DESIGN OF DIGITAL SYSTEM PROJECT
ON
<u>AUTOMATED PARKING SYSTEM</u>

By Tanya Priyadarshini

INTRODUCTION

Automation plays a vital role in modern development, particularly in enhancing efficiency and organization. Space management and timekeeping have long posed challenges, especially in densely populated or busy areas, due to the limited adoption of advanced technology. As the world progresses across various domains, the need for intelligent infrastructure becomes essential. With this motivation, we have undertaken the design of an **Automated Parking System** to contribute to smart urban solutions.

OBJECTIVE

The primary goal of this project is to develop a real-time automated parking solution using basic digital integrated circuits. This system is designed to manage vehicle entry and exit in public parking areas by automatically controlling access based on availability. It prevents over-parking by allowing or denying entry depending on current occupancy and calculates the parking fee based on the duration of stay, ensuring accuracy and ease for both users and operators.

APPLICATION SCOPE

The scope of this project is to target places where there is a need for effective management of parking thereby avoiding congestion. It can be implemented in any public places. For example, shopping malls, airport, stadium, etc.

LIST OF COMPONENTS USED

1. Decade counter – IC 74LS90N
2. 4-bit Binary adder – IC74LS83N
3. Memory element – 2K8RAM
4. BCD Up/Down counter – 4510BD_10V
5. 4-bit Magnitude Comparator – IC74LS85N
6. Seven segment display
7. Basic gates (AND, OR, NOT, NOR)
8. Digital Clock
9. Voltage Sources
10. Connecting wires
11. Single Pole Double Throw (SPDT) switch

1. DECADE COUNTER – IC 74LS90N

A decade counter is one that counts in decimal digits, rather than binary. A decade counter uses a cascaded set of mod 2 and mod 5 counters. The normal clock is passed to the mod 2 counter, and for the mod 5 counter the single output of the mod 2 counter is fed as a clock. As the mod 5 counter resets after reaching 100 and the mod 2 counter resets after 1,the entire counter resets after reaching 1001(nine), thereby functioning as DECADE COUNTER.

State Diagram
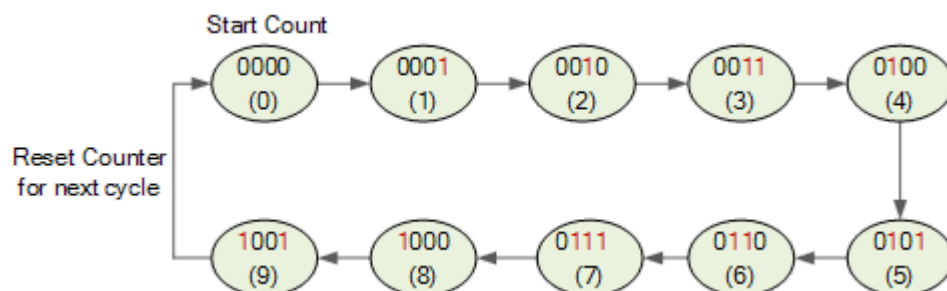
Pin Diagram
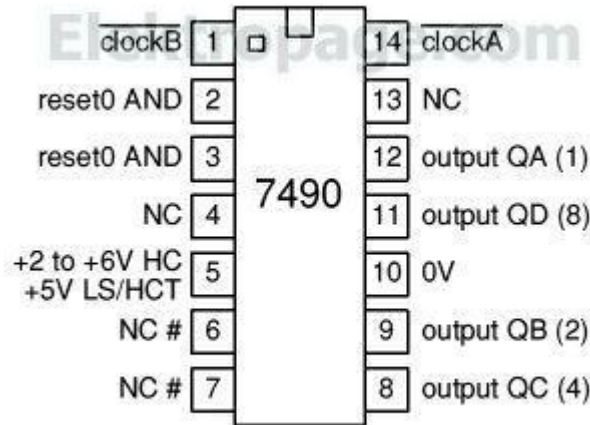
In this project we use 2 Decade counters for the Main Clock. One of the decade counter is used to increase the count of the main clock. The clock to the decade counter is set to a frequency of 10Hz.

