Jeremy Barthélemy
ECE 645
Project

Montgomery Multiplication
Verification Report: Architecture 2

Strategy for Verification:
I was unable to verify the Processing Elements that I created, although Matt was able to verify his. I constructed Architecture 2 using the PEs that I developed, and perhaps there is an issue in them, thus the malfunction in my Architecture2.vhd circuit.
The method of verification was to run the software implementation of the Montgomery Multiplication scheme, to produce a relatively large list of test vectors, and then use these test vectors in single tests or in an array of test vectors (although we ended up not reading from the file but simply including an array as such in the testbenches). From here, the goal was then to compare our actual values produced by our circuits and ensure that the actual values matched the expected values for all cases, and, if this were not the case, to determine the root of the discrepancy and debug it as best as possible. Unfortunately, I ran out of time in the debugging phase, although to my knowledge the circuit is complete.

Highest Level Tested:
The entity tested was Architecture2.vhd. The testbench used for verification purposes was the arch2_tb.vhd file. The test vectors derived for the purpose of comparing actual results with the expected results are the tv_array1 and tv_array2 files. The result of these verifications was negative. I decided to test using Architecture2 because the added control logic of Top.vhd would slightly more complexity in the testbench and I wanted to at least verify the core function of the Montgomery Multiplier. Sadly, upon testing, I am getting strange values for almost all multiplications, and in the latest testbench I made, I am getting all of my values as undefined. In terms of the sources of the error, I'm not totally certain where it can stem from. I believe that I implemented the small D, E, and F entities properly, but perhaps there are some mistakes inside. Otherwise, I suppose the error lies in the communication between these entities with regard to the processing elements.

Lower Level Tested:
The PE developed by Matt was tested, but this was not the one used in my implementation. Initially, we wanted to develop the lower-level entities separately, then share them and determine whose lower-level entities had better performance and use them both in our larger implementations. This would, in general, require very small changes to the larger architecture, if any, to structurally port them in. Matt tested his PEs by testing the results of additions and the selection of outputs for the pevious clock cycle. The inputs were S_in, Y, and M.

Issues:

The expectation was that the tested architecture would output the same result as given by the test rogram provided. For neither Arch1 nor Arch2 was this the case. For this architecture the logic was written exactly as described by the Huang IEEE document but the results never matched. Very much of the available time was devoted to debugging this issue, but we were unable to fix it.