CDA 3201L R 3:30-5:20pm S005
Lab number 8 – Final Project
11-10-2016, Ashley Suh

Purpose and Objectives

The purpose of Lab 8 is to design a finite state machine which controls the operation of an elevator in a 2-story building. The finite machine has one input (the enable control) and three outputs: the door (0 if closed, 1 if on), the floor number(0 for 1st floor, 1 for 2nd floor), and whether or not the elevator is moving (0 for not moving, 1 for moving). The elevator has a specific sequence with four states, which is discussed in greater detail during the design description.

Components Used

Name	**Type**	**Quantity**
Power Supply	3.5V	1
Wire Kit		1
LED	Red LED	2
DM 74LS32	Quad OR Gate	2
DM 74LS04	Quad Inverter Gate	1
DM 74LS08	Quad 2-input AND Gate	1
DM 74LS11	3-input AND Gate	2
7-segment LED	0-15 LED output display	1
DM 74LS247	BCD-to-7 segment decoder	2
DM 74LS163	Binary Counter	1
DM 74LS86	Quad XOR Gate	1
DM 74LS74	D Flip Flop	3

CDA 3201L R 3:30-5:20pm S005
Lab number 8 – Final Project
11-10-2016, Ashley Suh

Design Description

The elevator performs the following sequence:

1) Opens the door for 4 seconds
2) Closes the door
3) Moves the elevator to the destination floor in 4 seconds and display the current floor
4) Open the door for 4 seconds
5) Close the door
6) Remain in the same floor until the enable input is set again

The sequence is divided into four states:
S0 : Idle - the door is closed and not moving
S1 : Door is open for 4 seconds and not moving
S2 : Door is closed and moving floors
S3 : Door is open for four seconds and at the new floor, not moving

S3 then returns to the initial state, **S0**, and remains in **S0** until enable is turned back on. The counter is used to calculate the duration of the stay in the current state and is also used as a trigger to change states. This variable is denoted "TC", which outputs a 1 when the counter has counted through 0-15. TC is one of the four variables in the truth table, along with State (A, B), and Enable (X). There are also two outputs for the table: **Z1** (door on or off) and **Z2**(elevator moving or not).

From looking at the state table (following page), we can see the door is dependent only on **B** and A is don't care. Moving is also only outputting during the state **S2**, which is denoted as "10". This translates to **A*B'**. Then we have

Door (Z1) = B, and
Moving (Z2) = AB' .

Once the elevator has been enabled (X = 1), during the initial state, the machine will continue to follow through the entire sequence of outputs regardless if X is turned off or not. However, if X is set to 0 during the sequence, the elevator will revert and remain in the initial state once it has completed **S3**. The variable **TC** is treated as a trigger, that is, when **TC = 0**, the state will remain the same. When **TC = 1**, the state will also trigger and move to the following state. After completing the state table, the next state K-maps are created for **A+** and **B+** (following page). We are left with

A+ = A'BT + AT' + AB'
B+ = A'B'X + BT' + AB'T

The last output to account for is the floor level. The floor only changes with the elevator moving, and it only depends when the counter has triggered. So floor is implemented with a T flip flop with the input as the moving output, and the clock is the output of TC. The logic table is shown on the following table.

CDA 3201L R 3:30-5:20pm S005
Lab number 8 – Final Project
11-10-2016, Ashley Suh

1

Present State	enable X	carry T	A^+B^+ Next State	door Z_1	moving Z_2
S_0 (00)	0	0	S_0 (00)	0	0
	0	1	S_0 (00)	0	0
	1	0	S_1 (01)	1	0
	1	1	S_1 (01)	1	0
S_1 (01)	0	0	S_1 (01)	1	0
	0	1	S_2 (10)	0	1
	1	0	S_1 (01)	1	0
	1	1	S_2 (10)	0	1
S_2 (10)	0	0	S_2 (10)	0	1
	0	1	S_3 (11)	1	0
	1	0	S_2 (10)	0	1
	1	1	S_3 (11)	1	0
S_3 (11)	0	0	S_3 (11)	1	0
	0	1	S_0 (00)	0	0
	1	0	S_3 (11)	1	0
	1	1	S_0 (00)	0	0

