MCTE2332

ELEVATOR

PROJECT

Name: Mohamed Nur Jamila Ali

Matric: 1627540

LECTURER: DR. HAZLINA BT. MD. YUSOF

DATE OF SUBMISSION: 25 /01/ 2021

SEM 1 20/21

Table of content

1. INTRODUCTION

1.1 Overview

An elevator, also called a lift, is a vehicle that moves in a vertical column to transport passengers or freight between levels of a multi-storey building. Most modern elevators are driven by electric motors, with the help of a counterweight, through a system of cables and beams (pulleys). By opening the way to tall buildings, the elevator has played a critical role in creating the distinct urban geography of many modern cities, especially in the United States, and promises to fill an indispensable role in the city's future development.

1.2 Objectives

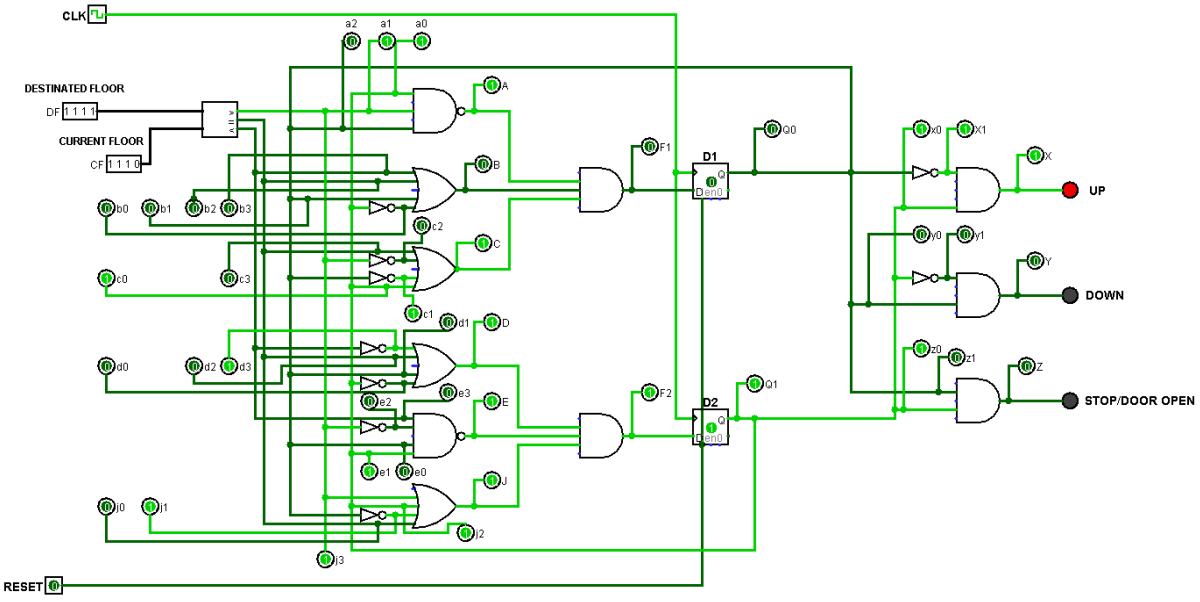
- Design a system of controlling elevator using combinational and sequential logic circuits.
- Apply all that has been learned in the course.

2. Design Process

The design askes the user to enter the desirable floor and current floor by changing the 4-bit binary inputs then, the inputs will be fed to a comparator to compare between the number of desirable floor and current floor to determine if the elevator will go up, down or stops. The outputs of the comparator will be fed to a set of combinational circuits which contains of NOT, AND, OR, and NAND gates then, the output of these gates will be the input to the D-flip flops. The D-flip flops are triggered by positive edge clock, and their outputs are connected to AND gate and NOT gate that will trigger the LEDs to show of the movement of the elevator.

