Our FSM has 7 different states: INITIALIZE, INPUT\_0, INPUT\_1, ADD\_c0, ADD\_c1, DISPLAY\_SUM, DISPLAY\_EQUAL.

The zeroth state is INITIALIZE which will initialize the head, flags and the sate, and then transitions into the first state.

The first state is input\_0, which allows the input of the first number using switches to set each of the registers of the first tape. When the user has set the switches to the desired number, he/she must press and release the center button, and then the FSM transitions into the next state, INPUT\_1, which similarly allows for the input of the second number using switches to set each of the registers of the second tape.

When this is finished the user must again press and release the center button, and the FSM transitions into the state ADD\_c0, add with no carry (since there is no carry for the addition of the first bit). ADD\_c0 has 5 possible transitions, which are indicated in the spec. All of these transitions make sure that the corresponding output tape register is initially blank. 4 of these transitions reflect the 4 different combinations of 0 and 1 in two registers of each tape, which will cause the resulting addition to be stored in the corresponding register. 3 of these are self-loops (i.e. there is no carry) and the other case (the case when both registers are 1) will transition into the state ADD\_c1. The last transition of add\_c0 is when both registers are B (blank) indicating the end of the number, which causes a transition into display\_sum.

Similarly, add\_c1 has 5 possible transitions indicated in the spec- again, 4 corresponding to the 4 possible combinations of 0 and 1 in the two registers (three of which are self-loops and one- when both registers are 0- which goes back to add\_c0. Lastly, there is the case when both registers are blank, transitioning to DISPLAY\_SUM

DISPLAY\_SUM will display the output tape in the leds based on the values of each register, and this is accomplished by iterating through each of the registers of the output tape. DISPLAY\_SUM will transition into DISPLAY\_EQUAL if switch[7] is high. DISPLAY\_EQUAL will show all 0’s on the leds if the two input tapes are not equal and all 1’s on the leds if they’re equals. It will transition back into DISPLAY\_SUM if switch[7] is toggled again, and the sum will again be displayed on the leds.