2. 4-BIT BINARY ADDER – IC74LS83N

These full adders perform the addition of two 4-bit binary numbers. The sum ($\sum$) outputs are provided for each bit and the resultant carry (C4) is obtained from the fourth bit. These adders feature full internal look ahead across all four bits. This provides the system designer with partial look ahead performance at the economy and reduced package count of a ripple-carry implementation. The adder logic, including the carry, is implemented in its true form meaning that the end-around carry can be accomplished without the need for logic or level inversion.

Internal Circuit Diagram
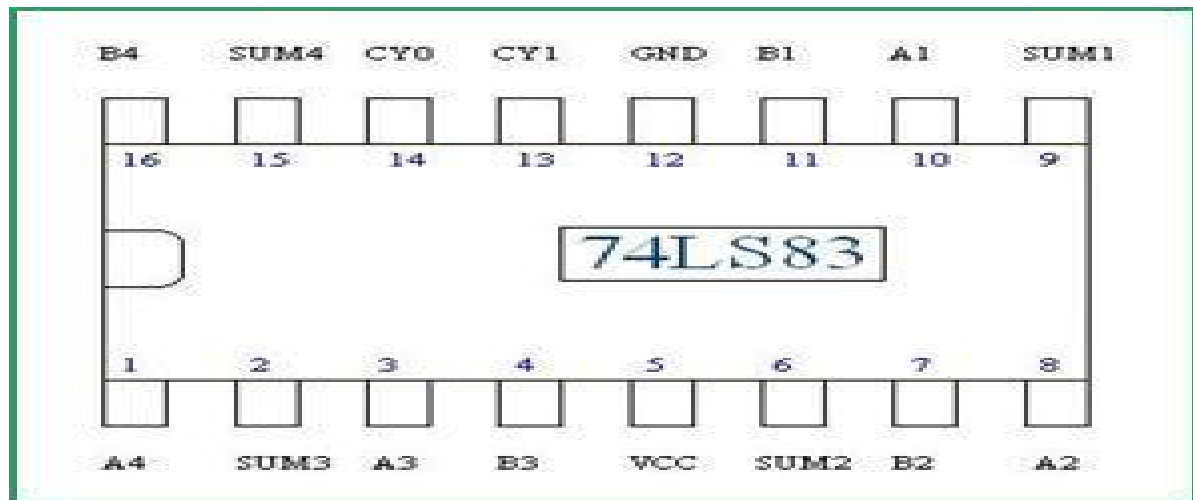
Logic diagram
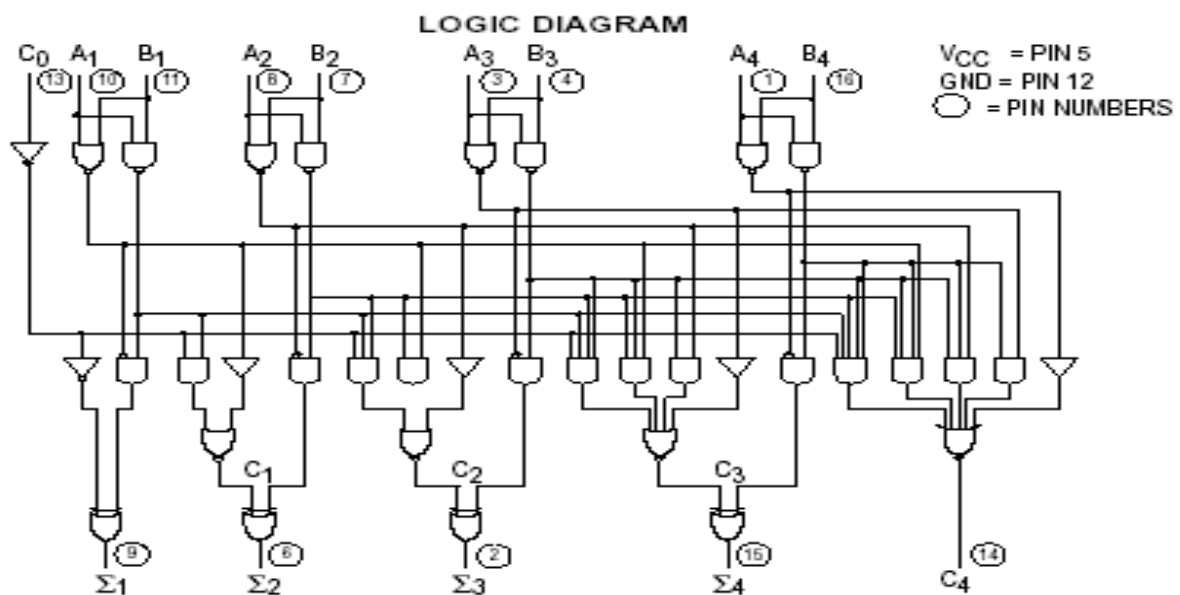
FUNCTIONAL TRUTH TABLE

