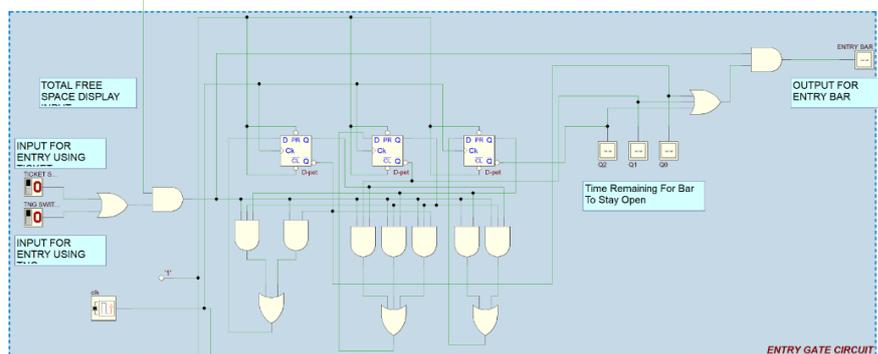
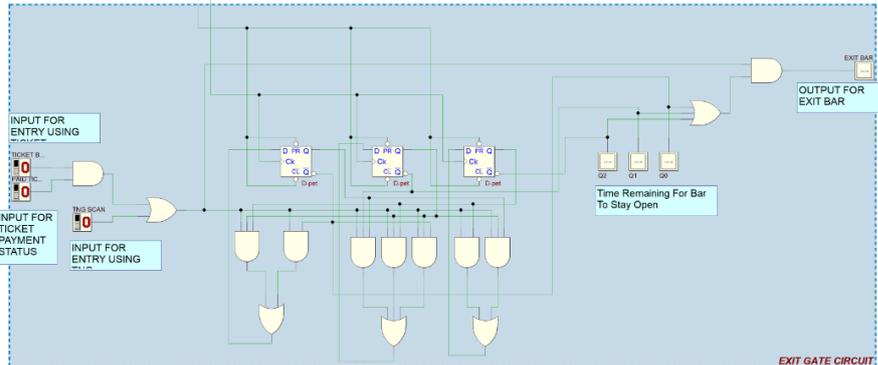




# OUR COMPLETE CIRCUIT



- INPUTS:**
- ENTRY GATE
  - TICKET INPUT
  - TNG INPUT
  - FREE SPACE COUNTER OUTPUT
- EXIT GATE**
- TICKET INPUT
  - TICKET PAYMENT STATUS INPUT
  - TNG INPUT
- PARKING FLOOR(EACH FLOOR)**
- 8 SENSOR INPUTS
- OUTPUT:**
- ENTRY GATE**
- ENTRY BAR ACTIVE-TIMER
  - ENTRY BAR OUTPUT (will only be active for 7 seconds after button is pressed)
  - TOTAL FREE SPACE AVAILABLE OUTPUT
- EXIT GATE**
- EXIT BAR ACTIVE-TIMER
  - EXIT BAR OUTPUT (will only be active for 7 seconds after button is pressed)



**BAR COUNTER PROPERTY:**  
 COUNT DOWN FROM 111 TO 000  
 WHEN INPUT = 1, RESET BACK TO 111  
 WHEN (INPUT = 1 AND COUNTER != 0), BAR



**UTM**  
**UNIVERSITI TEKNOLOGI MALAYSIA**

**SECR 1013 - DIGITAL LOGIC**

**PROJECT - PART A**

**MEMBERS:**

Khairul Fahmi Bin Mohd Yusof	- A21MJ5011
Ching Jie Kai	- A21MJ5037
Soo Wei Xiang	- A21MJ5038
Adam Iskandar Bin Sharudin	- A21MJ5041
Mervinraj	- A21MJ5015

## 1. Background (PART - A)

In this project, my team has designed a smart car park system. Our design can be separate into two part which is “Floor ” and Entry/Exit,

### **Floor Counter (Part A)**

For the floor part, we will have 4 floors in this car park, each floor will have a LED screen that shows the number of parking slots that are still available at the current floor and the total available space of the whole car park. For each parking slot, there will be one sensor with LED at the top of the parking slot. When the car is parked in the slot, the output of the sensor will become 1, the LED in the sensor will turn red (**LED = 0**), then the total space number for the whole car park (including the LED screen at the entry) and the available space number of that floor will -1. Else if, when car is leaving the parking slot, the output of sensor will become 0, the LED in the sensor will turn green (**LED = 1**), then the total space number for whole car park (**including the LED screen at the entry**) and the available space number of that floor will +1.

