SECURITY PASSWORD ENTRANCE SYSTEM

DLD project

ALI AL-KELABI 1722625

Lecturers: Dr. YASIR BIN MOHD. MUSTAFAH, Dr. HAZLINA BT. MD. YUSOF

Section 1

Semester 1 2020/2021

Department of Mechatronics Engineering

International Islamic University Malaysia

Introduction

The incorporation of security lock and security systems is quite essential especially during these days, where simple key lock system is inappropriate since many people figured out some ways to decode and unlock these doors that have such implemented mechanical systems. Additionally, the fatigue of continuous usage of these lockers and key is going to cause them break down and possibly in the most urgent conditions. On the other hand, electrical locking systems are proven to sustain much longer time and are hard to decode unless the card you are using or the password you put in the system has been stolen, there you must change the card or password. But the general and overall view of the current used system, it is much better than old mechanical based systems.

Objectives

1-To build a security system that can compare between the saved password in the system and the inserted password at the input terminal.
2-The system will block the user if a wrong password is entered, where it will not activate the door.
3-The system is mainly implemented to be used in general life applications for security purposes, such as in entrances to staff offices, private rooms or even as a security system for houses.

Design process

I have used two software environments for two reasons, firstly to make the general design of the logic of the system that is by using Logisim, secondly to verify the designed circuit and make sure that it’s going to function in real applications that is by using Multisim.

The first one Logisim which helps in constructing the logic of my proposed system, it gives me two approaches one is the truth table method where you can write what your system will do with the outputs by conditioning them and giving them the values based on the probability of it to occur and then it can build the circuit based on that logic. However, sometimes the circuit is too long and complicated so by using k-maps it is possible to simplify it to a smaller more comprehensible logic circuit. The other way is to build your circuit manually by including the components and logic gates you have deduced from your truth table after doing the k-maps and got the logic equations.

The logic I have used in my system is to compare between two entered values, one from user and the other is from the owner of the system, it is exactly like password verification. So, after the password checking in the system, it will decide either to unlock the door or keep it locked based on the password entered if it is true the door opens, if it is false the door remains closed.

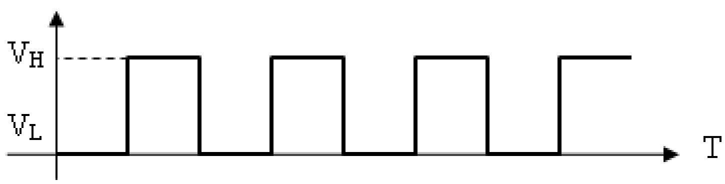
The second software Multisim, I have simulated the logic circuit I Logisim in Multisim with not much changes other than adding a ground to the output LEDs, which is a good thing in Multisim that the logic gates are similar in shape.

Detailed design

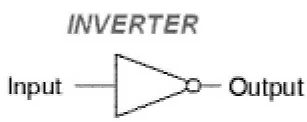
I have used the truth table in Logisim as an easier way to construct my design but as I said it required a lot of simplification from simple AND and NOT gates to XNOR gates and AND gates. The inputs to the system are 8, 4 for user entry and the other 4 are used as password setting, which is used by the owner. There is a switch to on or off the system and there are outputs as LEDs representing the password is true or false. All in total I have used 4 XNOR gates, 6 AND gates and 3 NOT gates. However, there is an improved design of mine adding 4 JK flip-flops to do a time counting for 4 seconds before the person can enter to avoid any collusions and notify if there was any person heading to the entrance from the other direction.

Constant: a value that is inserted to the logic gate and from it we get an output, it could work as an input to an a logic gate.

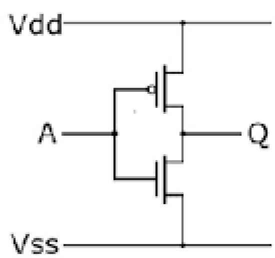
A clock: In electronics and especially synchronous digital circuits, a clock signal (historically also known as logic beat) oscillates between a high and a low state and is used like a metronome to coordinate actions of digital circuits. A clock signal is produced by a clock generator.



Inverter: An inverter circuit outputs a voltage representing the opposite logic-level to its input. Its main function is to invert the input signal applied. Digital electronics circuits operate at fixed voltage levels corresponding to a logical 0 or 1.