State	Function
S_0	Idle-door is closed & not moving
S_1	Door is open for 4 seconds & not moving
S_2	Door is closed & moving floors
S_3	Door is open for 4 seconds & at the new floor

* Z_1 depends only on B^+
* Z_2 depends on $A^+\overline{B}^+$

CDA 3201L R 3:30-5:20pm S005
Lab number 8 – Final Project
11-10-2016, Ashley Suh

K-map

A^+

XT \ AB	00	01	11	10
00	0	0	1	1
01	0	1	0	1
11	0	1	0	1
10	0	0	1	1

$$\boxed{A^+ = \bar{A}BT + A\bar{T} + A\bar{B}}$$

B^+

XT \ AB	00	01	11	10
00	0	1	1	0
01	0	0	0	1
11	1	0	0	1
10	1	1	1	0

$$\boxed{B^+ = \bar{A}\bar{B}X + B\bar{T} + A\bar{B}T}$$

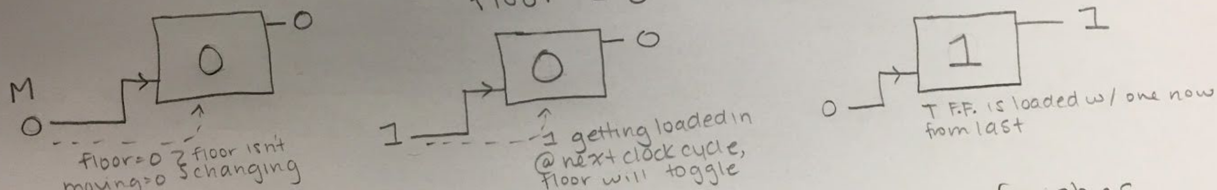
CDA 3201L R 3:30-5:20pm S005
Lab number 8 – Final Project
11-10-2016, Ashley Suh

(4)

