Birla Institute of Technology and Science

K.K. Birla Goa Campus

Design Assignment

Submitted in partial fulfillment of

EEE F-215 Digital Design

Afraj Shaikh- 2019A3PS0294G

Prakhar Jain - 2019A3PS0370G

Aryan Singh - 2019A3PS0372G

Aditya Agarwal - 2019A3PS0356G

Kaustubh Dwivedi - 2019A3PS0412G

Samarth Agarwal- 2019A3PS0418G

Submitted To:

Prof. Abhijit Pethe

Dr. Ravi Kadlimatti

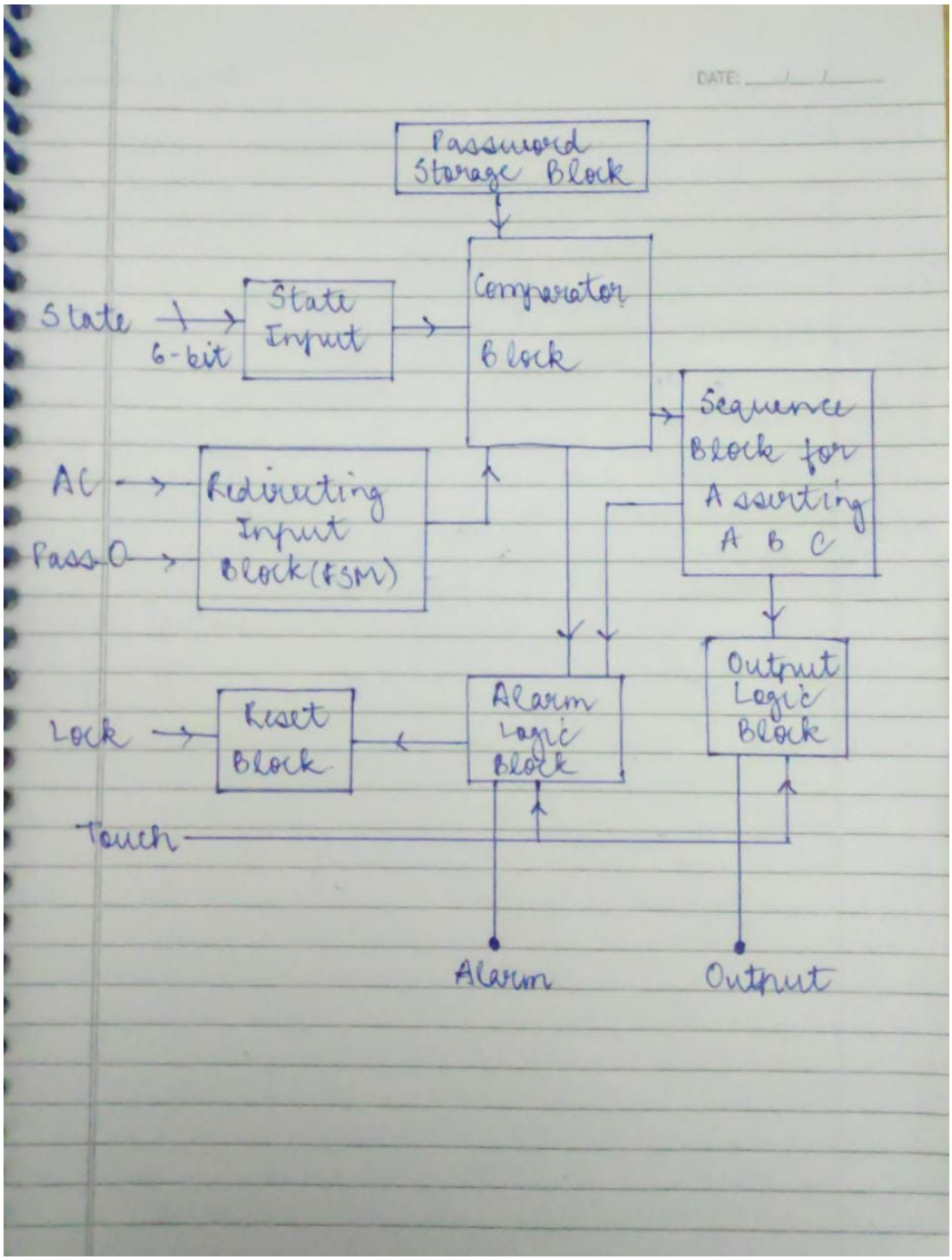
Dr. Sudeep Baudha

And the entire faculty of Digital Design

Acknowledgement

We would like to thank all the faculty members of Digital Designs BITS Pilani, KK Birla Goa Campus without whose guidance and encouragement, this project would have been impossible to complete. We are deeply indebted to all and thankful that we could take part in this project and had an opportunity to test the skills that we had developed throughout the semester.

