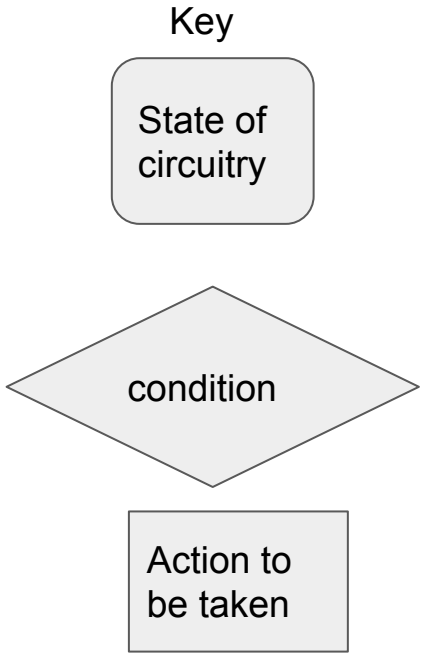
Final Project: Report and Flowchart

By: Ali Areiqat and Mario Han

Notes

1. The buttons for change, clr and ent use a debouncer to generate one clock pulse output.
2. The SSD is used to drive the segments and display the different messages like CLSd and OPEn.
3. VGA is used to print coloured characters on coloured background on screen.
4. We used 3 different clock dividers, one for buttons, one for display and one for the backdoor.



Start Here

Circuit in IDLE state
Displays ‘CLSd’

Press Ent(BTN2), enters GETFIRSTDIGIT.

Ready to Input Password

Forgot Password?

Yes

Set switches SW[8:5] to 1010 which corresponds to A

Press ent(BTN2) 6 times.
A counter variable (bdcount) is used to keep track of the number of enters.

System enters BACKDOOR state.

SSD displays a sliding:
“ I LOVE EC311
starting with “ I” and ending with “1 “
(text moves at 0.25sec freq).

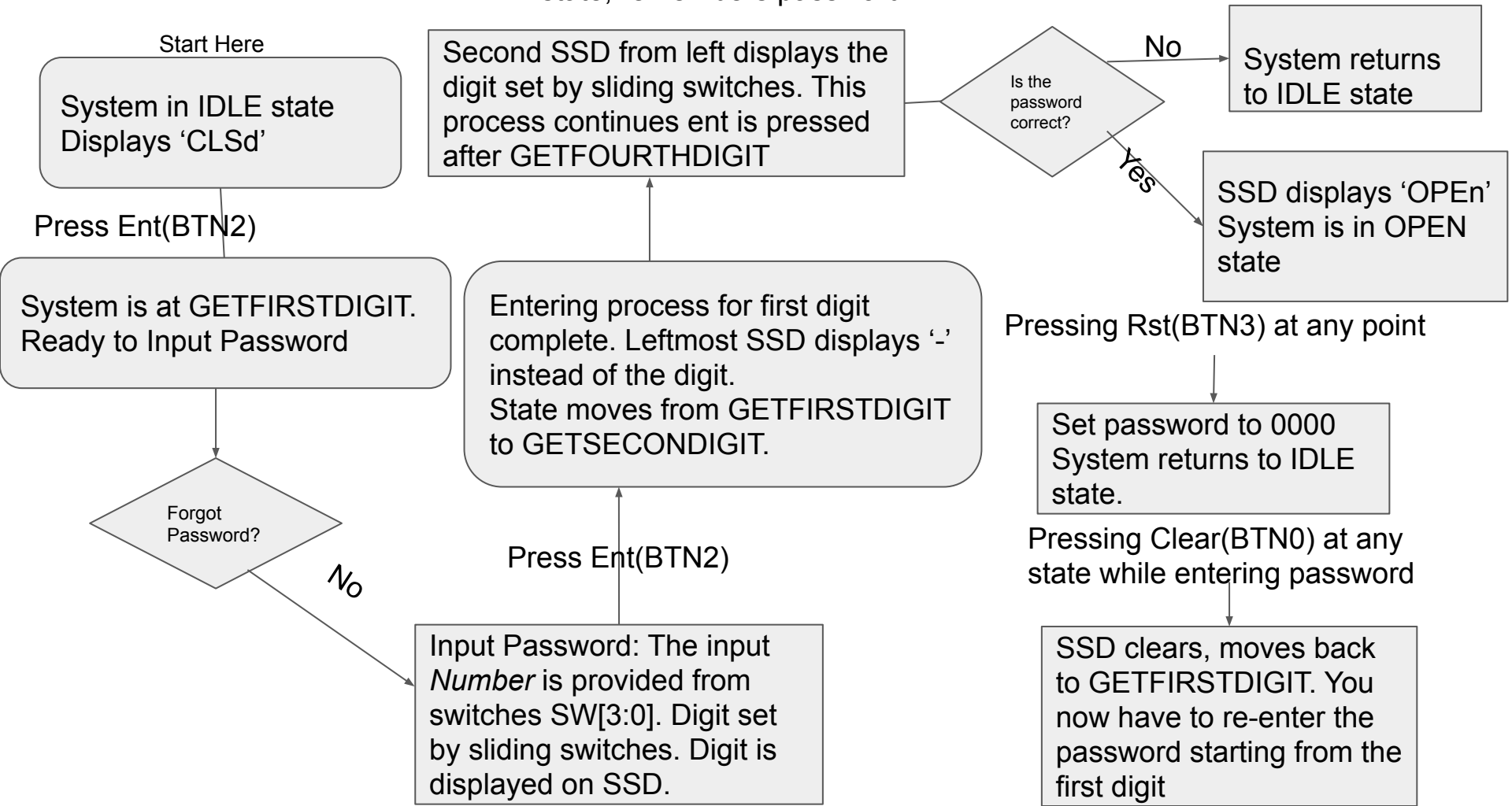
Sets password to 0311 and goes to IDLE State

