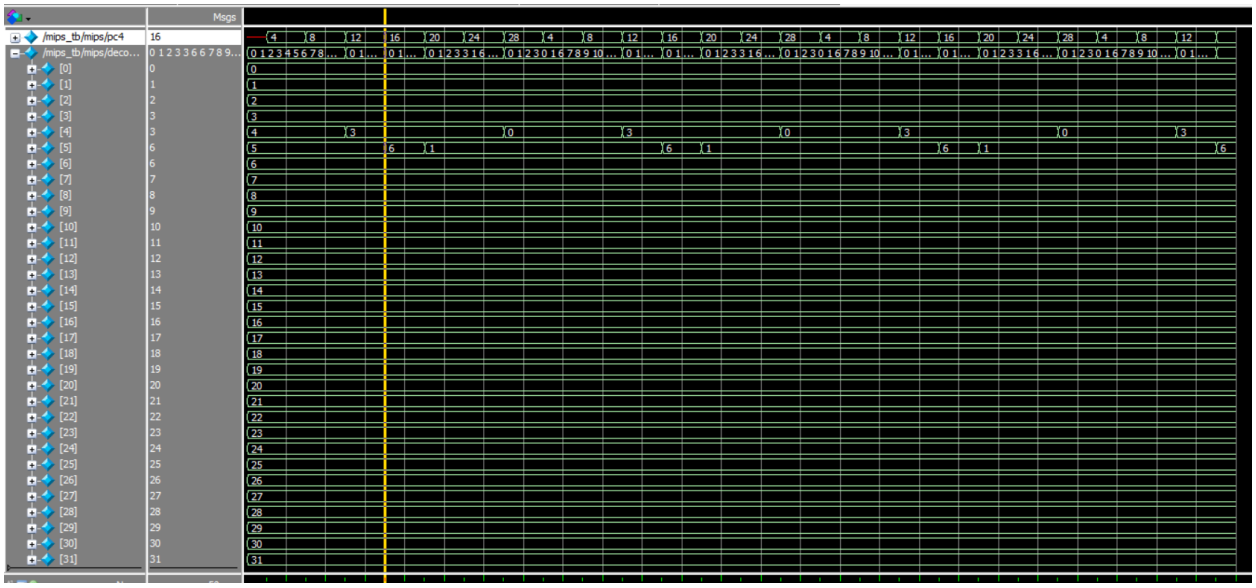

Figure 3. Waveform showing that slt is working. In this waveform, on the fifth instruction, it is comparing $5 to $0 and setting $4 to 1 if $5 is less than $0 or setting $4 to 0 if $5 is greater than $0. Since $5 is greater than $0, $4 is being set to 0.

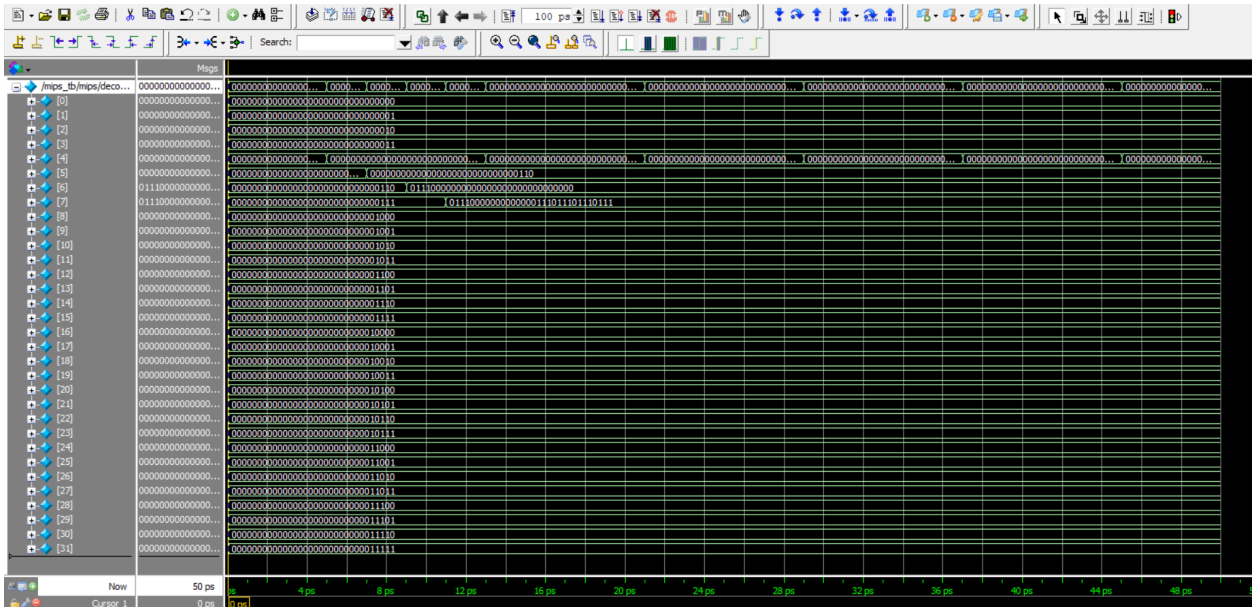

Figure 4. Waveform showing that ori and lui is working. In this waveform, on the third instruction, it is loading the value 0x7000 (binary 0111000000000000) into register $6. As you can see, the bottom bits of the number are filled with zeros just as the lui operation should do. On the next instruction after the lui operation, the value of 0x7777 (binary 0111011101110111) is the immediate in the ori operation with register $6. The immediate value of the ori operation is not sign-extended and is instead filled with zeroes.