Top Level Block Diagram



Assumptions

1. Input keys A or B can’t be zero since at zero rotation and determination whether the user has passed 0 is not practical
2. For simulation purposes a bidirectional counter has been used which would be replaced by direct user input from the lock in the end diagram, as shown in the IC implementation.
3. The user is supposed to release the knob when he is done entering the combination into the circuit,only after which the lock would open.
4. The speed at which the user is rotating the knob of the clock is assumed to be much lower than the clock frequency
5. The 6 bit input will change only when the user has been stable at a combination.
6. It is expected that the machine will provide input whether it is moving clockwise or counterclockwise along with a sensor that determines when the person has reached zero while finding the second number.
7. The machine is equipped with a sensor which will sense whether the end user has held the knob or released it.

Inputs Assumed

1. **AC:** A signal from a knob sensor which determines if the user is moving clockwise (0) or counterclockwise (1)
2. **Pass_0:** A signal which asserts when the user has reached zero while finding the second number.
3. **Touch:** A signal which asserts when the user takes off his hand off the knob, stays zero otherwise,
4. **Lock:** Serves two purposes
    a. Locks the circuit once it has been opened after entering the right combination, resets all the state machines and the clock input.
    b. Alternatively can be used to reset the circuit at anytime in the middle of entering the code

Outputs Given

1. **o/p:** The final output which asserts when the user has sequentially given correct input values.
2. **Alarm:** Rings when user has given wrong values.

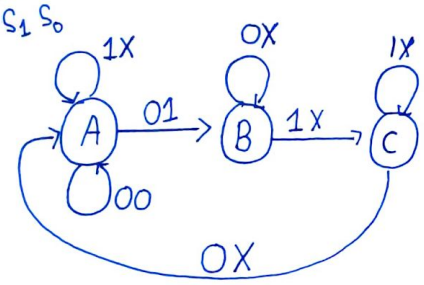
State Diagram of both Finite State Machines

State Digram For Tranffering the user i/p through a De-Mux

i/p → Anticlockwise Signal (AC)
Pass_0 Signal

o/p → nil

States → S_1 S_0



Assuming A = 00 B = 01 C = 10 and rest as don't cares

$S_0(t+1) = (S_1' AC' Pass0 + S_0 AC')t$

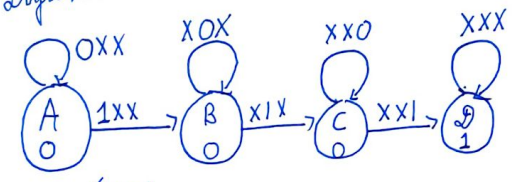
$S_1(t+1) = S_1 S_0(t) + AC$

State Digram to Unlock the circuit

i/p → Assert_A ⇒ Asserts when user inputs first correct key
Assert_B ⇒ Asserts when user inputs second correct key
Assert_C ⇒ Asserts when user inputs third correct key

o/p → y = Asserts if all the 3 keys are correct (sequentially)

State Digram:



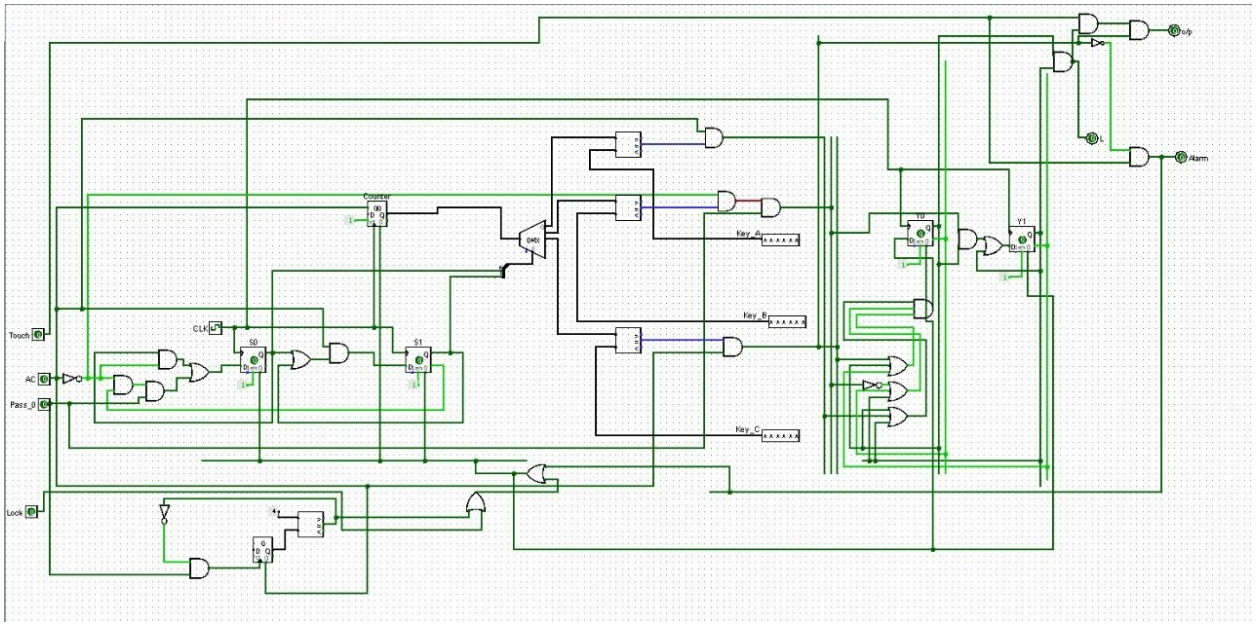
Assuming $(Y_1 Y_0)$

A = 00 B = 01 C = 10 D = 11

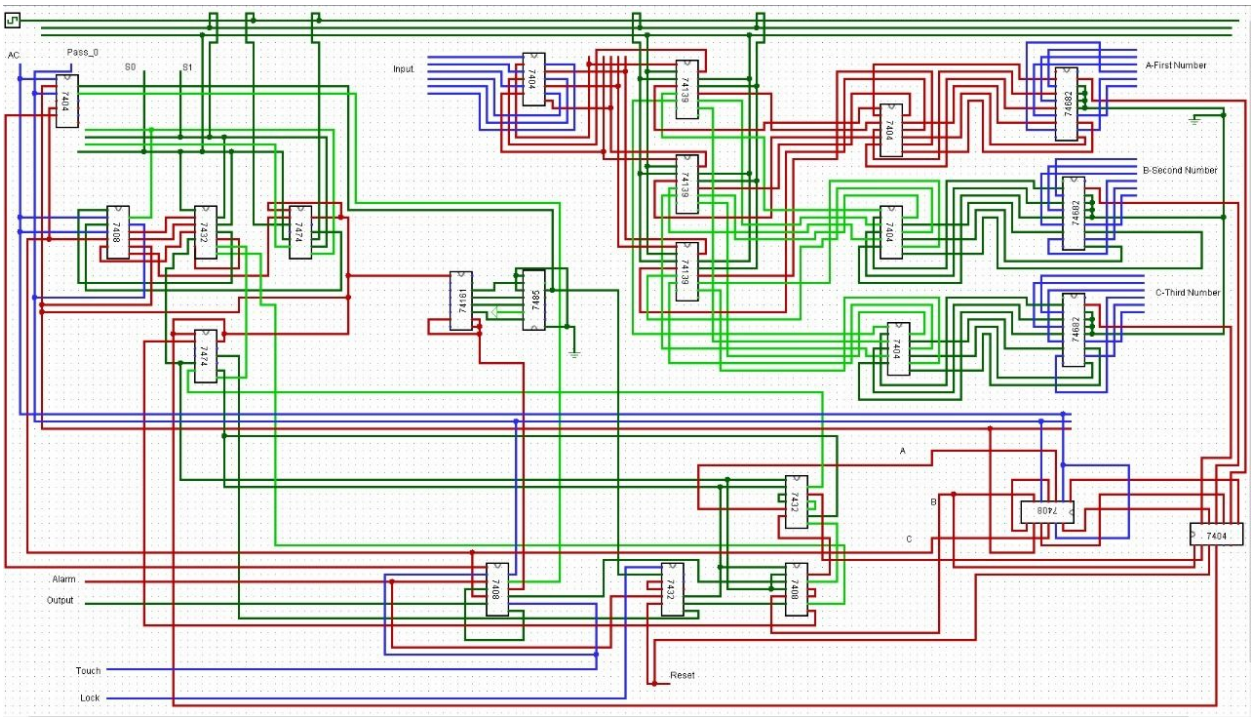
$Y_1(t+1) = Y_0(t)[Assert\_B] + Y_1(t)$

$Y_0(t+1) = (Y_1(t) + Y_0(t) + Assert\_A)(Y_1(t) + Y_0'(t) + Assert\_B)(Y_1(t)' + Y_0(t) + Assert\_C)$