Input	Output
1	0
0	1

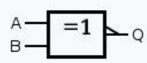


And gate: The AND gate is a basic digital logic gate that implements logical conjunction - it behaves as follows A HIGH output results only if all the inputs to the AND gate are HIGH. If none or not all inputs to the AND gate are HIGH, LOW output

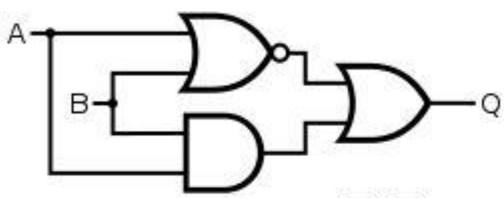
AND			$A \cdot B$ or $A \wedge B$	

INPUT		OUTPUT
A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

XNOR: is a digital logic gate whose function is the logical complement of the Exclusive OR gate. The two-input version implements logical equality, behaving according to the truth table to the right, and hence the gate is sometimes called an "equivalence gate". A high output (1) results if both inputs to the gate are the same. If one but not both inputs are high (1), a low output (0) results.

XNOR			$\overline{A \oplus B}$ or $A \odot B$	

INPUT		OUTPUT
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	1

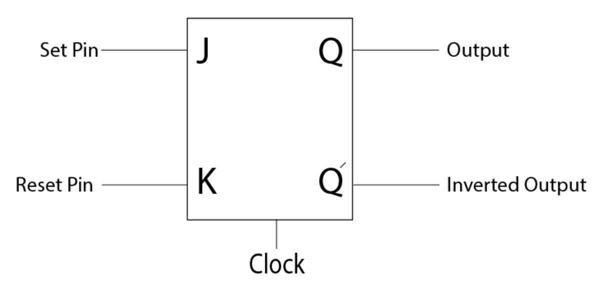


$$Q=(A\cdot B)+(\bar{A}\cdot\bar{B})$$

INPUT		OUTPUT					
A	B	AND	NAND	OR	NOR	XOR	XNOR
0	0	0	1	0	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	0	1	0	0	1

JK flip-flop: The J-K flip-flop is the most versatile of the basic flip-flops. It has the input-following character of the clocked D flip-flop but has two inputs, traditionally labeled J and K. If J and K are different then the output Q takes the value of J at the next clock edge.

JK Flip-Flop Symbol

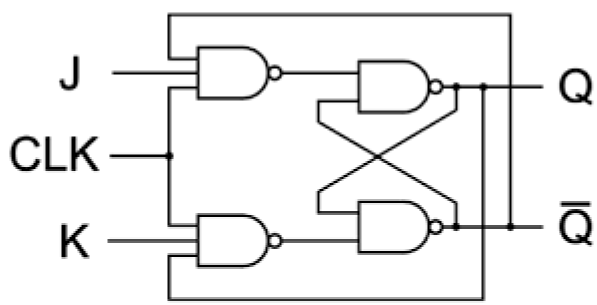


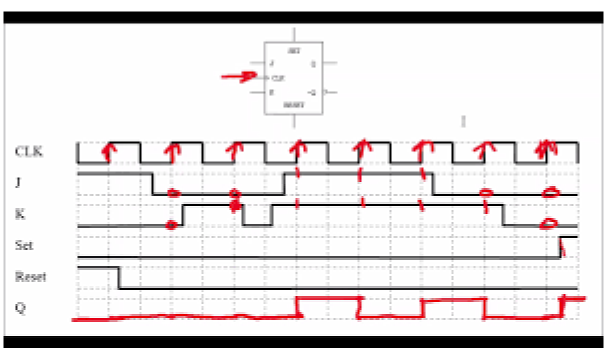
JK Flip-Flop Logic Table

C	J	K	Q	Q`
HIGH	0	0	Latch	Latch
HIGH	0	1	0	1
HIGH	1	0	1	0
HIGH	1	1	Toggle	Toggle
LOW	0	0	Latch	Latch
LOW	0	1	Latch	Latch
LOW	1	0	Latch	Latch
LOW	1	1	Latch	Latch

Introduction to JK Flip Flop

www.TheEngineeringProjects.com





The logic equations

In POS

$$x2\,(d+\bar{h})\,(c+\bar{g})\,(b+\bar{f})\,(a+\bar{e})\,(\bar{d}+h)\,(\bar{c}+g)\,(\bar{b}+f)\,(\bar{a}+e)$$