### **Entry Gate (Part B)**

The next part is Entry and Exit, at the entrance there will have a barrier gate system with LED to block vehicle from going inside and a LED screen that connected to the counter and sensor comes from each parking space and it will show the total empty spaces (**Total space - number of parking slot with car**). If the screen show that there still has empty space in the car park (**empty space > 0**), driver need to press the button (**input**) to get a parking ticket (**output**) or put their TouchNGo card on the sensor (**input**) and the LED on the barrier will turn green from red (**LED = 1**). Drivers need to wait for 7 second until the barrier open, after that driver is allowed to go into the car park. But if the total number of empty spaces shown by the screen is 0 (empty space = 0) then the driver will not be able to scan the TouchNGo card or get the parking ticket.

### **Exit Gate (Part B)**

Next is for the Exit barrier, for those who use the TouchNGo card to enter the car park, they need to scan their card again to pay (input), after that the LED on the barrier will turn green (**LED = 1**) and the barrier will open. (**barrier = 1**). For those who use parking ticket to enter,

they must pay manually in the pay machine first, else when they scan the ticket using scanner at the barrier, the output from the scanner to the barrier will be 0, and they are not allowed to exit from the car park. Conversely, for those who already paid, the scanner will output a 1 to the barrier and its LED, after 7 second the barrier open and the LED turn green, the vehicle is allowed to leave.

## **2. Problem Statement – Floor Counter**

### **(Part A)**

Most of the car parks out there don't really indicate to you if the parking is full, even if there are spaces available, some people will enter the car park and spend so much time finding parking when there is none available. Even if there is parking available, the fact that someone has to drive to all the floors to find it, is just a tough process and time consuming.

### **(Part B)**

Some car parks only have a ticket-cash system available, we are moving on to a digital era where some people do not really carry cash everywhere. Besides that, people who are taking the ticket have to queue in line to pay for their parking ticket. This process is clearly not efficient so we came up with a few solutions ourselves to make it easy for everyone.

## **3. Solution – Floor Counter**

### **(Part A)**

We have an LED board indicating the number of available spaces in each row and floor, so that the customers will know exactly where the parking is empty. We also have an LED indicator for each parking bay which will turn green when there is no car and red when there is a car, this will not make customers want to pray and hope to get a parking space. Everyone can feel lucky to get a parking space.

### **(Part B)**

First of all, our parking system allows touch and go, and also cash. This will let customers decide on which way they are going to pay depending on what is the easiest for them. Before entering the parking lot, there is an indicator which will show how many more parking spaces are available and at which floor it is available to make it easier for customers. For the exit once the system successfully deducts the parking fee, the exit barrier will be opened for 7 seconds, which is an ample amount of time for a car to exit.



## 4. Requirements

### 4.1 Requirements for Floor Counter (Part A)

- Input switch
  - Represents a parking space. Input is 1 if a car is parked; vice versa.
- NOT gate
  - To invert the inputs.
- Wire
  - To connect all the components in the circuit, so that it can operate.
- LED
  - As an indicator. It will light up if input is 1. (Turns red if parking space is occupied; vice versa)
- Full adder
  - To sum the inputs. Can add the carry obtained from previous addition.
- One Digit, Seven segment display
  - To display the number of parking spaces available.

