Alex Kahng, George Zhang
CS141 – Lab 3 Part 2
Design and Simulation Description
Due: Oct. 14th 2016

Design
We had to design a finite state machine. We realized while designing the FSM that we would need to edit our implementation of the tape. Thus we started by going back to the tape and instead of using additional regs to pass in the values to the register, we fed it in directly using combinational logic. In addition, we let our tape receive a head instead of moving it itself. In this way, we can more easily control the head in the FSM, and we no longer have a delay in writing to the registers.

We then went on to design the FSM. We gave it three tapes: 2 for input and 1 for output. To initialize, we simply waited until the user clicked the center button, at which point we kept track of where the head is and wrote each switch into the appropriate position on the tape. Another press of the center button will trigger the same process for the second tape. This then triggered the add flag in the FSM, which started the addition process. We drew out the state and transition diagram for the FSM and used it to construct a transition table. Using this, we were able to construct logic circuits for the addition. By keeping track of the states in regs and passing the combinational logic into the third tape, we were able to write the sum into the tape. Finally, we made one last pass through the third tape to read the tape to the LEDs. We implemented equal by setting the equal flag high and dropping it low if we ever see that the tapes are different.

Simulation
To simulate our FSM, we took two numbers and set center button / ctr appropriately to get the FSM to write the two numbers to the tape. We then waited an appropriate amount of time and then checked the output of it via the waveform diagram. We then triggered a reset, and confirmed that all the tapes and the LEDs were reset. We then input two other inputs (which are the same) in order to test the equal flag. After ensuring that the simulation proceeded correctly, we transferred it to the FPGA and confirmed that our program worked as expected. Using the FPGA allowed us to test faster, so we were able to test various corner cases, such as when the inputs are zero or all on.