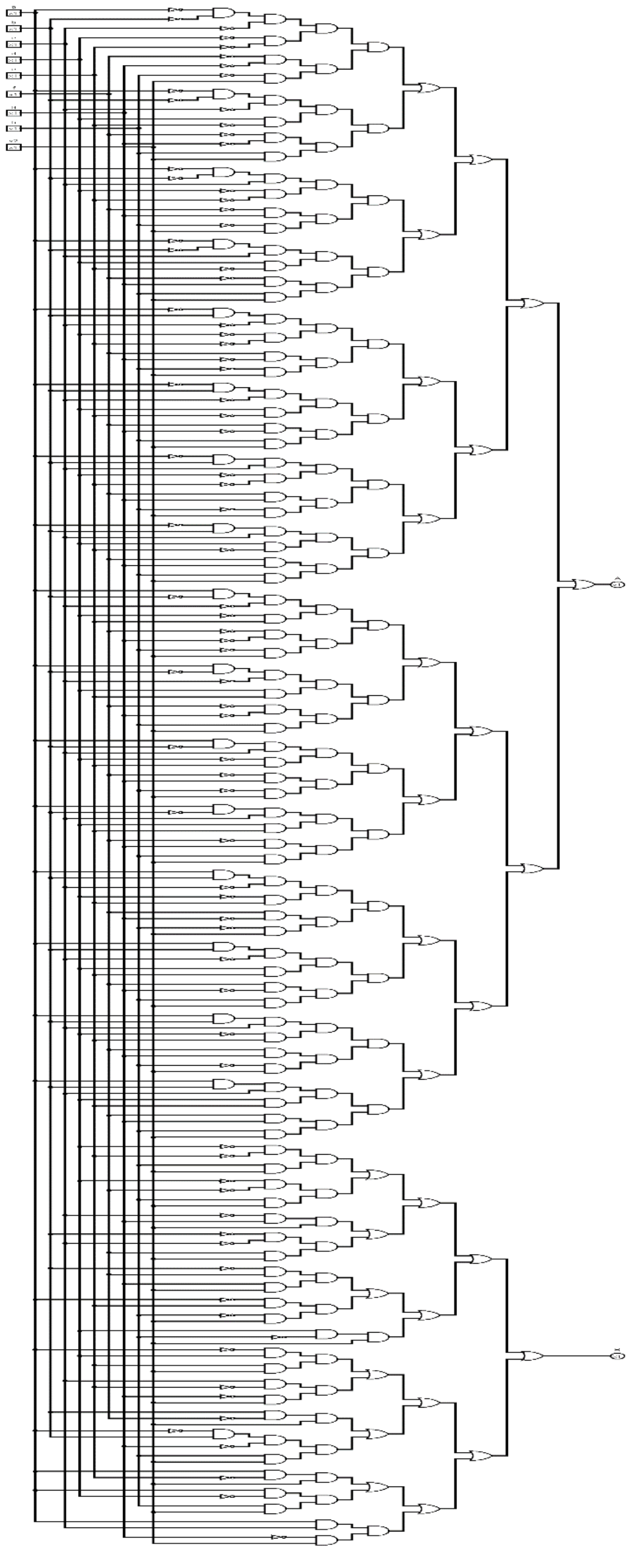
```
x2 (d + ~h) (c + ~g) (b + ~f) (a + ~e) (~d + h) (~c + g) (~b + f) (~a + e)
```

In SOP

$$\bar{a}\bar{b}\bar{c}\bar{d}\bar{e}\bar{f}\bar{g}\bar{h}x2+\bar{a}\bar{b}\bar{c}d\bar{e}\bar{f}\bar{g}hx2+\bar{a}\bar{b}c\bar{d}\bar{e}\bar{f}g\bar{h}x2+\bar{a}\bar{b}cd\bar{e}\bar{f}ghx2+\bar{a}b\bar{c}\bar{d}\bar{e}f\bar{g}\bar{h}x2+\bar{a}b\bar{c}d\bar{e}f\bar{g}hx2+\bar{a}bc\bar{d}\bar{e}fg\bar{h}x2+\bar{a}bcd\bar{e}fghx2+a\bar{b}\bar{c}\bar{d}e\bar{f}\bar{g}\bar{h}x2+a\bar{b}\bar{c}de\bar{f}\bar{g}hx2+a\bar{b}c\bar{d}e\bar{f}g\bar{h}x2+a\bar{b}cde\bar{f}ghx2+ab\bar{c}\bar{d}ef\bar{g}\bar{h}x2+ab\bar{c}def\bar{g}hx2$$
$$+abc\bar{d}efg\bar{h}x2+abcdefghx2$$

```
~a ~b ~c ~d ~e ~f ~g ~h x2 + ~a ~b ~c d ~e ~f ~g h x2 + ~a ~b c ~d ~e ~f g ~h x2 + ~a ~b c d ~e ~f g h x2 + ~a b ~c ~d ~e f ~g ~h x2 + ~a b ~c d ~e f ~g h x2 + ~a
b c ~d ~e f g ~h x2 + ~a b c d ~e f g h x2 + a ~b ~c ~d e ~f ~g ~h x2 + a ~b ~c d e ~f ~g h x2 + a ~b c ~d e ~f g ~h x2 + a ~b c d e ~f g h x2 + a b ~c ~d e f ~g
~h x2 + a b ~c d e f ~g h x2 + a b c ~d e f g ~h x2 + a b c d e f g h x2
```

