IF – The IF module that we implemented will take care of the Instruction Fetch stage of the MIPS instruction pipeline. The purpose of the module is to fetch the instruction opcode, which identifies the instruction is being executed, from the instruction memory. The other responsibility of the IF module is to increment the Program Counter (PC) each time an instruction is executed. Our implementation increments the counter by a value of 1 each time an instruction is implemented.

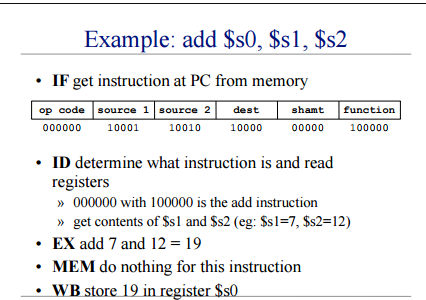
ID – The purpose of the ID module is to complete the Instruction Decode stage of the MIPS instruction pipeline. The main purpose of this stage of the pipeline is to simply figure out what the instruction returned by the IF stage of the pipeline is supposed to do. The ID module figures out what the instruction is from the opcode and breaks apart the instruction code into its different parts. This step also decides what registers are required and can make this decision before the instruction is fully decoded due to the simple instruction format it is given.

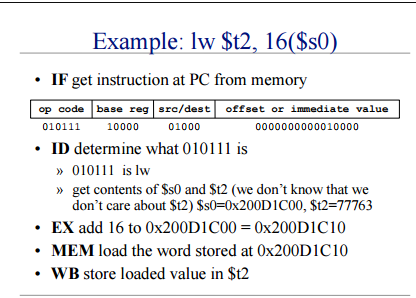
EX – The main purpose of the EX module to do the calculations that are required for the connection. This includes doing the arithmetic calculations required in mathematical instruction, like add or sub, and calculating memory references or adding up base and offset values. Our implementation uses a 32bit ALU that we created to calculate these values.

MEM – The MEM module in our project handles all the memory access required for MIPS commands. This module is used if an instruction needs to load, store, or access data that is stored in memory. This stage of the pipeline also replaces the PC with a destination address if the instruction a branching instruction. If the instruction does not require either of these functions, then this module does nothing.

WB – The purpose of the WB module is to handle the Write Back part of the MIPS pipeline. This module has one simple function. This is to place the results of the instruction in the appropriate register, which is designated in the instruction.

Here are a couple examples





<https://courses.cs.washington.edu/courses/cse410/05sp/lectures/cse410-10-pipelining-a.pdf>

Conclusion – We were able to successfully implement and test all five stages if the MIPS instruction pipeline successfully, however, the MIPS processor that we implemented does not work completely right. We created test benches for the five stages of the pipeline and we are convinced that they are working correctly. The test benches and waveforms created from the modules are included in the appendix of the report.