For this project (#3) I was given an asynchronous JK flip-flop to construct. Constructing the flip-flop itself was not enormously difficult; the real problem arises when creating the primitive state diagram. This is because there are eight possible states for it to be in without including an output (Q). Once Q is included this doubles the possible states to 16. Each state can have three possible directional changes. However only two states can change the value of Q. The possible states depend on the inputs of J, K, and CLK. This infers that states are listed as 000 through 111 in binary. With The given truth table as shown below in figure 1, it can be extrapolated that Q’s output will be high when JK is “10” and clock is high, or if Q is ‘0’ and JK is “11” when clock is high.

//Where Notation is Listed As\\

The entire system depends on the primitive state diagram (PSD), and if that fails, the system will fail within the next few steps of construction. If the PSD is incorrect, them the primitive state table will fail. If the PSD is somewhat okay and the PST is passable, then the merge diagram will fail. If the PSD, PST, and MD somehow are passable, the K-maps will show that there is an error.

Once the final state table has been checked and is verified as giving the correct outputs and is operable, the gate mapping of F0 and F1 can begin. In my case, F1 was shown to be the output Q. Once all gate equations had been found, the logic construction was simple to construct.

Included in this report are the following:

1. Primitive State Diagram
2. Primitive State Table
3. Merge Diagram
4. Final State Table
5. Final K-Map
6. Output Mapping (Logic Mapping)
7. VHDL Code
8. Wave Form analysis
9. File Requested:: “asynsysvals.txt”