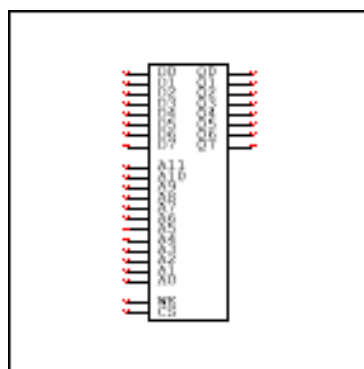| C (n–1)	A_n	B_n	Σ_n	C_n
L	L	L	L	L
L	L	H	H	L
L	H	L	H	L
L	H	H	L	H
H	L	L	H	L
H	L	H	L	H
H	H	L	L	H
H	H	H	H	H

$C_1 - C_3$ are generated internally
C_0 — is an external input
C_4 — is an output generated internally

3. MEMORY ELEMENT – 2K8RAM

The 2K8RAM is a memory component that lets you write and read 8 bit data; it has 4096 memory addresses where you to store your data.

Pin Diagram

Pin Description

D0 -D7 - Digital inputs

A0 - A11 – Addresses

WE - Write when high and read when low

CS - Address select

Q0-Q7 - Digital outputs

## *Truth Table*

| CS	WE	ACTION
0	X	No Action
1	0	Read from Address
1	1	Write to Address

4. BCD UP/DOWN COUNTER – 4510BD_10V

The HEF4510B is an edge-triggered synchronous up/down BCD counter with a clock input (CP), an up/down count control input (UP/DN), an active LOW count enable input (CE), an asynchronous active HIGH parallel load input (PL), four parallel inputs (P0 to P3), four parallel outputs (O0 to O3), an active LOW terminal count output (TC), and an overriding asynchronous master reset input (MR). Information on P0 to P3 is loaded into the counter while PL is HIGH, independent of all other input conditions except the MR input, which must be LOW. With PL LOW, the counter changes on the LOW to HIGH transition of CP if CE is LOW. UP/DN determines the direction of the count, HIGH for counting up, LOW for counting down. When counting up, TC is LOW when O0 and O3 are HIGH and CE is LOW. When counting down, TC is LOW when O0 to O3 and CE are LOW. A HIGH on MR resets the counter (O0 to O3 = LOW) independent of all other input conditions.
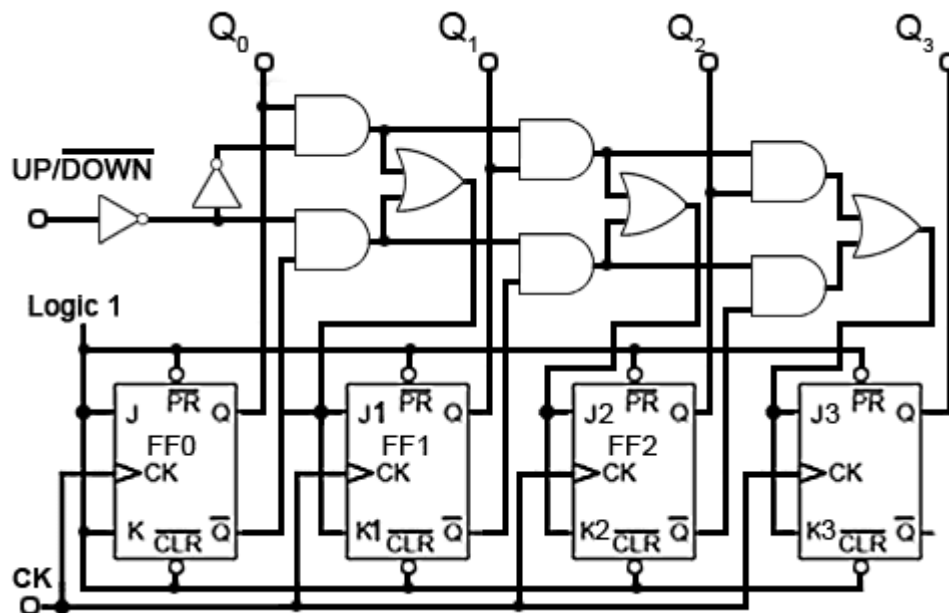
Pin Diagram
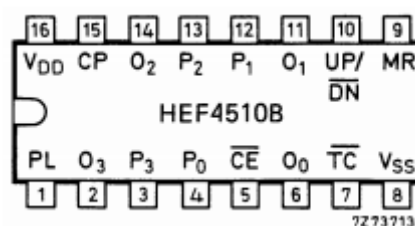
State Diagram