Block Diagram



IC Implementation



Sample Input and Output

Key used in simulation: A-39, B-3, C-38

Simulation when the correct key is entered:

	AC	Pass_0	Touch	Counter	Y0	Y1	o/p	
	0	0	0	0	0	0	0	
We start moving anticlockwise towards the first number	1	0	0	0	0	0	0	
	1	0	0	39	0	0	0	
	1	0	0	39	1	0	0	States change as soon as we reach the first number
We start moving clockwise direction towards second number	0	0	0	39	1	0	0	
	0	0	0	0	1	0	0	
Pass_0 input changes to one as we pass 0	0	1	0	0	1	0	0	
	0	1	0	1	1	0	0	
	0	1	0	2	1	0	0	
	0	1	0	3	1	0	0	
	0	1	0	3	0	1	0	Flip Flops changes states again as we reach the second number
We start moving in anticlockwise direction to the last number	1	1	0	3	0	1	0	
	1	1	0	2	0	1	0	
	1	1	0	1	0	1	0	
	1	1	0	0	0	1	0	
	1	1	0	39	0	1	0	
	1	1	0	38	0	1	0	
	1	1	0	38	1	1	0	Flip Flops changes it's state as we reach the final number
	1	1	1	38	1	1	1	Lock is unlocked as soon as the person releases the lock

Simulation when the user passes the last key while entering the inputs:

AC	Pass_0	Touch	Counter	Y0	Y1	Alarm	o/p
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	39	0	0	0	0
1	0	0	39	1	0	0	0
0	0	0	39	1	0	0	0
0	0	0	0	1	0	0	0
0	1	0	0	1	0	0	0
0	1	0	1	1	0	0	0
0	1	0	2	1	0	0	0
0	1	0	3	1	0	0	0
0	1	0	3	0	1	0	0
1	1	0	3	0	1	0	0
1	1	0	2	0	1	0	0
1	1	0	1	0	1	0	0
1	1	0	0	0	1	0	0
1	1	0	39	0	1	0	0
1	1	0	38	0	1	0	0
1	1	0	38	1	1	0	0
1	1	0	37	1	1	0	0
1	1	1	0	0	0	1	0

Flip Flops changes it's states to the final state as we reach the last key, but the lock does not open as we have not released the key

Since we release the lock after crossing 38, the lock does not open and the alarm goes off indicating that the user has entered a wrong combination

Simulation of different ways to reset the entire circuit

AC	Pass_0	Touch	Lock	Counter	Y0	Y1	o/p
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	39	0	0	0
1	0	0	0	39	1	0	0
0	0	0	0	39	1	0	0
0	1	0	0	39	1	0	0
0	0	0	0	39	1	0	0
0	1	0	0	39	1	0	0
0	0	0	0	39	1	0	0
0	1	0	0	39	1	0	0
0	0	0	0	39	1	0	0
0	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	39	0	0	0
1	0	0	0	39	1	0	0
1	0	0	0	38	1	0	0
1	0	0	0	37	1	0	0
1	0	0	1	0	0	0	0

The entire circuits resets as soon as the sensor moves past 0 at least four times in the clockwise direction

The entire circuits resets when the user manually locks the lock

Additional Functionalities

1. We have provided a Lock option so that the owner can directly set the password for the lock without moving it clockwise a few times.
2. We have also provided an Alarm feature so that whenever someone enters third number wrong, an alarm would ring indicating the owner about the misuse of the lock and the lock would automatically reset.

Bill of Materials

Sr. no	IC Name	Quantity	Description	Datasheet Link from TI
1	7404	6	This device contain six independent inverter	7404
2	7408	4	This device contain four independent 2-input AND Gate	7408
3	7432	3	This device contains four independent 2-input OR Gate	7432
4	7474	2	This device contains two positive edge triggered Dual D Flip Flops with set and reset inputs	7474
5	74139	3	This device contains two Dual 2 to 4 Line Decoder/Demultiplexer	74139
6	74682	3	This device contains a 8-bit magnitude comparator, which compares two 8 bit numbers	74682
7	7485	1	This device contains a 4-bit magnitude comparator, which compares two 4 bit numbers	7485
8	74161	1	This device contains a High-speed CMOS logic presettable 4 bit synchronous counter	74161

Appendix

https://github.com/r0the/logi7400

https://en.m.wikipedia.org/wiki/List_of_7400-series_integrated_circuits

http://www.cburch.com/logisim/links.html