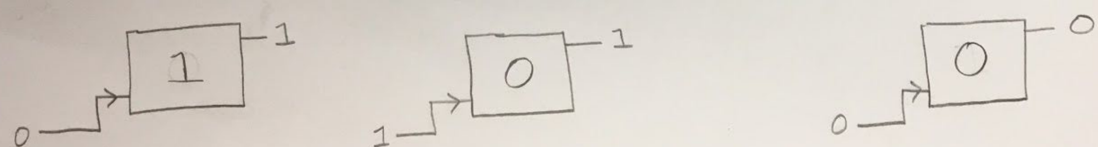
Q	T	Q(t+1)
0	0	0
0	1	1
1	0	1
1	1	0

Floor : T flip flop

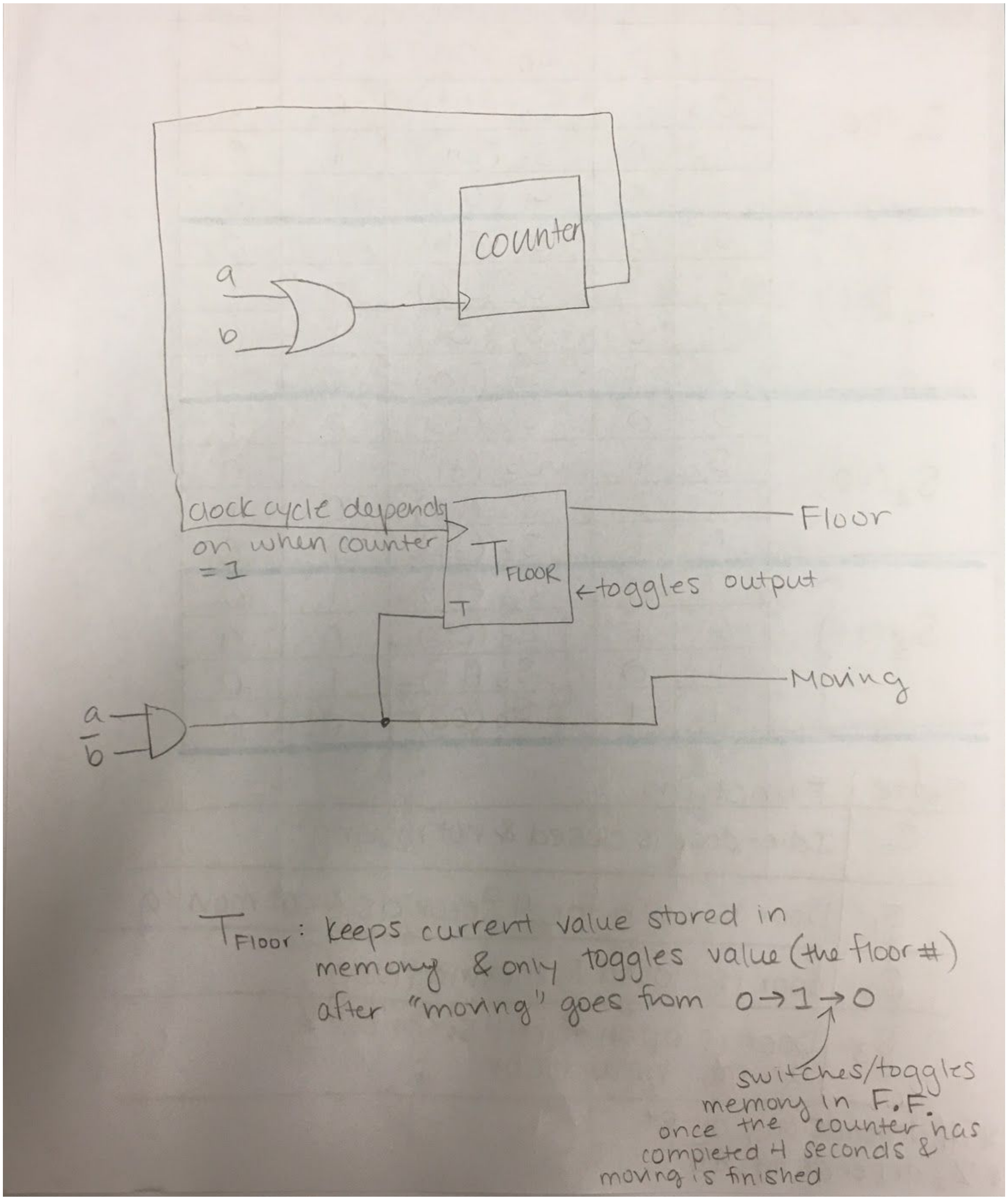
0 → Starts @ floor 0 → Elevator begins moving, moving = 1, floor = 0 → Elevator finishes moving, moving = 0 and floor = 1