Fig.5 State diagram.
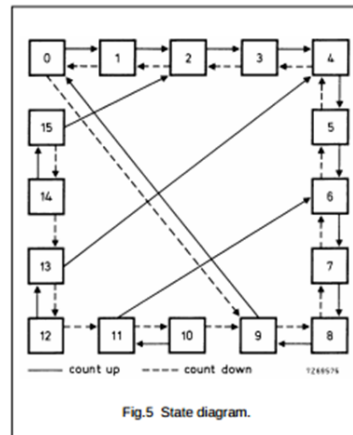
Internal Circuit

5. 4-BIT MAGNITUDE COMPARATOR – IC74LS85N

These 4-bit magnitude comparators perform comparison of straight binary or BCD codes. Three fully-decoded decisions about two 4-bit words (A, B) are made and are externally available at three outputs. These devices are fully expandable to any number of bits without

external gates. Words of greater length may be compared by connecting comparators in cascade. The A > B, A < B, and A = B outputs of a stage handling less-significant bits are connected to the corresponding inputs of the next stage handling more-significant bits. The stage handling the least-significant bits must have a high-level voltage applied to the A = B input. The cascading paths are implemented with only a two-gate-level delay to reduce overall comparison times for long words.

Pin Diagram
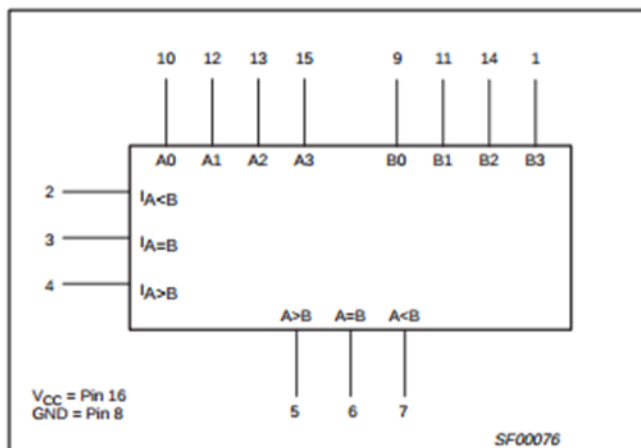
Truth Table

COMPARING INPUTS				EXPANSION INPUTS			OUTPUTS		
A3,B3	A2,B2	A1,B1	A0,B0	$I_{A>B}$	$I_{A<B}$	$I_{A=B}$	A>B	A<B	A=B
A3>B3	X	X	X	X	X	X	H	L	L
A3<B3	X	X	X	X	X	X	L	H	L
A3=B3	A2>B2	X	X	X	X	X	H	L	L
A3=B3	A2<B2	X	X	X	X	X	L	H	L
A3=B3	A2=B2	A1>B1	X	X	X	X	H	L	L
A3=B3	A2=B2	A1<B1	X	X	X	X	L	H	L
A3=B3	A2=B2	A1=B1	A0>B0	X	X	X	H	L	L
A3=B3	A2=B2	A1=B1	A0<B0	X	X	X	L	H	L
A3=B3	A2=B2	A1=B1	A0=B0	H	L	L	H	L	L
A3=B3	A2=B2	A1=B1	A0=B0	L	H	L	L	H	L
A3=B3	A2=B2	A1=B1	A0=B0	L	L	H	L	L	H
A3=B3	A2=B2	A1=B1	A0=B0	X	X	H	L	L	H
A3=B3	A2=B2	A1=B1	A0=B0	H	H	L	L	L	L
A3=B3	A2=B2	A1=B1	A0=B0	L	L	L	H	H	L