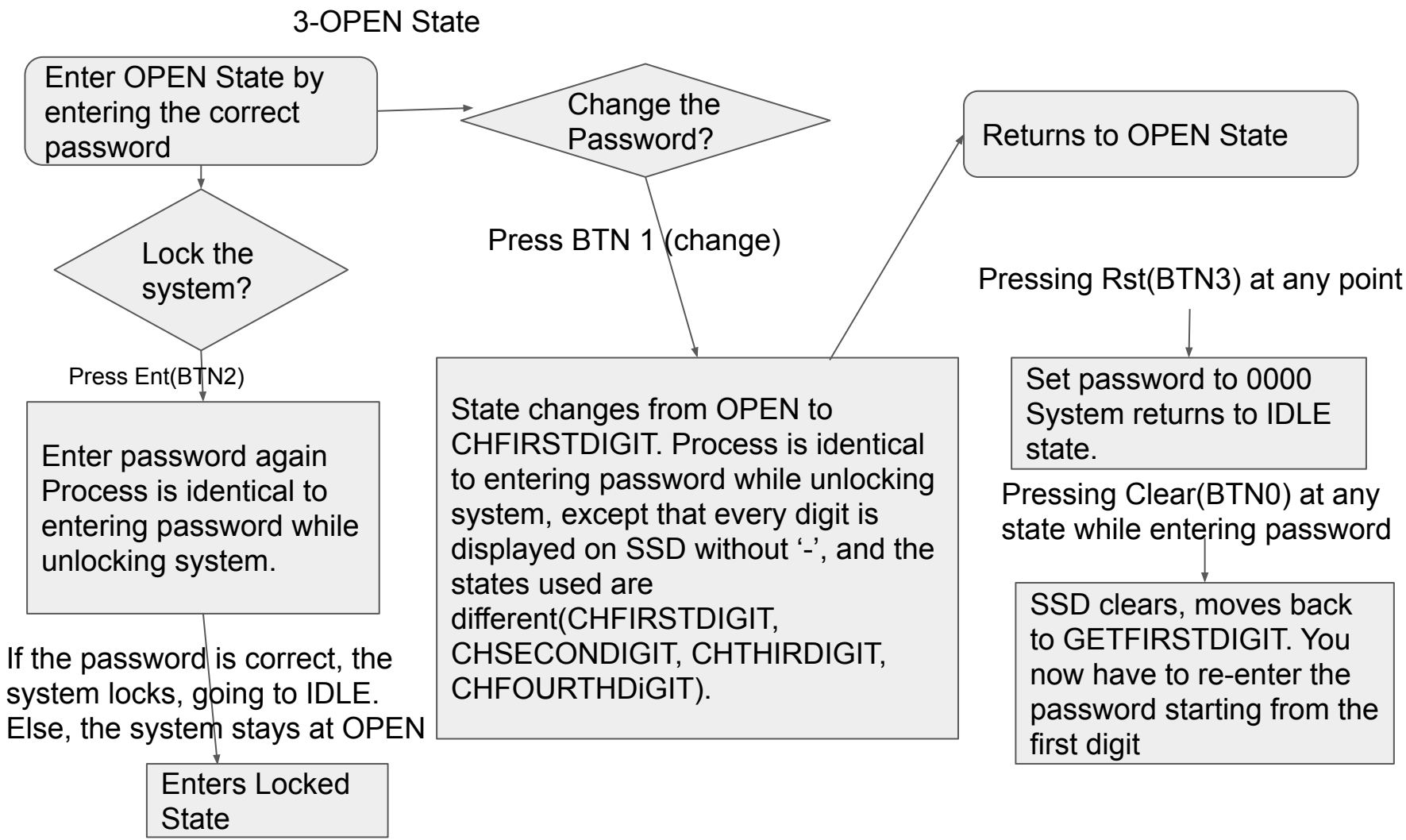
1- IDLE state, forgot password.

