

**NAME: Sai Srinivasa Haranadh Baba Singamsetti**

**UB NUMBER: 50314313**

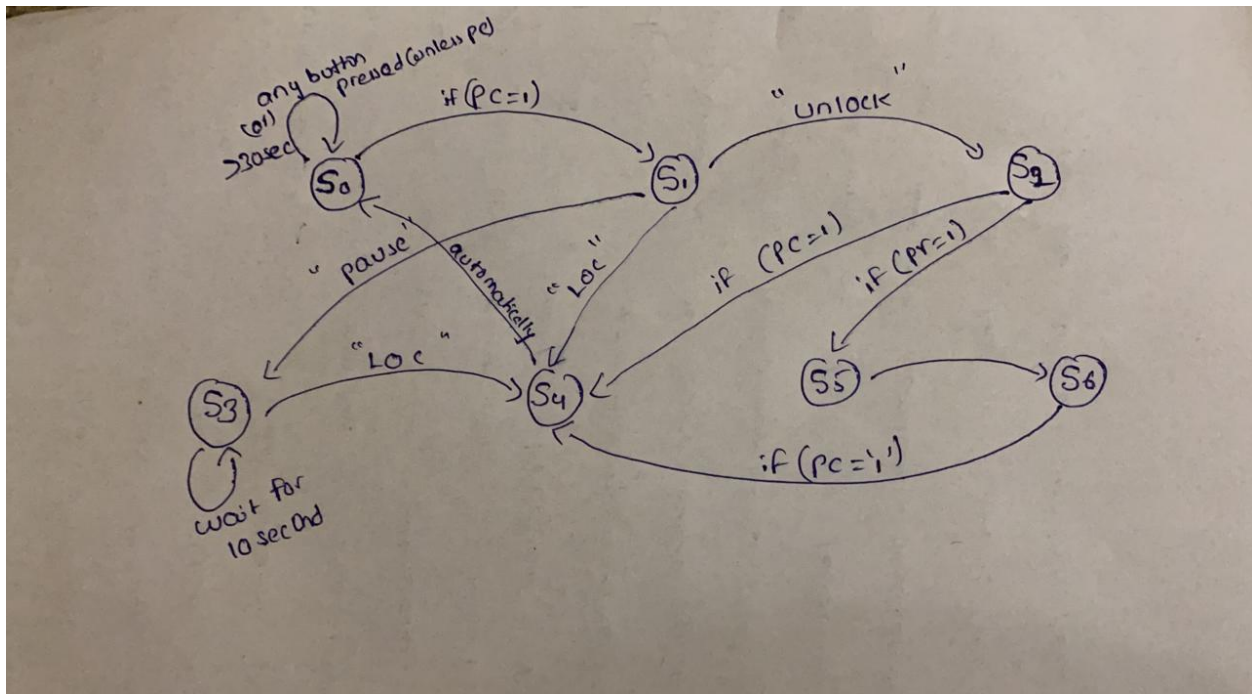
## **COMPUTER ARCHITECTURE**

### **ELECTRONIC LOCK**

#### **Project summary:**

Design an electronic combination lock. A four-digit code (D3D2D1D0) can be entered by the push buttons and slider switches. Slider switches 0 through 3 are used to enter a BCD, while the momentary push buttons are used to latch the value on the switches. The left push button (L) is used to latch D0, the upper push button (U) is used to latch D1. The lower push button (D) is used to latch D2. The right push button (R) is used to latch D3. When the entire code has been entered, the center push button (C) should be pressed. When the correct combination is entered and the device unlocks, the LEDs will flash. If two attempts are made to unlock the device without correctly entering the combination, the device should be deactivated for 10 seconds. The seven segment display will be used to display the state of the lock. The states are: "LOC" (locked), "####" (Latched digits during an attempt), "UnLC" (Unlocked), "PAUS" (deactivated), or "CHnG" (Changing combination). For each unlock attempt, the entire code must be entered within 30 seconds (from the first button press to the last button press), or the attempt resets and seven segments show "LOC" again. Note that this does not count towards incorrect attempts. When the device is unlocked, it can be locked again by pressing the center momentary push button (C), or the user can change its combination by pressing the right momentary push button (R). After pressing the button, the seven segments should show "ChnG", and the same procedure for unlocking is repeated. After changing the combination, the device should immediately go to locked state. The default combination for your lock should be the last 4 digits of your UB person number

## DOCUMENTATION



### State diagram explanation:

Initial passcode: **4313**(ub id last 4 digits)

#### State (s0):

- The FSM machine for lock starts from state(s0) with initially displaying "**LOC**" as output.
- The four segment displays are latched by four buttons **btnR**, **btnL**, **btnU**, **btnD**.
- The **timer** is started when **any button is pressed**. If **BtnC is not pressed** within **30 seconds**. it will **restart**.
- Whenever the **BtnC** is pressed the state changes to **state(s1)**.

