Milestone 1:

The CPU of choice is an accumulator because it only has one register, do not need any bits to store the register, and it is faster than load word. Encountered first problem, cannot compare two instructions, must think of ways around this. Another problem occurred trying to implement shift with the ALU. Confused at where to put the parameters in the code for the algorithm. We have decided to use hexadecimal for machine language, it would be easier to keep track of and understand.

Milestone 2:

Discussed in the meeting on 10/12/14 was the new amount of registers and resizing the instructions. We have settled on using 2 registers and making instructions 16 bits. Also, the instructions list has been revisited and several were deleted after discussing their usefulness.

Discussed in the meeting on 10/13/14 was how to make our datapath. It was decided to make a multi-cycle processor. The datapath was drawn in order to assist in the writing of the RTL code.

Discussed in the meeting on 10/14/14 was more of the datapath. It was decided that all of the parts of a traditional multicycle datapath would be needed for our datapath. The different control signals were also decided. We decided we did not need PCWriteCond, IorD, MemWrite, and RegDst because we are only using two registers. A new signal called WriteBlocker was added. We also discussed how to write the RTL code. Complication arose in converting the multicycle RTL to RTL that uses our types, our size of 16 bits, and our choice of fewer registers. RTL code was written for every instruction or set of similar instructions.

Milestone 3:

Discussed at the meeting on 10/20/14 was updates to Milestone 2. The RTL code was updated to reflect the meeting with Micah on 10/17/14. The datapath was also discussed. The datapath was completed with all the necessary parts and was begun to be implemented in Xilinx.

Discussed at the meeting on 10/21/14 was how the test cases would be implemented.

Milestone 4:

Most of milestone 4 was preparing the tests necessary to verify the control unit, the actual testbench was actually started early. Components takes unexpected long time to finish, also the testbench. The main problem is focusing on the testbench. There were 3 short meetings, one on 10/24/14 to finish the code for the control and create the symbols for the various components. The second meeting 10/26/14 was to finish up the specifications, and another short meeting on 10/27/14 to wrap up all documentation.

Milestone 5:

Headed into Milestone 5 the control unit test was almost finished and most components had been successfully tested. On 11/3/14 the control unit test was finished and integration testing was updated. On 11/4/14 the ALU16 and RegFile were fixed, the complete Xilinx model implemented, and the system test plan was begun.

Milestone 6:

Integration testing was begun on 11/08/14. Changes were made to the RTL, control unit, and machine code. Integration testing was completed on 11/12/14. System testing was begun on 11/13/14 and is still in progress.