H = High voltage level
L = Low voltage level
X = Don't care

Circuit Diagram
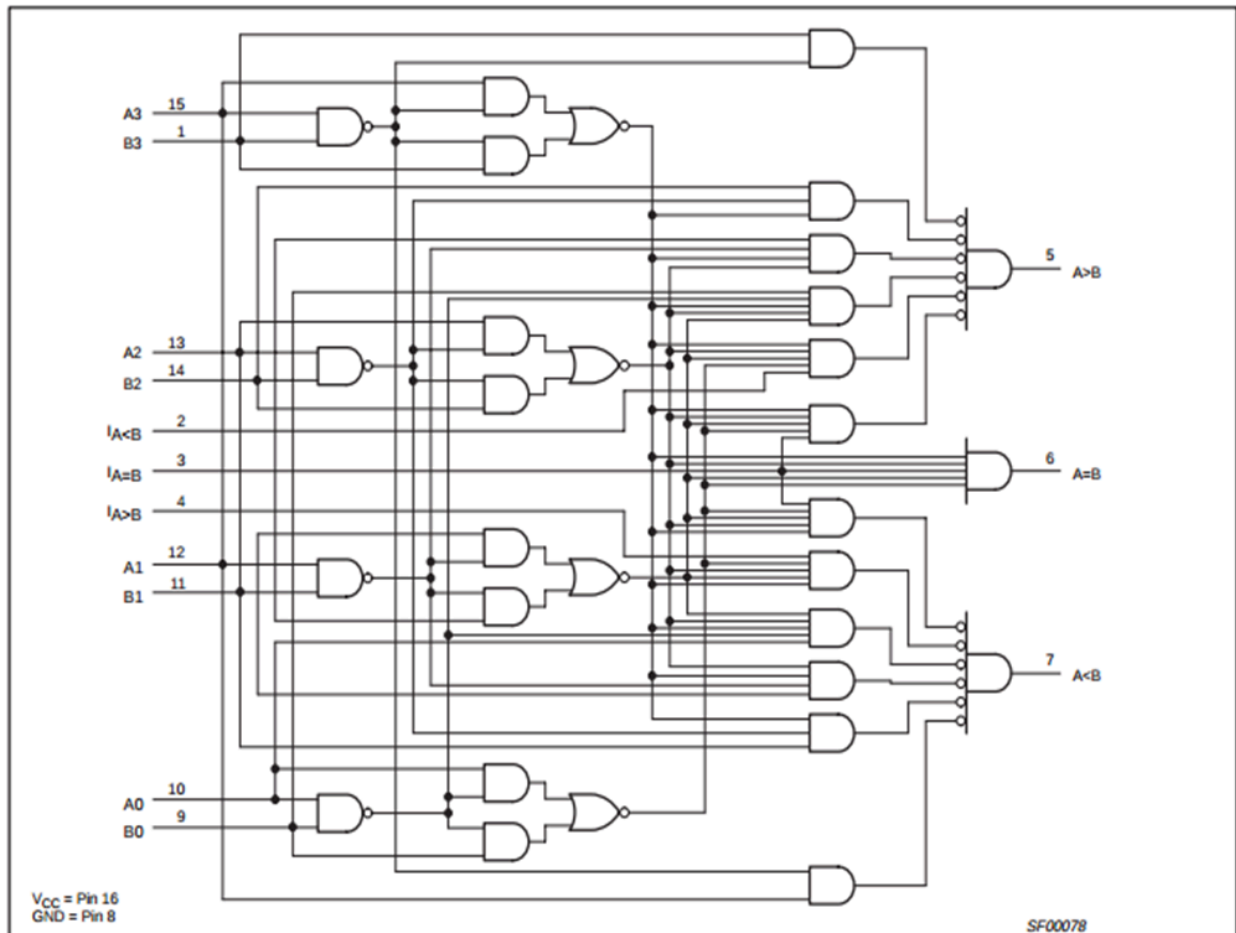
6. SEVEN SEGMENT DISPLAY

7-segment LED (Light Emitting Diode) or LCD (Liquid Crystal Display) type displays, provide a very convenient way of displaying information or digital data in the form of numbers, letters or even alpha-numerical characters.

