Having worked on a five-part project involving the conception, realization, and testing of a CPU datapath, first and foremost I learned of the interworking of computer hardware and how it is interfaced with software components. There were several architectural schemes to chose from, i.e. single-cycle, multi-cycle, and piplined datapaths. I chose to implement the single-cycle because it seemed the easier to understand and realize. This idea turned out to be somewhat misleading because of inherent timing problems associated with this type of design. Though the overall datapath is synchronous, certain segments of the datapath are not. This asynchronous operation introduced randomness which effected the accuracy of results. For example, when trying to read from and then write to the same register, it is not known whether the signals are being processed in the proper sequence.

 The second thing I learned was the VHDL (very-high-speed integrated circuit hardware description language) which, in my opinion, could have been an acronym for “very hard to learn”. Unlike a programming language, VHDL is geared toward producing hardware and thus is more restrictive. Many script which are perfectly acceptable in code are not necessarily reproducible in the hardware.

 If I once again had the task of building a CPU, I might possibly try an alternative architecture such as the multi-cycle datapath. This way the timing issue encountered in the single-cycle type might be avoided. Furthermore, given the proper amount of time, I would test each instruction to ensure correct operation.

 Anyone attempting to construct a datapath should first of all become familiar with VHDL. Many issues that arise simply involve a lack of knowledge of the syntax. Furthermore, a good understanding of the chosen datapath architecture is a quintessential part of the process.