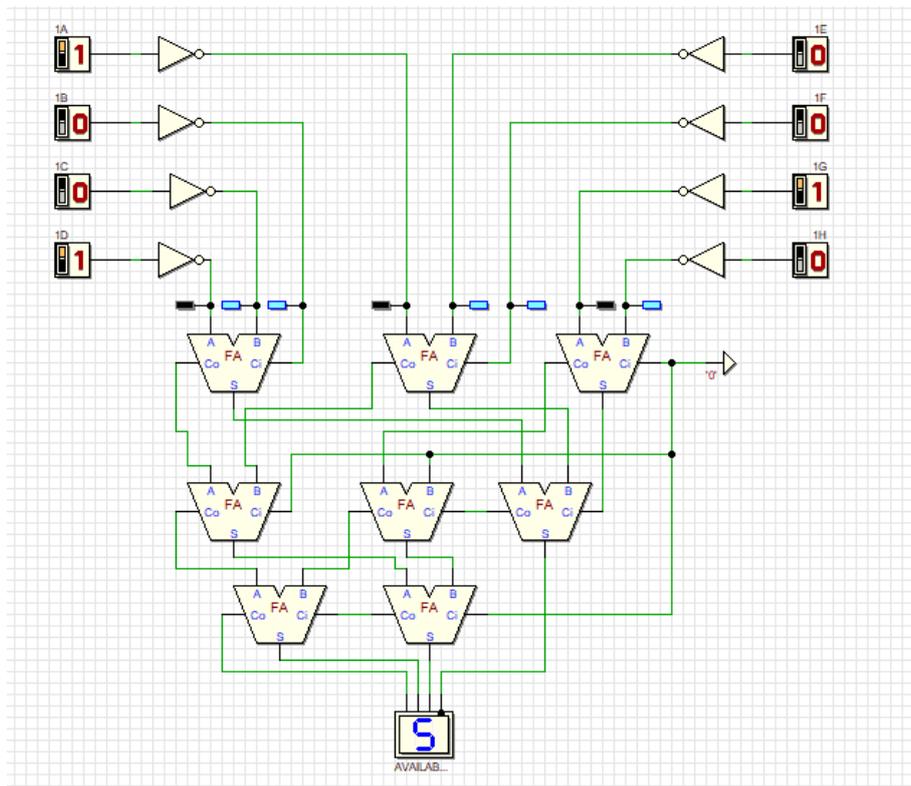
### 4.2 Requirements for Floor Counter (Part B)

- Input switch
  - Represents choice of payment. (Touch n Go or Ticket)
- Wire
  - To connect all the components in the circuit, so that it can operate.
- Full adder
  - To sum the inputs. Can add the carry obtained from previous addition.

- One Digit, Seven segment display
  - To display the number of parking spaces available.
  
- Flip - flop
  - To store binary data. The stored data can be changed by applying varying inputs.
  
- One Bit Output
  - To indicate the output of the circuit.

## 5. System Implementation

### 5.1 Floor Counter (Part A)



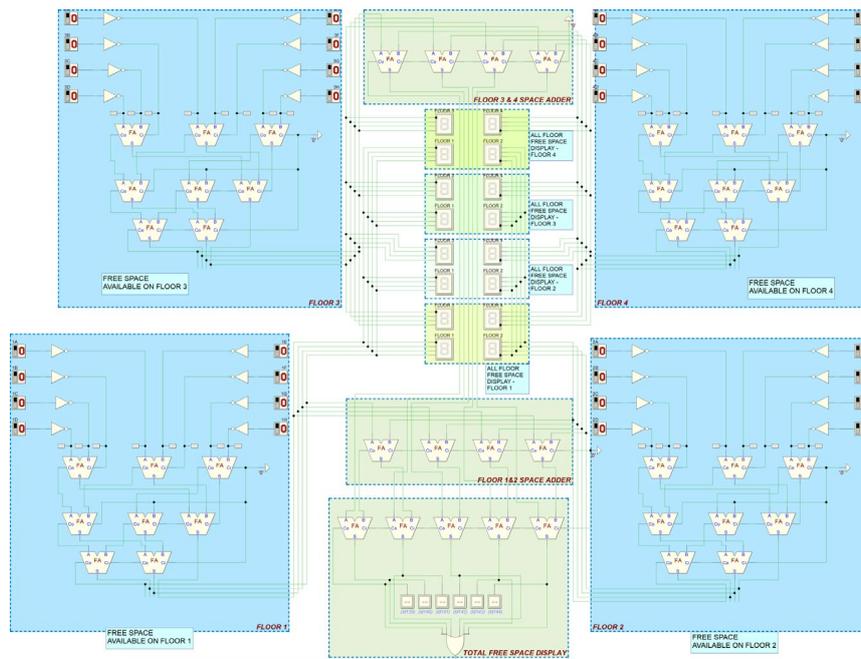
This circuit is designed to represent a floor's parking area. Since each floor will have 8 parking spaces, we have 8 inputs for the circuit. Each input represents 1 parking space. For each input, if there is no car parked in the parking space, the input will be 0. If there is a parked car, the input will be 1.