3. Details

3.1 Circuit



3.2 4-BIT Comparator

Expression

A=B

A=B is true only when (A3=B3 and A2=B2 and A1=B1 and A0=B0). Hence

Z (A=B) = A3B3 . A2B2 . A1.B1 . A0.B0 = x3x2x1x0

A>B

X(A>B) = A3B3′ + x3A2B2′ + x3x2A1B1′ + x3x2x1A0B0′

A<B

Y(A<B) = A3′B3 + X3A2′B2 + X3X2A1′B1 + X3X2X1A0′B0

Since the truth table contain 256 rows which is quite long but I will provide the truth table for 2-bit because the same principles are used to derive a 4-bit

Truth table 2-bit comparator

A1	A0	B1	B0	A>B	A<B	A=B
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	1
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	0	1

3.3

Truth Table

A

a0	a1	a2	a3	A
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

B

b0	b1	b2	b3	B
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

C

c0	c1	c2	c3	C
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

E

e0	e1	e2	e3	E
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

D

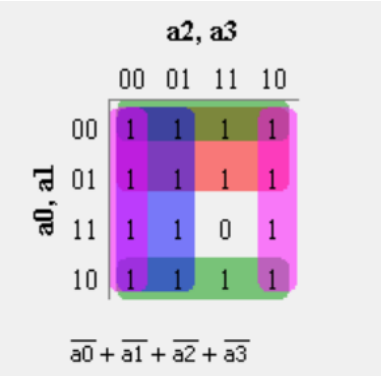
d0	d1	d2	d3	D
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

J

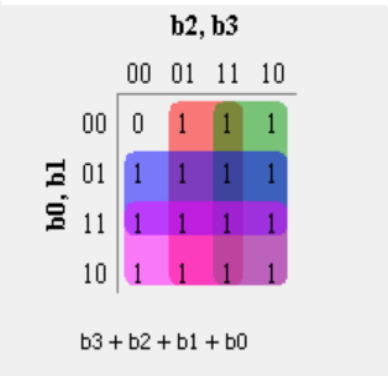
j0	j1	j2	j3	J
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