Typically 7 segment displays consist of seven individual coloured LED's (called the segments), within one single display package. In order to produce the required numbers or HEX characters from 0 to 9 and A to F respectively, on the display the correct combination of LED segments need to be illuminated and BCD to 7-segment Display Decoders such as the 74LS47 does just that.

A standard 7-segment LED display generally has 8 input connections, one for each LED segment and one that acts as a common terminal or connection for all the internal display segments. Some single displays have also have an additional input pin to display a decimal point in their lower right or left hand corner.

In electronics there are two important types of 7-segment LED digital display:

1. *The Common Cathode display*
2. *The Common Anode display*

We use a common cathode display which is explained below.

The Common Cathode Display (CCD) – In the common cathode display, all the cathode connections of the LED's are joined together to logic "0" or ground. The individual segments are illuminated by application of a "HIGH", logic "1" signal to the individual Anode terminals.
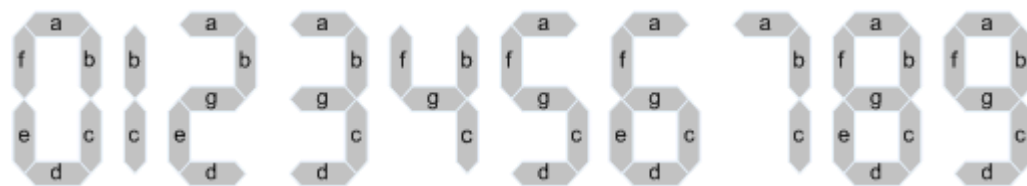
7-Segment Display Elements for all Numbers.

7. BASIC GATES

AND gate

It is a logic gate which takes in two or more inputs and gives one output. The output will be 1 if and only if all the inputs are 1. In other cases output will be 0.

OR gate

It is a logic gate which takes in two or more inputs and gives one output. The output will be 1 any one of the inputs is 1. It will be 0 only if both of the inputs are 0.

NOT gate

It is a logic gate which takes in one input and gives one output. The output will be 1 if the input is 0 otherwise 1.

NOR gate

It is a logic gate which takes in two or more inputs and gives one output. The output will be 0 if any one of the inputs is 1. It will be 1 if all the inputs are 0.

8. DIGITAL CLOCK

It is used to supply the input pulse. It is a function generator which supplies square pulses at a particular frequency.

In this project we use the function generator to generate a square pulse of frequency 10Hz for to the decade counter which is used to maintain the main clock.

9. VOLTAGE SOURCE

We use two voltage sources. One is ground which be used to make the particular inputs low. To make them high we use VCC of particular high voltage. Here we set VCC as 5 volts.
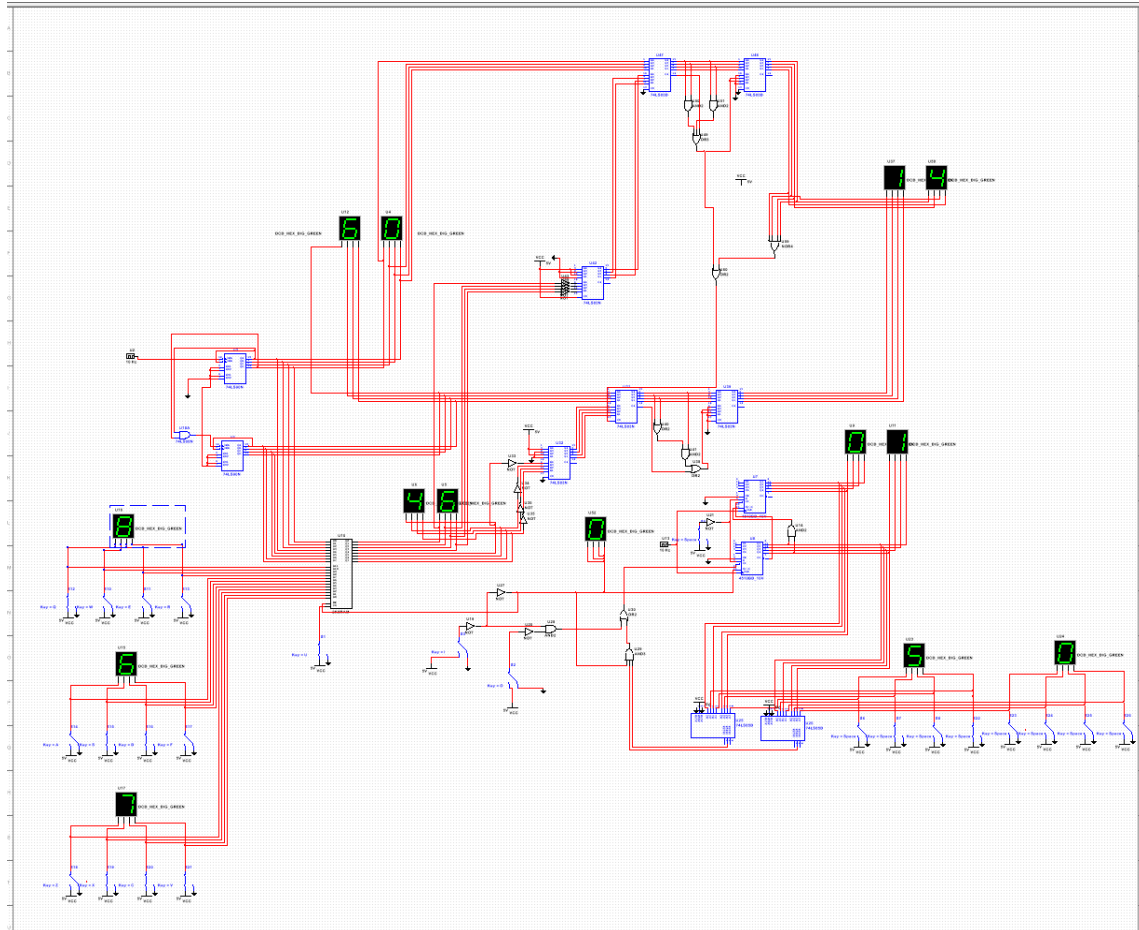
10. CONNECTING WIRES