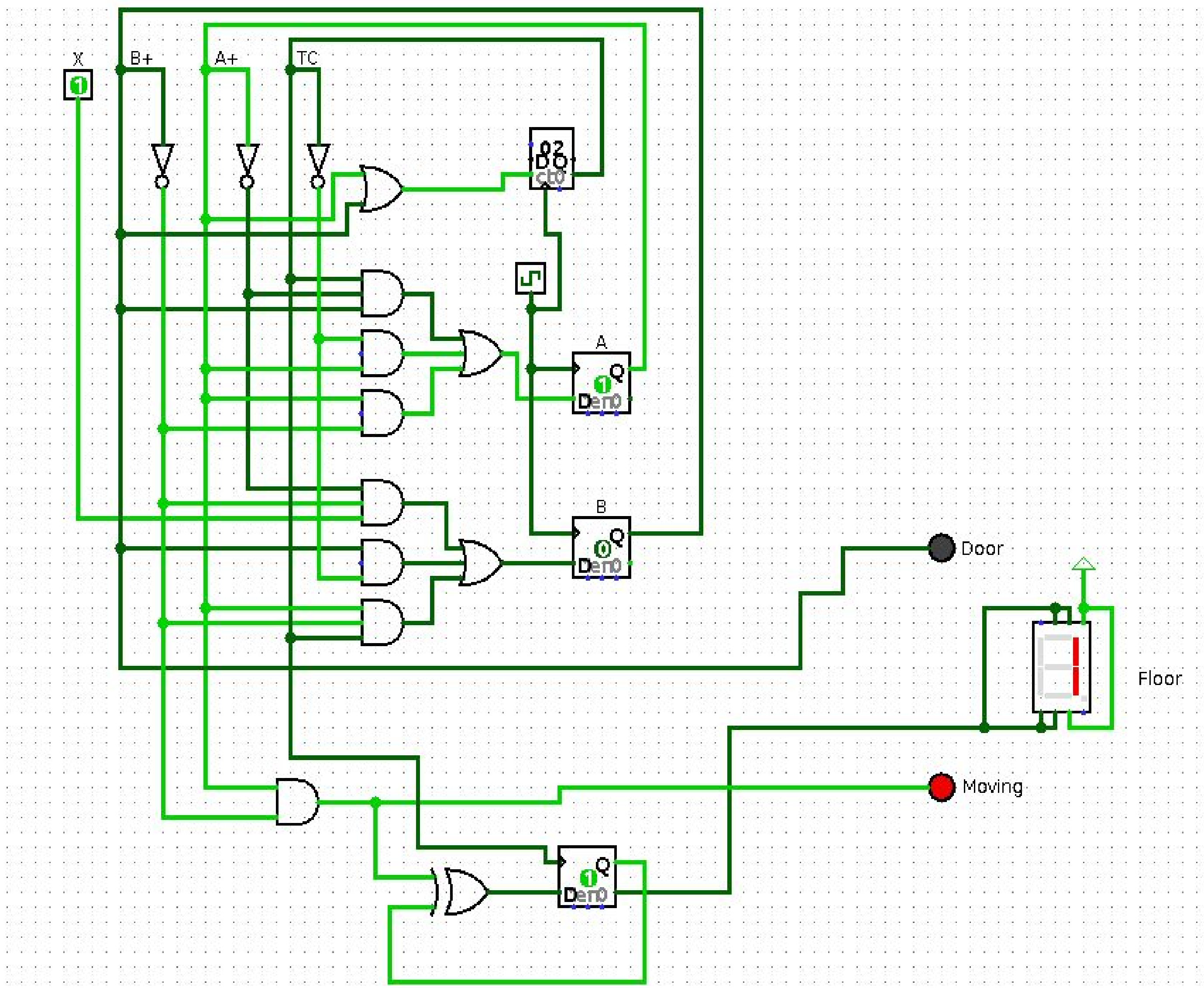
1 → starts @ floor 1 → Elevator begins moving, moving = 1, floor = 1 → Elevator finishes moving, moving = 0 and floor = 0



CDA 3201L R 3:30-5:20pm S005
Lab number 8 – Final Project
11-10-2016, Ashley Suh



CDA 3201L R 3:30-5:20pm S005
Lab number 8 – Final Project
11-10-2016, Ashley Suh



Discussion and Conclusions

Results were correct and all tests were successful in the end. The most difficult implementation was tracking the floor number for where the elevator travelled to. After testing multiple attempts through logism, it was finally concluded to store the memory of the floor the elevator is on, we must use a T flip flop with the input of the moving bit and the TC as the clock cycle. It is also observed that the sequence will continue while enable is on, but when enable is off, the sequence will end at **S0** without restarting.