All of these inputs will be inverted, since we want the circuit to represent the number of empty/ available parking spaces, not the number of occupied parking spaces.

All of the inputs will be added using Full Adders. A total of 8 Full Adders are required to add all 8 of the inputs, and the resulting data will be 4-bits (1000) at most.

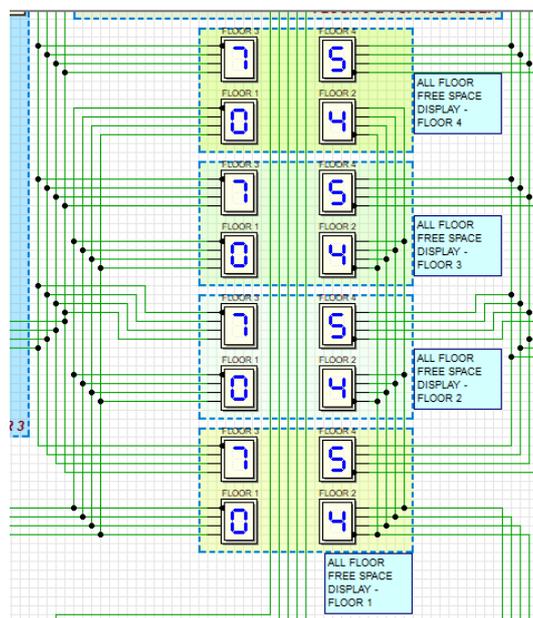
The output is shown using a 7-segment display for each floor. This can be done since the maximum combinational input will be 1000 in binary, which is 8 in decimal.

## 5.2 Total Floor Counter (Part B)



We plan to have 4 floors in total, with each floor having 8 parking spaces. Each floor will consist of a Floor Counter circuit as shown in 5.1. Then, each floor will show the number of spaces available to all the floors. This means a floor will produce 4 identical outputs of the free space available in that floor, and the outputs will be shown on each of the 4 floors. Since we have 4 floors in total, we will have 16 outputs for the number of free space available, with a floor having 4 each, one from each floor.

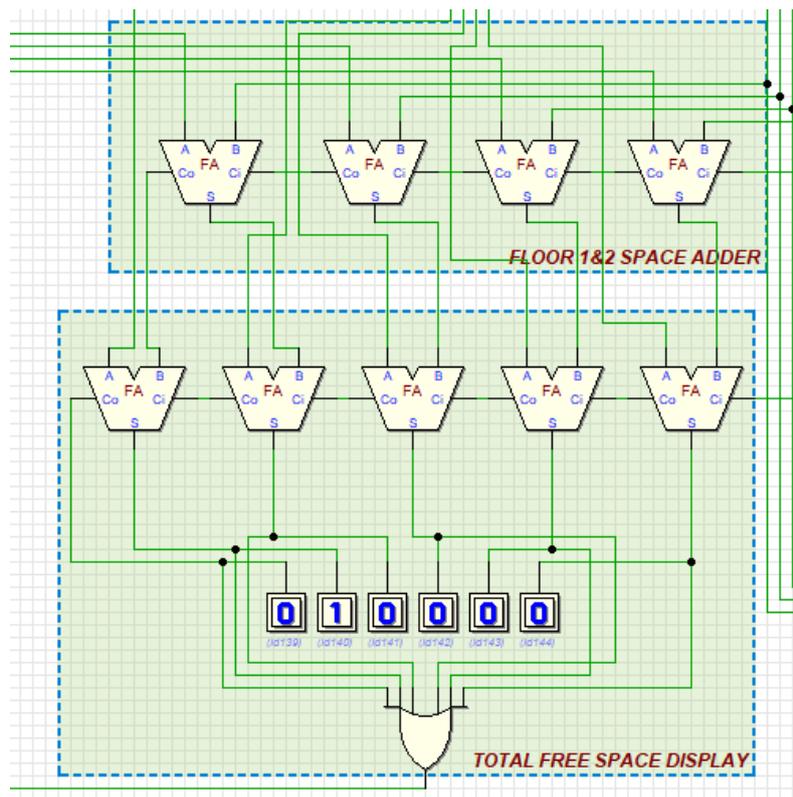
Below is the example of the number of free space output displays, with each floor having 1 output from each of the 4 floors. Since there's 4 floors in total, we will have 16 outputs.