They are used to connect different components with each other.

11. SINGLE POLE DOUBLE THROW SWITCH

It is used to connect a given part of a circuit to two different parts of a circuit with only one operational at a given point of time and it allows switching between two operations.
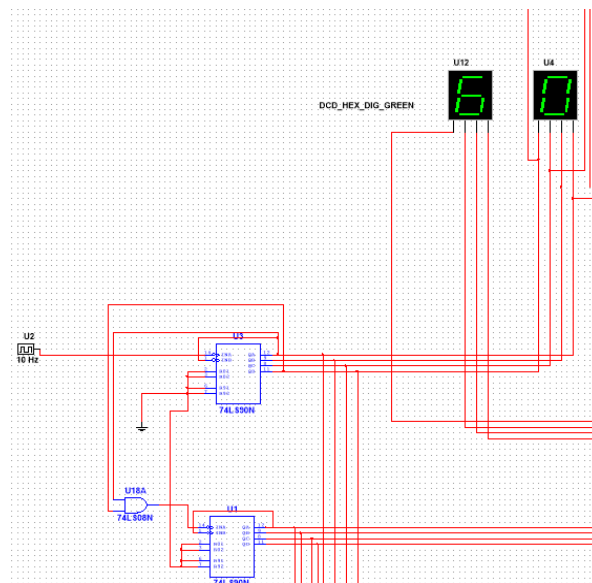
LOGIC CIRCUIT

# METHODOLOGY ADOPTED (IMPLEMENTATION DETAILS)

This entire project can be divided into 4 subsections which perform specific tasks and display corresponding outputs.

Master Clock

The main time element of this project is the master clock. 2 decade counters are used for this purpose. The clock input for the 1st counter is taken from the digital clock of frequency 10Hz. When the 1st counter counts up to 9 it is reset and the count of the 2nd counter is increased by 1. The master clock runs from 00 to 99 and its value is used by the memory element and the cost calculator.