#### State(s1):

- In **state s1** it undergoes comparison operation.
- If both lock code and code we entered are **same** it moves to **state s2**.
- If both lock code and code we entered are **not same** and **attempt equal to two (2)** it moves to **state s3**.
- If both lock code and code we entered are **not same** and **attempt not equal to two (2)** it moves to **state s4**.

#### State(s2):

- In **state s2** it displays "**UNLN**" along with **led flash**.
- If **btnC** is pressed it goes to **state (s4)**.

- If **btnR** is pressed it goes to **state(s5)**.

#### State(s3):

- In **state 3** it displays "**paus**" as output.
- The FSM moves to wait state for **10 seconds** in this state and moves to **state (s4)**.

#### State(s4):

- In **state s4** it displays "**LOC**" as output.
- It will display for two seconds and **automatically** moves to **state (s0)**.

#### State(s5):

- In **state s5** it will display "**CHNG**" as output.
- It will display for two seconds and **automatically** moves to **state (s6)**.

#### State(s6):

- In this we update **old passcode** to **new passcode** which are latched using all buttons. And displays the number.
- If **btnC** is pressed it will change to **state(s4)** which automatically moves to **state(s0)** after **2 seconds**.

The process repeats again until we shut down our system.

#### Code implementation:

Some inputs for better understanding how it work.

- After change(**s5**) state when it enters the **s6** where we can update the passcode. When it enters **s6** initially it will display **old passcode**.
- When **unlock** occurs led will **flash**.
- **State (s4)** acts as intermediate state for reset state. It will wait one sec in **s4** and **automatically** reaches **s0**.

#### Implementation of code:

- I have used states for implementing this project. They are totally 6 states in code with a clock divider
- Initially I stored the LOC in temporary variables **temp0, temp1, temp2, temp3** and unlock passcode in **temp4, temp5, temp6, temp7**.
- Then using **right, left, center, up and down** buttons I latched those temporary variables and parallelly displayed them.
- After pressing one if entire code is not entered and button center not pressed within **30 sec**. it again moves to starting state.
- Then after **pressing button Center**, it starts compare all temporary variables respectively.

- If they are **equal** it will display **UNLC** then if we press button center it will go to LOC state else if we press button right it will move to change state and we can update the values of new passcode by updating **temp4, temp5, temp6, temp7** and it reaches LOC state by pressing button center.
- If **comparison is false** and **attempt is not zero** it enters into **Loc state**. Else if **attempt is zero** it enters pause state and wait for **10 seconds** and it reaches **LOC state**.

#### Key points:

##### Note1:

- Don't press all buttons within a second. At least give one second gap for pressing each button.
- This is because **long pressing one button** sometimes may **skip states**. To make it more **comfortable and robust**, I added **timer of one second** in every state to remove that **ambiguity**.

##### Note2:

- Whenever the bit file is loaded the **30sec-timer won't work for the first time**. After that it will work for infinite number of tries until again we load our bit file.
- Precisely, I am saying that the 30sec-timer won't work for **first clock cycle**. It works from the **second cycle**.
- If we enter the **state of changing passcode**, initially it will display **old passcode**.

#### Problems Encountered and how they are solved:

- **Problem1:** The first problem is at beginning I used to implement use rising edge clock in every state which works but the output was not displayed.  
**Solution:** After having a talk with professor during office hours I realized to use that rising edge of clock only at beginning.
- **Problem2:** The second problem in my project lock is after pressing center button it should display unlock if the code matches and again, we press center button in that state it should go to LOC state. one button is used for shifting three states. Sometimes if we long press that button UNLN state will skip  
**Solution:** I used timer additionally in every state to make sure it not going to skip any state if we had a long press on button center for 2 or 3 seconds.

**I hope all these inputs and explanations and notes makes it easy how my code works**