K-map

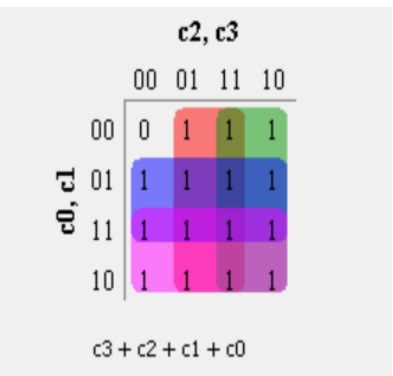
A



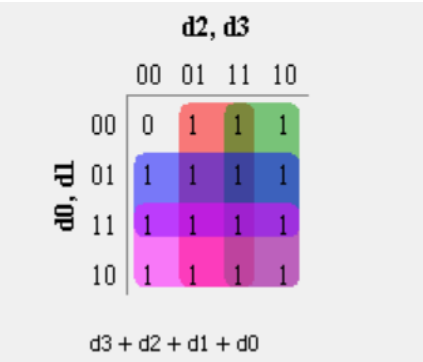
B



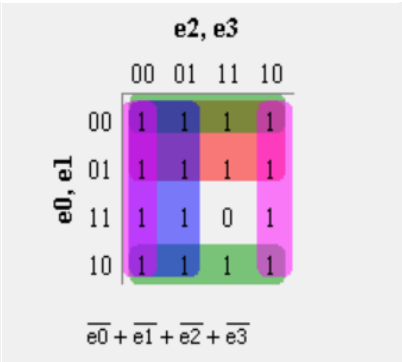
C



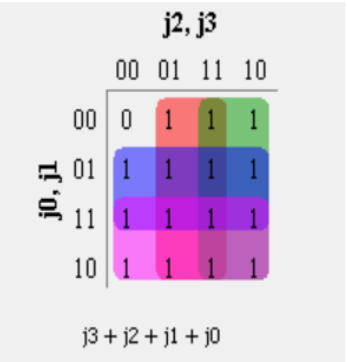
D



E



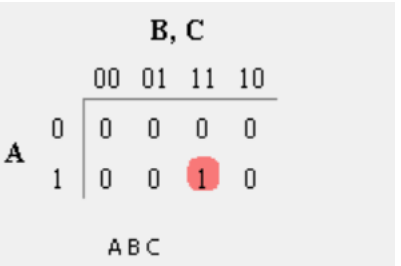
J



F1= ABC

F2= DEJ

A	B	C	F1
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



D	E	J	F2
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

D \ E, J	00	01	11	10
0	0	0	0	0
1	0	0	1	0

DEJ

3.4 D-FLIP FLOP

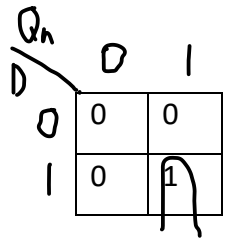
Truth table

Clk	D	Q(next)	Action
0	X	Q	No change
1	X	Q	No change
↑	0	0	Reset
↑	1	1	Set

1st and 2nd D flip flops have the same truth table

Clk	D_1=F1 or D_2=F2	Q_{Next}	Action
↑	0	0	Reset
↑	0	0	Reset
↑	0	0	Reset
↑	0	0	Reset
↑	0	0	Reset
↑	0	0	Reset
↑	0	0	Reset
↑	1	1	Set