Memory Element and Car number Input
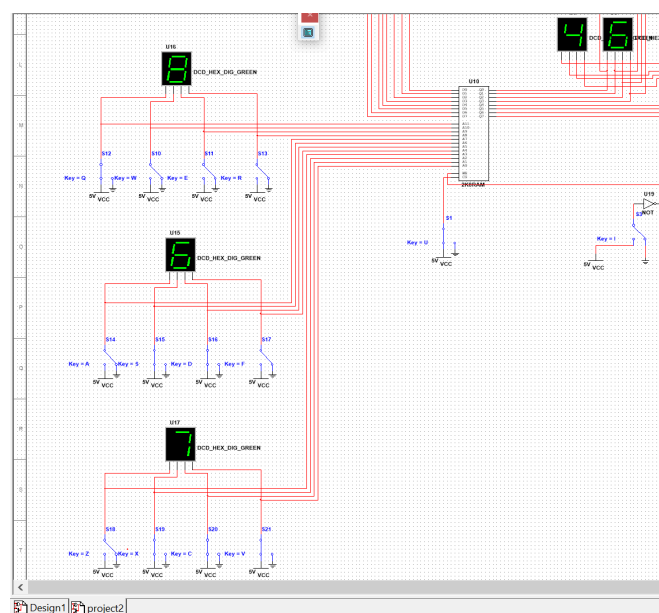
A BCD requires 4 bits, so here we have used 4 switches per input. We have assumed that a car number consists of three numbers making a sum of 12 bits which is fed as address lines to the 2K8RAM. The data input to the 2K8RAM is taken from the master clock. Whenever a car enters the Write pin of the 2K8RAM is enabled and the present time is stored in the RAM. By default each car number is initialised to a time value of 0. When the input to the write pin is low the data stored in the address specified by the car number is read and fed to the seven segment display and also forwarded to the cost calculator.

Car Counter
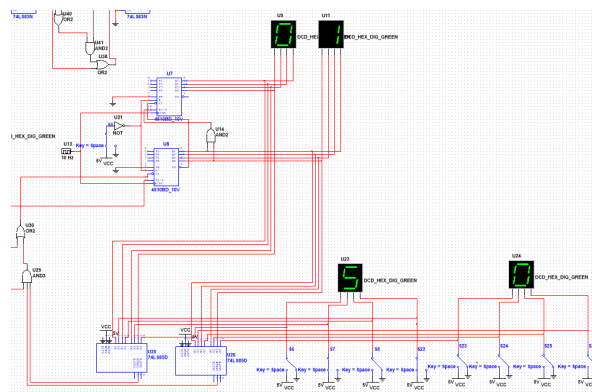
The capacity of the parking lot can be modified using 8 switches (2 BCD numbers).

Whenever a car enters the parking lot the comparator checks whether there are any vacancies. This is done using comparators. The input to the comparator is the maximum count (50 in the above figure) and the present car count. If the present count is lesser than the maximum count then the car is allowed to park and the count is increased by 1.

Whenever the car leaves the parking lot the present count is decreased by 1 and the cost of that car is calculated using the car cost calculator. Present count is stored and changed accordingly using a BCD UP/DOWN counter.

Parking Fee Calculator
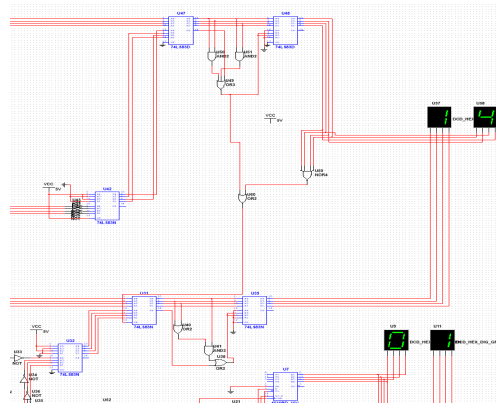
When a car is leaving the parking lot the cost has to be calculated and displayed. This is done using binary adders. The present time is taken from the master clock and the time at which the car entered is retrieved from the 2K8RAM and fed to the BCD subtractor. BCD subtractor is implemented using 4-bit binary adders using 9's complement and 10's complement method. The BCD subtractor calculates the difference between the present time and the stored time and displays it as the amount to be paid.

MERITS

1. Efficient way to implement.

2. Less prone to manual errors since the calculation of parking fee or maintaining the count of cars is done by the electronic components.

3. It's very easy to operator on.

4. Tells if there is a vacant place in the parking system.

DEMERITS

1. At each instant of time either one of input or output happens, thus increasing the waiting time of cars in the queue.

2. There is always an upper bound for master Clock.

SCOPE FOR IMPROVEMENT

These are some of the things which can improve the present system.

- Car numbers are usually a combination of letters and numbers and a better mapping mechanism needs to be designed to address this.
- Parking fee is some constant multiplied by the time and a multiplier can be used for this.