Original circuit from the truth table

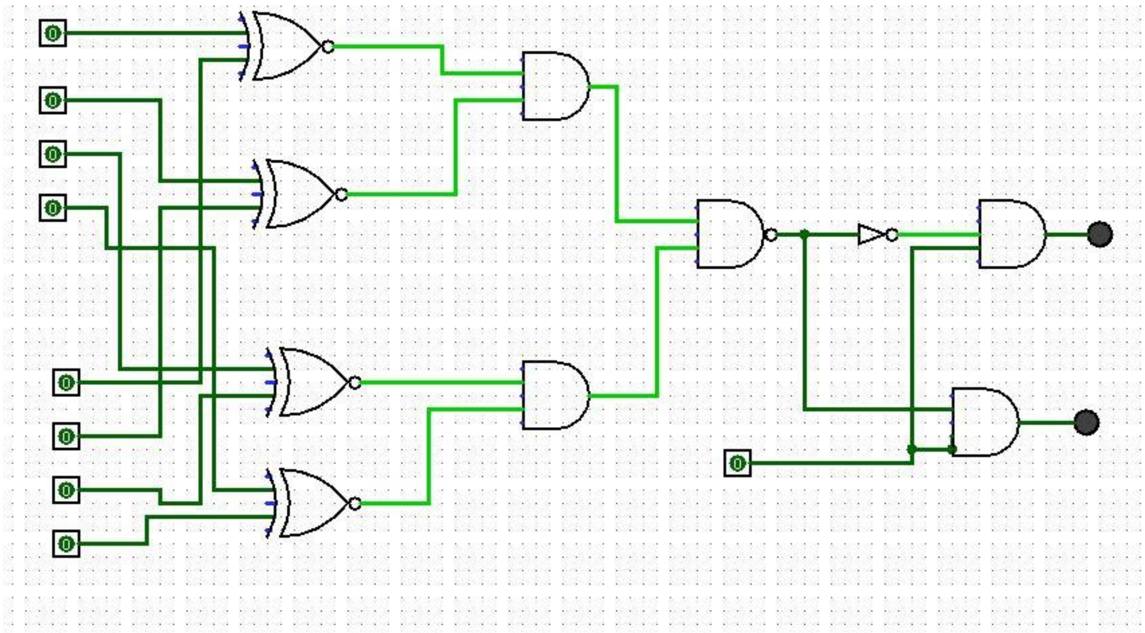


Truth table

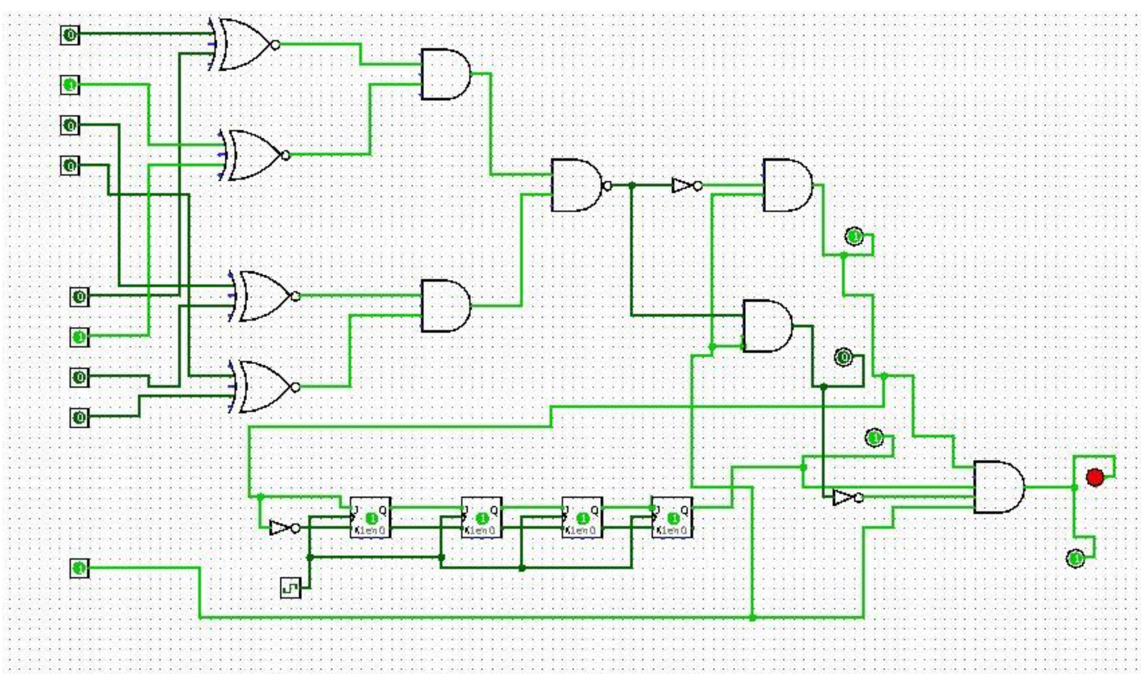
How ever it is much longer than this, but this is a small fraction of it.

a	b	c	d	e	f	g	h	x2	A	B
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1	1	0	1
0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	1	0	1
0	0	0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	1	1	1	0	1
0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	1	0	1	1	0	1
0	0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	1	1	0	1	0	1
0	0	0	0	0	1	1	1	0	0	0
0	0	0	0	0	1	1	1	1	0	1
0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0	1	0	1
0	0	0	0	1	0	0	1	0	0	0
0	0	0	0	1	0	0	1	1	0	1
0	0	0	0	1	0	1	0	0	0	0
0	0	0	0	1	0	1	0	1	0	1
0	0	0	0	1	0	1	1	0	0	0
0	0	0	0	1	0	1	1	1	0	1
0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	1	1	0	0	1	0	1
0	0	0	0	1	1	0	1	0	0	0
0	0	0	0	1	1	0	1	1	0	1
0	0	0	0	1	1	1	0	0	0	0
0	0	0	0	1	1	1	0	1	0	1
0	0	0	0	1	1	1	1	0	0	0