K-map for the D Flip flops

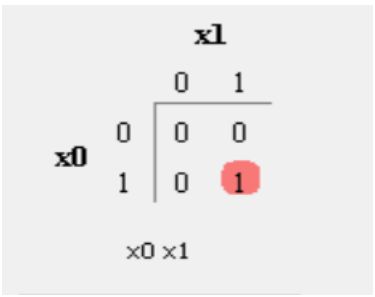


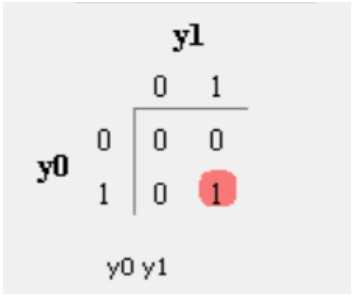
X **Y** **Z**

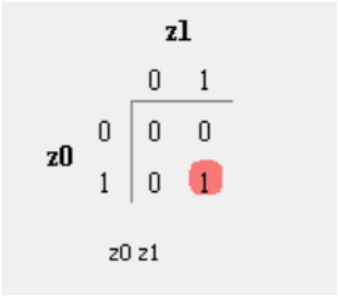
x0	xl	X
0	0	0
0	1	0
1	0	0
1	1	1

y0	yl	Y
0	0	0
0	1	0
1	0	0
1	1	1

z0	zl	Z
0	0	0
0	1	0
1	0	0
1	1	1







ELEVATOR GOING UP

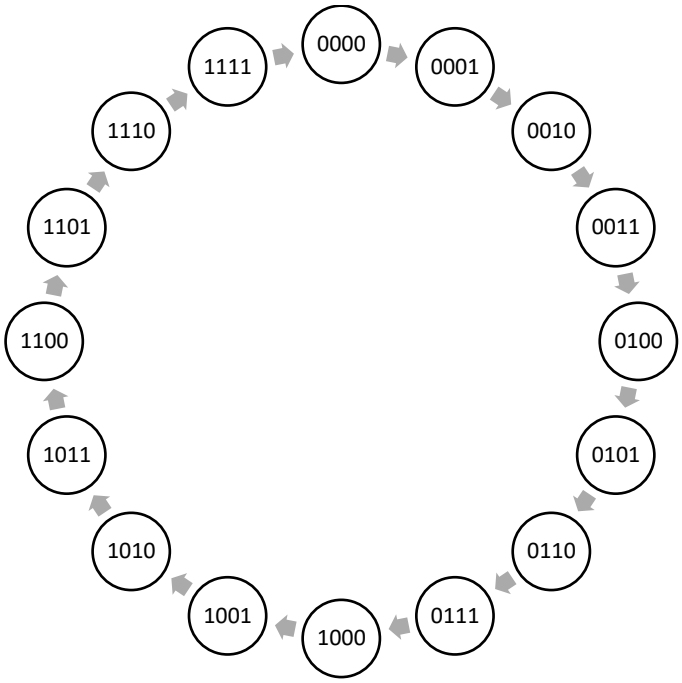
D/F	C/F	D1	D2	UP/LED1	DOWN/LED2	STOP/LED3
0000	0000	1	1	0	0	1
0001	0000	0	1	1	0	0
0001	0001	1	1	0	0	1
0010	0001	0	1	1	0	0
0010	0010	1	1	0	0	1
0011	0010	0	1	1	0	0
0011	0011	1	1	0	0	1
0100	0011	0	1	1	0	0
0100	0100	1	1	0	0	1
0101	0100	0	1	1	0	0
0101	0101	1	1	0	0	1
0110	0101	0	1	1	0	0
0110	0110	1	1	0	0	1
0111	0110	0	1	1	0	0
0111	0111	1	1	0	0	1
1000	0111	0	1	1	0	0
1000	1000	1	1	0	0	1
1001	1000	0	1	1	0	0
1001	1001	1	1	0	0	1
1010	1001	0	1	1	0	0
1010	1010	1	1	0	0	1
1011	1010	0	1	1	0	0

1011	1011	1	1	0	0	1
1100	1011	0	1	1	0	0
1100	1100	1	1	0	0	1
1101	1100	0	1	1	0	0
1101	1101	1	1	0	0	1
1110	1101	0	1	1	0	0
1110	1110	1	1	0	0	1
1111	1110	0	1	1	0	0
1111	1111	1	1	0	0	1