Then, the free space from each floor will be added using Full Adders. The total number of free spaces will be displayed at the Entry Gate, since the entry gate will not open if there is 0 empty space remaining out of all 4 floors. The maximum number of empty space will be:

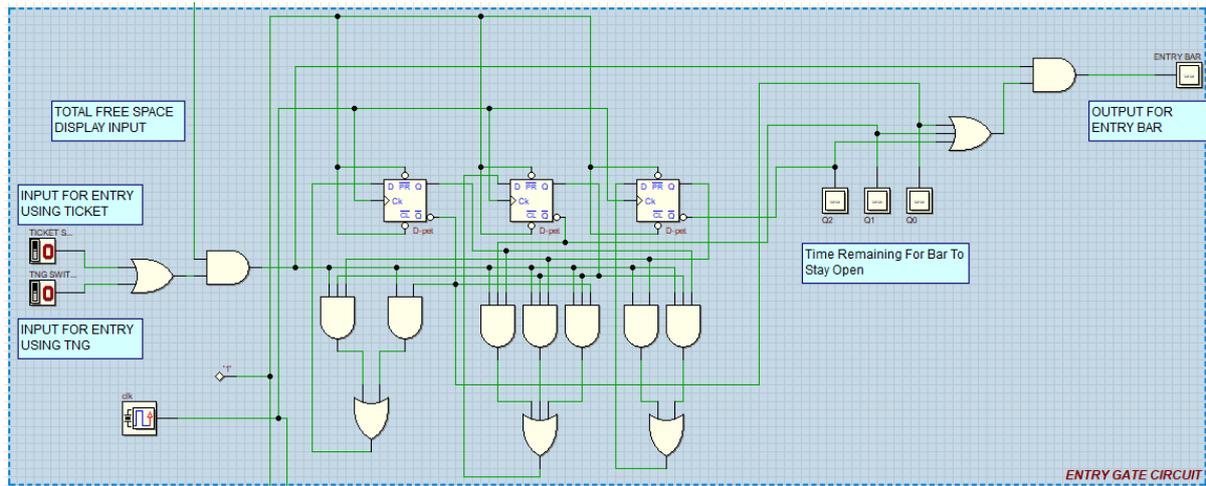
$$-(\text{Parking Space on Each Floor}) * (\text{Number of Floors}) = 8*4 = 32$$

This means the highest number that will be shown on the counter is 100000(6 bit), which is 32 in binary. This output will be sent to the Entry Gate, in order to allow the Entry Gate to check whether there is an empty space available in the parking area or not.



## 5.3 Entry & Exit Gate (Part B)

### 5.3.1 Entry Gate Circuit



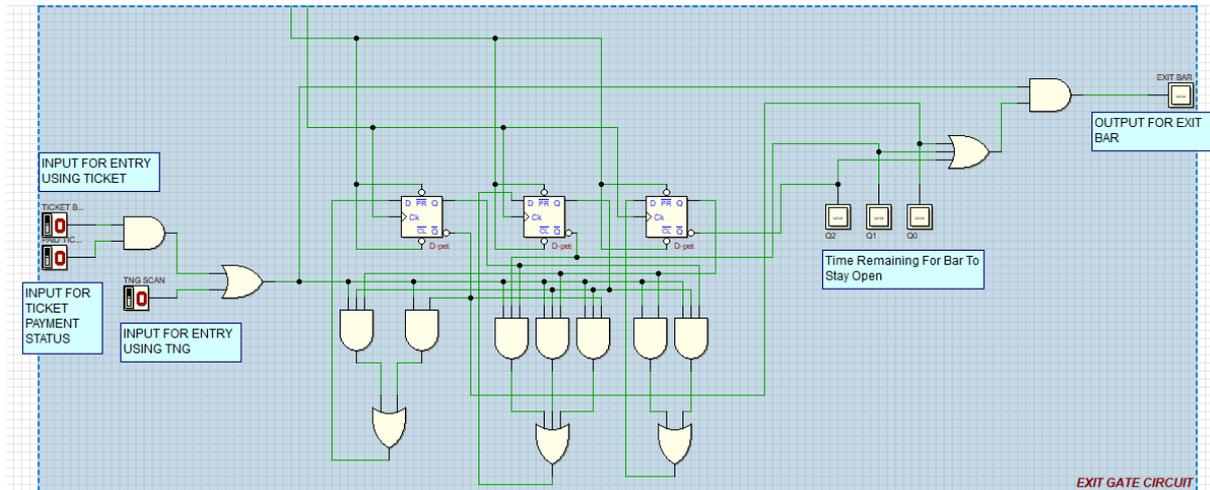
The entry gate circuit has 3 inputs. These are the Ticket Input, the TNG Input, and the output from the Total Free Space Display as the third input. The circuit will allow the users to enter by using either the Ticket Input or the TNG input. Next, if the circuit detects that there's free space available from the Total Free Space Display, the counter input for the circuit will be 1 if either the Ticket Input or TNG Input is 1. This means the input formula for the counter is:

$$\text{-Counter Input} = (\text{Ticket Input OR TNG Input}) \text{ AND Total Free Space Input}$$

The circuit uses a 3-bit synchronous down counter as a timer. The timer consists of 3 D Flip-Flops. The timer will start counting down if the counter input is 1 (formula shown above). The frequency for the Clock Input is 1 Hz, which means it will count for 7 seconds (111 until 000). While the timer is not 000 and the counter input is still 1, the output for the Entry Bar will remain open (1). But when the timer reaches 000, the Entry Bar output will close (0) even if the counter input is still 1.

The timer will stay at 000 until the counter input is 0. This is done by changing the TNG and Ticket input to 0. When the counter input is 0, the timer will reset from 000 to 111. Then, it will wait for the next input to start the countdown again.

### 5.3.2 Exit Gate Circuit



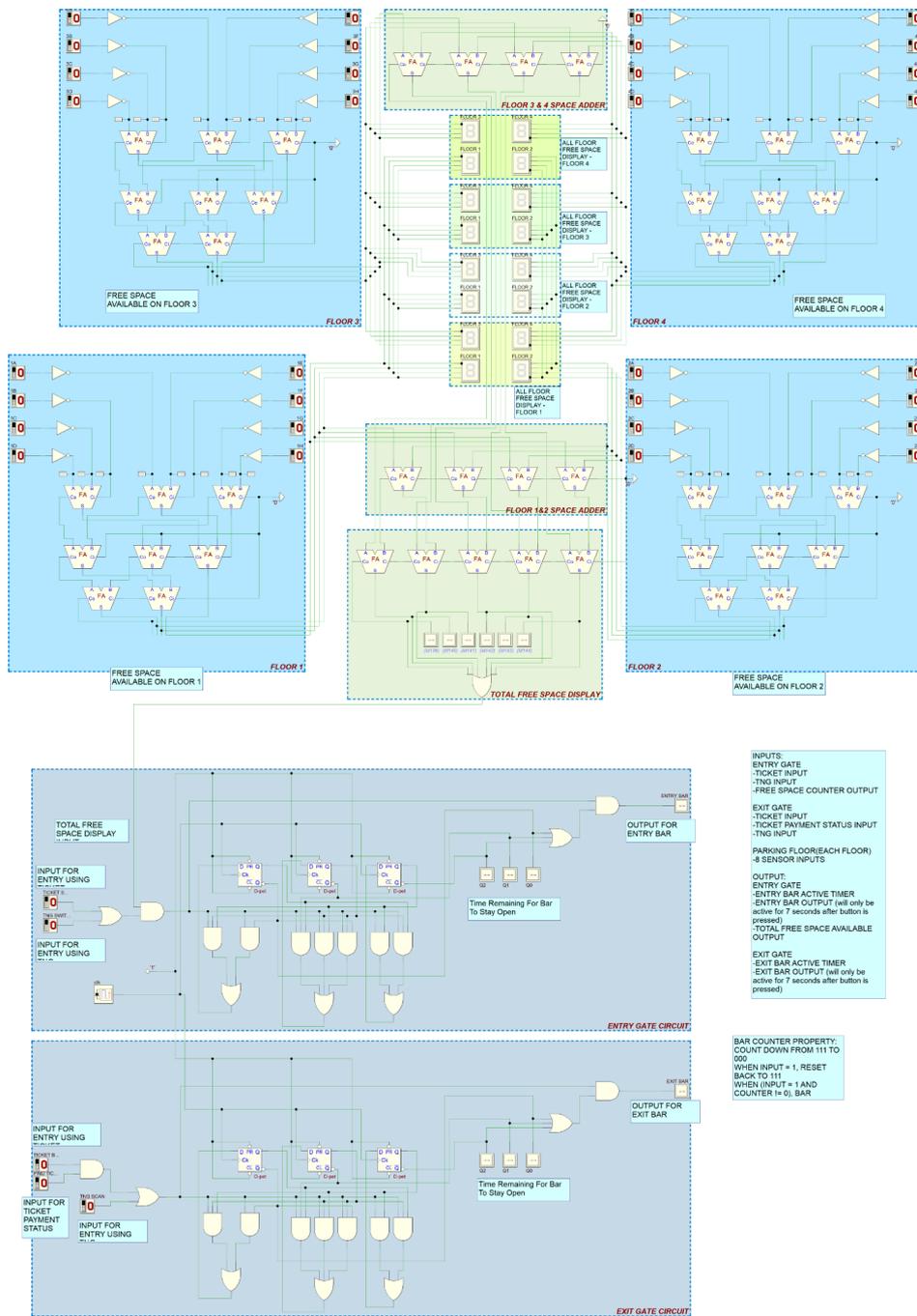
The exit gate uses the same timer counter as the entry gate, which counts for 7 seconds from 111 to 000. But the inputs for the exit gate circuit are different compared to the entry gate circuit.