The simplified circuit of the system

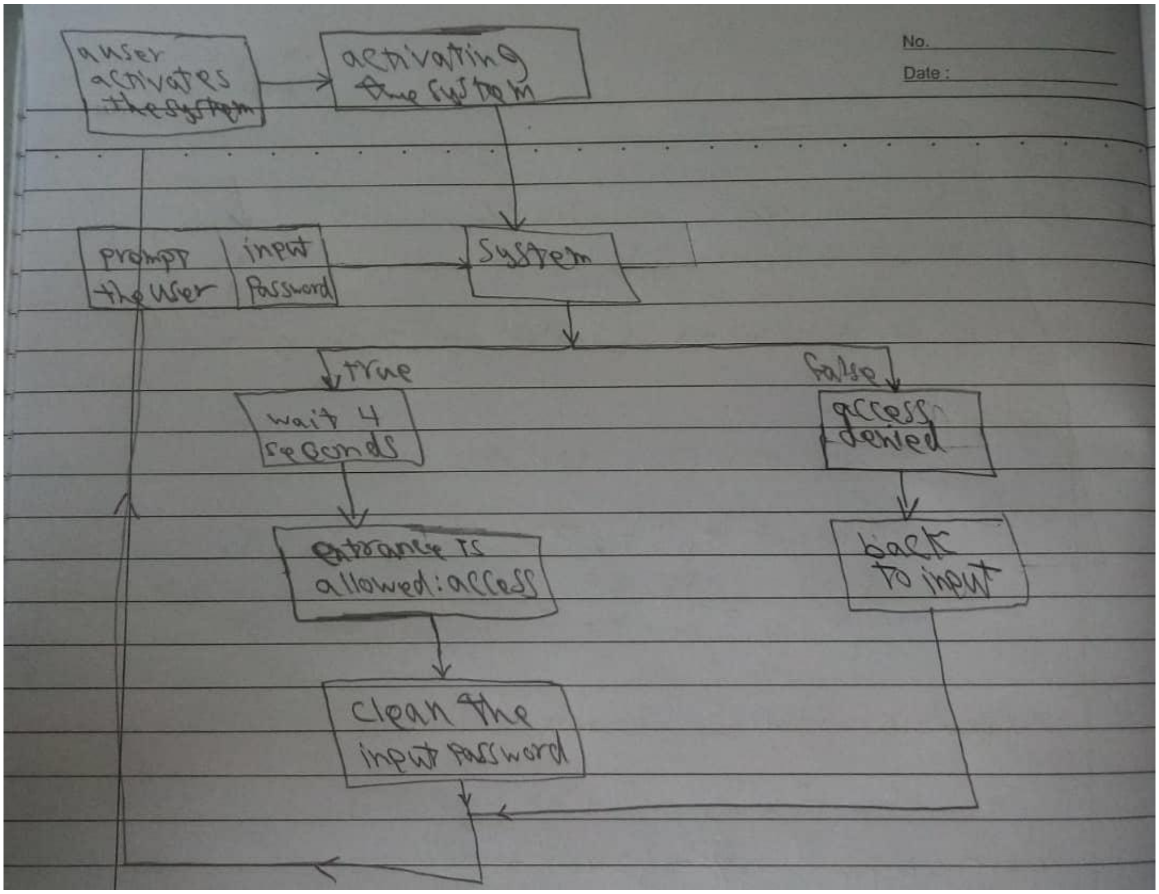


Improved circuit of the system



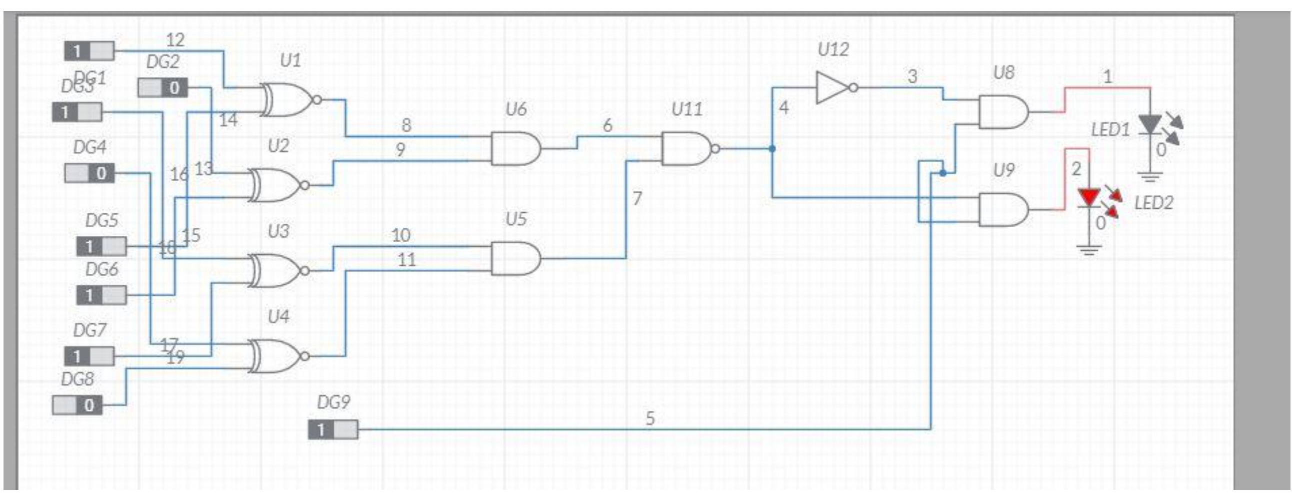
State diagram

The flow of system actions



Design verification

Due to limitations of Multisim, I have verified the simplified circuit only



Multisim link

https://www.multisim.com/content/CtZovH56UyUneuNifMtYxN/dldmulitism/open/

Conclusion

From this circuit, it is possible to understand how current entrance locking security systems are functioning and how to improve them in the future. From my prospective, what could be added to the system to make more advanced are more secure access verification methods such as face recognition, fingerprints or voice recognition. Generally, they all are going to depend on the same building blocks of the current design which are comparison of input to reserved data then decide allow access or not by the system. Additionally, there could be an AI method to notify the owner if the system has been accessed forcefully like breaking through the door or similar scenarios, all of this to strengthen the security of the system.