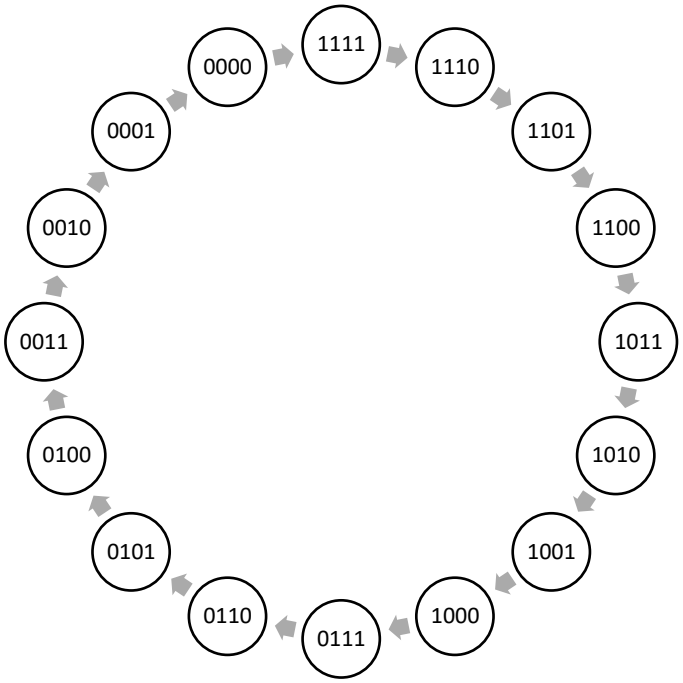
ELVATOR MOVING DOWN

D/F	C/F	D1	D2	UP/LED1	DOWN/LED2	STOP/LED3
1111	1111	1	1	0	0	1
1110	1111	1	0	0	1	0
1110	1110	1	1	0	0	1
1101	1110	1	0	0	1	0
1101	1101	1	1	0	0	1
1100	1101	1	0	0	1	0
1100	1100	1	1	0	0	1
1011	1100	1	0	0	1	0
1011	1011	1	1	0	0	1
1010	1011	1	0	0	1	0
1010	1010	1	1	0	0	1
1001	1010	1	0	0	1	0
1001	1001	1	1	0	0	1
1000	1001	1	0	0	1	0
1000	1000	1	1	0	0	1
0111	1000	1	0	0	1	0
0111	0111	1	1	0	0	1
0110	0111	1	0	0	1	0
0110	0110	1	1	0	0	1
0101	0110	1	0	0	1	0
0101	0101	1	1	0	0	1
0100	0101	1	0	0	1	0
0100	0100	1	1	0	0	1
0011	0100	1	0	0	1	0
0011	0011	1	1	0	0	1
0010	0011	1	0	0	1	0
0010	0010	1	1	0	0	1
0001	0010	1	0	0	1	0
0001	0001	1	1	0	0	1
0000	0001	1	0	0	1	0
0000	0000	1	1	0	0	1

3.5 State Diagram UP- count

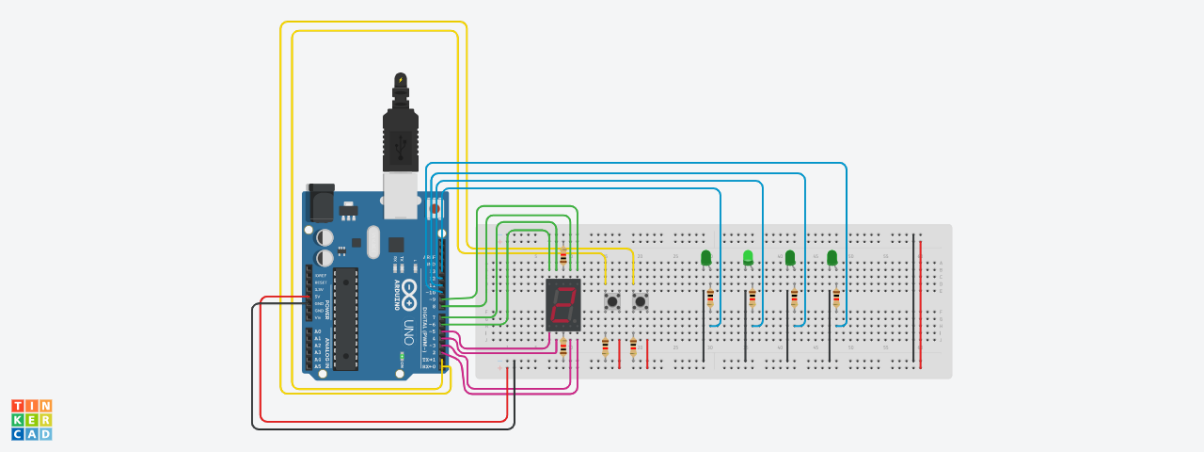


Down- Count



4 Design verification

4.1 TinkerCad



Link: https://www.tinkercad.com/things/k5HYfTL2C3i-copy-of-the-elevator-v3/editel?tenant=circuits

5 Limitation

In the design of the tinkercad circuit, there was some limitation that prevented to design of the exact circuit in Logisim. First, the tinkercad is being under maintenance these past days which resulted in lagging, some components not functioning as desired. Second, the D-flip flop in tinkercad (74HC74) required a maximum of 7voltage and 0.25 mA current flow which is a small amount to turn a LED on with all of the resistors and gates that are connected to the D-flip flop. Therefore, a similar design was constructed that works in the same principle but slightly different to overcome these limitations.

6 Conclusion

To sum up, the objectives which were to design a system of moving an elevator using combinational and sequential logic circuits was achieved in addition, implementing the knowledge that was learned in the course.