Pressing Reset(BTN3) at any point

Set password to 0000
System returns to IDLE state.

2 - IDLE state, remembers password.




3-OPEN State
Enter OPEN State by entering the correct password
Change the Password?
Returns to OPEN State
Lock the system?
Press BTN 1 (change)
Pressing Rst(BTN3) at any point
Press Ent(BTN2)
Enter password again Process is identical to entering password while unlocking system.
State changes from OPEN to CHFIRSTDIGIT. Process is identical to entering password while unlocking system, except that every digit is displayed on SSD without '-', and the states used are different(CHFIRSTDIGIT, CHSECONDIGIT, CHTHIRDIGIT, CHFOURTHDiGIT).
Set password to 0000 System returns to IDLE state.
Pressing Clear(BTN0) at any state while entering password
If the password is correct, the system locks, going to IDLE. Else, the system stays at OPEN
SSD clears, moves back to GETFIRSTDIGIT. You now have to re-enter the password starting from the first digit
Enters Locked State

General Notes

1. We used the combinational part of the circuit, solely to change states. The combinational part controlled the states which we are at (current_state) and which state we will go to (next_state). There were no assigning of values or anything of the like here.
2. The sequential part was used for changing the values of the counters, changing the values of password and inpassword, assignment of different values and any operation that isn't changing states.
3. A module seven_segments along with its submodule binary_to_segment was modified to be able to process all values that we need to show on the SSD.
4. We used three clocks. The one we used for the button was slower than that of the backdoor and display clocks, because the button relies on human input, so it shouldn't be too fast or too slow.

Notes: OPEN State

1. We used a counter variable(counter_open) to keep track of whenever the system is open. Whenever the system opened, counter_open = 1 and whenever the system went from OPEN to IDLE, counter_open = 0.
2. Pressing change(BTN1) takes the system to CHFIRSTDIGIT where you can enter a digit, which is saved automatically as part of the password. This continues for CHSECONDIGIT, CHTHIRDIGIT AND CHFOURTHDIGIT. During the entry of the digits, all digits would be displayed clearly on the SSD. They will not be replace with a - like while unlocking the system.
3. Pressing ent while its open, moves us to GETFIRSTDIGIT and sets a counter counter_locker = 1. Once the inpassword is inputted and it matches the password, counter_locker tells us that the system was open, so it is set to zero and the system goes to IDLE. If in password is wrong, the systems stays in OPEN and counter_locker is set back to 0.

Notes: IDLE State

The systems starts out in IDLE state, displaying CLSd on the SSD. Pressing ent(BTN2) moves it to GETFIRSTDIGIT, where you can use the switches to change the input digit you want. Pressing ent again moves the system to GETSECONDIGIT and the digit inputted is saved as part of a 16 bit variable inpassword. This process continues through GETTHIRDIGIT and into GETFOURTHDIGIT. After inputting the fourth digit and pressing ent, our inpassword is complete. Inpassword is compared to password, if they match the system moves to OPEN, otherwise, it moves back to IDLE.

On the SSD, the current digit is displayed, while the previous inputted digit is covered by a -. For Example, at GETTHIRDIGIT, the SSD will look like this “--1 “

Notes: Backdoor 1

We have dealt with a lot of trouble because of the backdoor. Several times, we had code that should've worked theoretically but it never did, so we had to redo it several times.

The way it works is that we have a counter bdcount that starts at 0. When switches[8:5] are switched to 1010, representing the hexadecimal character A, the bdcount activates, and starts counting whenever ent(btn 2) is pressed. This happens using if statements that are within the states we used to get the digits(GETFIRSTDIGIT, GETSECONDIGIT, GETTHIRDIGIT, GETFOURTHDIGIT).

After enter is pressed 6 times. The SSD starts displaying a sliding text that says " I love EC311" It moves 1 "digit" at a time at a frequency of 0.25seconds starting from " I" and ending with "1 ". The sliding happens as a result of a clock for the backdoor driving a counter. The counter goes through different case statements which say what the SSD should look like, and that is what results in the sliding.

Notes: Backdoor 2

After the message finishes displaying, bdcount is set to 0, the password is set to 0311 and the system returns to IDLE.

At least theoretically, this is what our code should do but in practice, the bdcount till the BACKDOOR activates changed everytime we pressed rst(BTN3) and tried again, and we could not find the cause of this error. We suspect it has to do with the clock, but we couldn’t fix it. The text displays perfectly, but there is a problem with the timing of the buttons, so it never properly works.

Notes: VGA

Normally, the VGA would be used to display everything on the SSD on the screen, but the background is red and the text is yellow. We neither had the time nor the expertise to finish this part of the project.

The main problem we had while approaching this part, is not knowing where to start working on it, and as such, we did not have enough time to do it by the end.