The exit gate circuit has 3 inputs, which are the Ticket Input, Ticket Payment Status Input, and the TNG Scan input. The Exit Gate does not have Total Free Space Counter as input, since the gate will open even if the parking space is fully filled, For the counter to get 1 as input, the TNG Scan input must be 1, or the Ticket Input and the Ticket Payment Status Input must be 1. This means the formula is:

$$\text{-Counter Input} = (\text{Ticket Input AND Ticket Payment Status Input}) \text{ OR TNG Input}$$

When the counter input is 1, the counter timer will start counting down from 7 (111) to 0 (000). The Exit Bar will open if the Counter Input is 1, and the counter timer has not reached 000 yet. Then, the exit bar will close when the timer is 000. The timer will stay at 000 until the counter input is 0, in which it will reset back to 111. Then, the timer will wait for the next car to begin the countdown.

## 5.4 Overall System Implementation



This is the image of the overall system circuit, and how they are connected.

## 6. Conclusion

### Part A & B:

In short, we will be using a smart parking system, where customers can find parking spaces with ease. This is because there will be a counter in the parking lot, which indicates the number of car parking spaces available. Thus, customers will now know whether a parking slot is available or not, without having to make multiple rounds just to find a parking space, which is time consuming. To make the counter, we have designed a circuit for it as shown above.

With this system, we achieve the reducing of time consuming for user to park a car. With the information from space counter, driver can easily know which area or which floor have an empty parking lot for them. This will save a lot of time compare to the old parking system that requires driver to drive around for a certain amount of time for them to able to find a parking spot. Not with our new smart system that will let driver understand the exact floor of area that have empty spaces and driver can head straight to the space without hanging around.

For our full circuit, we designed our circuit, where we also included the entry and exit barrier. There, we divided into two circuits, one for the entry gate circuit and the other for the exit gate circuit. A counter timer is also added in both circuits each that has a frequency of 1Hz which is also equivalent to 7 seconds. With the 7-second timer, our barrier will be set to only let a single car to pass through at a single time. This will prevent the barrier for being lifted too long in time and let a second or third vehicle to pass through without ticket.

Besides that, we manage to solve driver that pay for the entry ticket and doesn't get a parking space and waste their time and money. With the space counter system, driver can be inform wether there is a space available inside before get to pay at the barrier, this will prevent some driver to waste their money.

The main weakness of our system is that it will be hard to understand by some people, for example the first time user or the people who doesn't have the ticket. It will be a huge problem as they will stuck at the barrier and cannot turn back as it is a one way line and causing long queue.

The second weakness of our system is that, it requires electricity for it to able to function. If there was a electricity cut down, then whole system will be not functional and cars won't be able to get in as the barrier requires electricity to lift up.

For our future work, we will also make a guiding video on the screen leading the first time user. Plus, we will design a system specifically use when the electricity were cut off and add emergency powers supply storage.